ULBRA CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

FLÁVIO HENRIQUE MOURA STAKOVIAK

IMPLANTAÇÃO DE UM MECANISMO DE ENRIQUECIMENTO DO PERFIL DO USUÁRIO E RECOMENDAÇÃO DE TRABALHOS CIENTÍFICOS PARA O KONNEN

Palmas

2011

FLÁVIO HENRIQUE MOURA STAKOVIAK

IMPLANTAÇÃO DE UM MECANISMO DE ENRIQUECIMENTO DO PERFIL DO USUÁRIO E RECOMENDAÇÃO DE TRABALHOS CIENTIFÍCOS PARA O KONNEN

Trabalho apresentado como requisito parcial da disciplina Trabalho de Conclusão de Curso (TCC) do curso de Sistemas de Informação, orientado pelo Professor Mestre Edeílson Milhomem da Silva.

Palmas

2011

FLÁVIO HENRIQUE MOURA STAKOVIAK

IMPLANTAÇÃO DE UM MECANISMO DE ENRIQUECIMENTO DO PERFIL DO USUÁRIO E RECOMENDAÇÃO DE TRABALHOS CIENTÍFICOS PARA O KONNEN

Trabalho apresentado como requisito parcial da disciplina Trabalho de Conclusão de Curso (TCC) do curso de Sistemas de Informação, orientado pelo Professor Mestre Edeílson Milhomem da Silva.

Aprovada em xxxxxxx de 2011.

BANCA EXAMINADORA

_____

Prof. M.Sc. Edeílson Milhomem da Silva

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Jackson Gomes de Souza

Centro Universitário Luterano de Palmas

Palmas

2011

SUMÁRIO

AGRADECIMENTOS

É difícil agradecer a todas as pessoas que de algum modo, nos momentos serenos e ou apreensivos, fizeram ou fazem parte da minha vida, por isso primeiramente agradeço a todos de coração.

Dedico este trabalho “in memorian” aos meus avós Flaviano Moura e Ziza, e aproveito também para agradecê-los, estejam onde estiverem. Lembro-me de meu avô dizendo que a maior virtude do homem era a educação, pois era a única coisa que ninguém poderia roubar do indivíduo.

Agradeço a minha mãe, irmão, tios, primos, amigos e professores pela compreensão, apoio e incentivo.

RESUMO

Sistemas de Recomendação auxiliam usuários na busca de informações úteis a partir dos interesses do usuário, e podem ser utilizados por uma variedade de domínios que vão desde a recomendação de páginas web até a recomendação de artigos científicos. Garantir a qualidade das recomendações requer a utilização de técnicas que processem os dados de forma eficiente, sendo que tal eficiência é alcançada pela Recuperação da Informação, que é responsável por representar, armazenar e organizar esse conteúdo. A partir disso, este trabalho busca desenvolver uma ferramenta que enriqueça o perfil do usuário do Konnen, com a recomendação de artigos científicos relevantes ao seu perfil cadastrado na Plataforma Lattes.

PALAVRAS-CHAVE: Sistemas de Recomendação, Recuperação da Informação, *Clustering*

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

DBSCAN	*Density-based Spatial Clustering of Applications with Noise*
EM	*Expectation Maximization*
FBC	Filtagem baseada em Conteúdo
FC	FIltragem Colaborativa
HTML	*HyperText Markup Language*
RI	Recuperação da Informação
RSLP	Removedor de Sufixo da Língua Portuguesa
SR	Sistemas de Recomendação
URL	*Uniform Resource Locator*

1 INTRODUÇÃO

Com o advento da *internet*, os documentos, artigos científicos, relatórios, livros, etc. que antes eram trabalhados e armazenados apenas em papel, passaram a ser digitalizados e o armazenamento eletrônico destes documentos vem crescido de maneira exponencial. Uma vez que estas informações estejam disponíveis em um formato digital, a adoção de tecnologias, para o desenvolvimento de soluções, tornou viável a utilização destes dados para análise de vantagem competitiva entre empresas, ou até mesmo como auxílio à tomada de decisões.

Neste contexto, surgem os Sistemas de Recomendação, que podem auxiliar usuários e instituições a encontrar informações úteis, baseadas em seus interesses. As aplicações destes sistemas podem fornecer respostas a indagações como, por exemplo, qual funcionário é capaz de lidar com um problema encontrado em determinado projeto, ou quais documentos podem estar relacionados a uma informação que está sendo consultada. Os resultados dessas recomendações são computados a partir da Recuperação da Informação.

A Recuperação da Informação lida com a representação, com o armazenamento, com a organização e acesso de conteúdo. A representação e organização da informação devem fornecer ao usuário acesso simples e objetivo de acordo com suas preferências. Entretanto, caracterizar a necessidade do usuário para uso da informação não é simples. No cenário virtual, por exemplo, a quantidade de dados não estruturados e em sua maioria, sob o formato textual, não fornece parâmetros suficientes para uma busca eficiente de informações. Surge então, a necessidade de organizá-los em uma estrutura lógica, para que possam ser indexados e, consequentemente, recuperados de forma mais rápida.

O C*lustering* de Documentos tem a organização hierárquica como um de seus objetivos e é empregado com frequência na organização de resultados retornados por mecanismos de buscas, além de ser utilizado para percorrer grandes coleções de documentos e categorizá-los. A utilização de métodos de *Clustering* em vários contextos, por diferentes áreas, reflete sua grande utilidade na exploração de conhecimento sobre dados.

Desta forma, o principal objetivo deste trabalho é apresentar a utilização de *clustering* para buscar publicações científicas contidas no endereço eletrônico do Currículo Lattes dos usuários cadastrados no Konnen. Para isso, serão utilizados os conceitos de *clustering*,

visando assim organizar os artigos por similaridade utilizando os termos presentes no conteúdo dos mesmos. Possibilitando assim, buscar conteúdos relacionados em sistemas de consulta a publicações disponíveis na Web.

A estrutura deste trabalho é organizada da seguinte forma: o capítulo 2 (dois) apresenta conceitos relacionados a Sistemas de Recomendação, abordando os principais conceitos sobre as técnicas de filtragem baseada em conteúdo e filtragem colaborativa. Em seguida, descreve-se a análise realizada sobre a Recuperação da Informação, apresentando as etapas de recuperação que envolvem a aquisição, preparação, indexação, armazenamento e recuperação da informação. Por fim, relata-se sobre a metodologia de classificação automática de dados, *Clustering* de Documentos, bem como as técnicas envolvidas neste processo de classificação que são: *Clustering* hierárquico, Algoritmos particionais, *Density-based* e *Model-based*. O capítulo 3 (três) descreve os materiais utilizados para o desenvolvimento deste trabalho, além da metodologia adotada. O capítulo 4 (quatro) expõe os resultados obtidos. Por fim, são apresentadas as conclusões obtidas após todo o desenvolvimento deste trabalho, são apresentadas no capítulo 5 (cinco).

2 REFERENCIAL TEÓRICO

Esta seção apresentará os principais conceitos e definições relacionados a Sistemas de Recomendação, Recuperação de Informação e *Clustering* de Documentos. O entendimento destes conceitos é premissa necessária para o desenvolvimento do projeto proposto neste trabalho, que é o desenvolvimento de um mecanismo capaz de prover recomendações a partir do uso de clusterização.

2.1 Sistemas de Recomendação

Sistemas de recomendação (SR) são responsáveis por identificar um usuário e lhe apresentar conteúdo, produtos ou ofertas personalizadas (REATEGUI, BOFF & VICCARI, 2005, p. 478). Um importante exemplo de aplicação dos Sistemas de Recomendação são os sites de comércio eletrônico, que se utilizam dos benefícios proporcionados pelos SR para incentivar os seus clientes a consumirem.

Sistemas de recomendação podem ser aplicados também para apoiar um grupo ou uma equipe de trabalho. Nesta esfera a recomendação se dá a partir de informações relevantes acerca dos interesses do grupo ou da comunidade. Assim, a cooperação entre as pessoas da comunidade deve acontecer com mais naturalidade, pelo fato das pessoas da comunidade possuírem objetivos em comum. Motta & Borges (2000 *apud* MOTTA & LOPES, 2002, p. 381) afirmam que as avaliações das informações aspiram a ser mais seguras nos Sistemas de Recomendação para grupo ou comunidade pelo fato de serem feitas por profissionais da área e não leigos.

Um Sistema de Recomendação tem como entrada as avaliações de itens. A partir dessas avaliações o sistema adiciona e direciona estas sugestões para os usuários considerados potenciais interessados neste tipo de recomendação (CAZELLA, 2006, p. 28). O autor afirma ainda que um dos grandes desafios deste tipo de sistema é estabelecer a relação adequada entre os itens que estão sendo analisados com os usuários que estão recebendo a citada sugestão, isto é, determinar e encontrar este relacionamento de importância. Assim que o relacionamento de importância é determinado e encontrado, tem-se um sistema que realmente

recomende itens interessantes ao usuário. A Figura 1 apresenta o processo de sugestão de um Sistema de Recomendação.

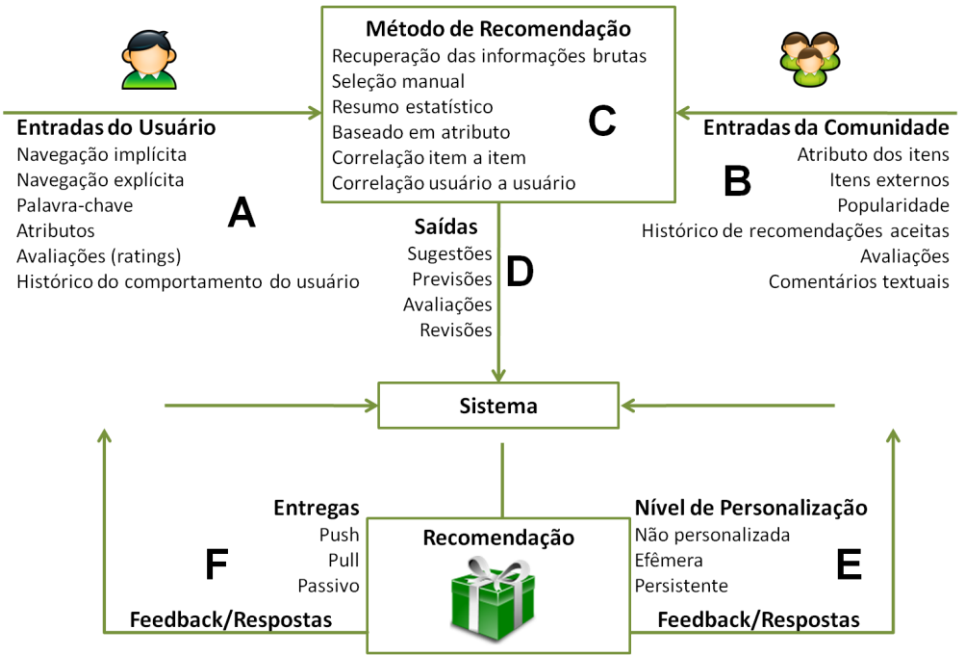

Figura 1: Taxonomia dos Sistemas de Recomendação (Modificada em SCHAFER, KONSTAN & RIEDL, 2001, tradução nossa, p. 10).

De acordo com a Figura 1A, para que haja recomendação é necessário que exista um perfil que represente as preferências dos usuários. Estas preferências podem ser representadas das mais variadas formas, como por exemplo, pelo histórico de navegação do usuário; por meio do histórico do conhecimento produzido pelo usuário, como por exemplo, um *post* em um *blog*; por meio de avaliações realizadas pelo usuário, dentre outras. Conforme a Figura 1B apresenta, os SRs podem ter também como entrada informações relacionadas à comunidade do site, que podem envolver atributos dos itens do sistema, popularidade destes itens, itens externos, histórico de recomendações aceitas e comentários textuais. As entradas referentes à comunidade servem como base para que o sistema faça a recomendação dos itens, uma vez que essa comunidade é composta por usuários com perfil semelhante ao usuário alvo. Uma vez que as preferências do usuário são representadas (Figura 1A), torna-se possível a aplicação de métodos e técnicas de Recomendação (Figura 1C), que podem ser realizadas a partir da recuperação das informações brutas, da correlação tanto entre item a item quanto usuário a usuário, dentre outras. Esta recomendação pode partir também da combinação das informações adquiridas sobre o usuário (Figura 1A) com as avaliações sobre os itens realizadas pela comunidade de usuários (Figura 1B). Obtidas as sugestões, previsões, avaliações e/ou revisões dos itens a serem sugeridos, Figura 1D, o nível de personalização e a

forma de apresentação são definidas, para que um bloco de recomendações específicas seja gerado (Figura 1E e 1F, respectivamente). Vale ressaltar que caso haja um *feedback* do usuário quanto à utilidade da recomendação, esta pode vir a servir como entrada adicional para as futuras recomendações.

Existem duas principais abordagens para a implementação de Sistemas de Recomendação, que são: Filtragem Baseada em Conteúdo (FBC) e Filtragem Colaborativa (FC). Apesar de serem abordagens distintas, ambas têm um único objetivo que é recomendar itens aos usuários. Na FBC, apenas o perfil do usuário e a base de itens são levados em consideração para selecionar os itens de sua preferência a serem recomendados. Por outro lado, a FC leva em consideração as preferências do usuário e dos demais usuários para realizar a comparação entre os perfis e analisar quais perfis são similares ao perfil do usuário alvo.

Como apresentado, este trabalho tem como foco criar um mecanismo de recomendação por meio da clusterização. Uma vez que *clustering* é um processo de aglomeração de dados fundamentados no grau de similaridade e este processo pode ser aplicado na recomendação de itens baseada em conteúdo, este trabalho abordará apenas a filtragem baseada em conteúdo, a qual será aplicada em conjunto com a clusterização.

2.1.1 Filtragem Baseada em Conteúdo

A filtragem baseada em conteúdo é definida a partir da similaridade entre os itens. Segundo Lichtnow *et al.* (2006, p. 50), a idéia desta abordagem é que os usuários têm uma tendência natural a se interessar por itens semelhantes aos que demonstraram interesse anteriormente. Desta forma, pode-se dizer que esta abordagem pode levar em consideração, para propor a recomendação, tanto a análise de conteúdo do item quanto o perfil do usuário.

O autor afirma ainda que para que seja possível estabelecer similaridades entre itens, faz-se necessária a identificação de atributos em comum que os itens apresentam para que, posteriormente, esses atributos possam ser utilizados como instrumento de comparação. Tais atributos correspondem às informações fornecidas pelo próprio usuário ou aos itens que o usuário consome. Caso fosse estabelecida a similaridade entre, por exemplo, calçados e perfumes, poder-se-ia levar em consideração atributos como preço, marca, dentre outros, para que seja realizada a comparação. “Já quando os itens correspondem a artigos (ou sites), este processo de comparação pode ser facilitado, pois documentos podem ser considerados similares se compartilharem termos em comum” (LICHTNOW et al., 2006, p. 50).

Além do fato da comparação entre itens não ser facilmente automatizada, a abordagem baseada em conteúdo não realiza a avaliação qualitativa dos itens recomendados (SHARDANAND & MAES, 1995, tradução nossa, p. 1). No caso de documentos, por exemplo, o fato de o conteúdo ser similar, ou seja, apresentar termos em comum, já é o suficiente para que dois textos sejam considerados semelhantes.

Como apresentado, a aplicação da filtragem baseada em conteúdo em Sistemas de Recomendação analisa o conteúdo que descreve o item a ser recomendado. Normalmente, muitas palavras do conteúdo do documento não são importantes para a análise de recomendação. Neste contexto, a recuperação da informação aplicada em conjunto com a filtragem de conteúdo, permitirá que informações úteis do texto sejam extraídas, a partir da utilização de técnicas como remoção de *stopwords*, *case folding*, *stemming* dentre outras, as quais serão apresentadas.

2.2 Recuperação da Informação

Recuperação da Informação (RI) ou *Information Retrieval* é uma área da computação responsável por manipular e recuperar informações procedentes de textos planos. A RI tem como objetivo auxiliar os usuários na busca por informações que os interessam em uma coleção de documentos. Tal auxílio se dá a partir da representação, do armazenamento, da organização e do acesso a essas informações (SILVA, 2006, p. 29; RODRIGUES, 2009, p. 19).

Para que RI possa ser aplicada em um conjunto de documentos é necessário, inicialmente, que o conteúdo dos documentos seja representado. Uma vez que tais documentos tenham sido representados, faz-se necessária também a representação das necessidades de informação do usuário, as quais servirão como elemento principal de seleção da informação. Por fim, é realizada uma comparação entre as representações do conteúdo dos documentos e das necessidades do usuário, para possibilitar o armazenamento de documentos e a recuperação automática de informação associada a esses documentos (CROFT, 1993 *apud* SILVA, 2006, p. 30). A Figura 2 apresenta o processo de recuperação da informação.

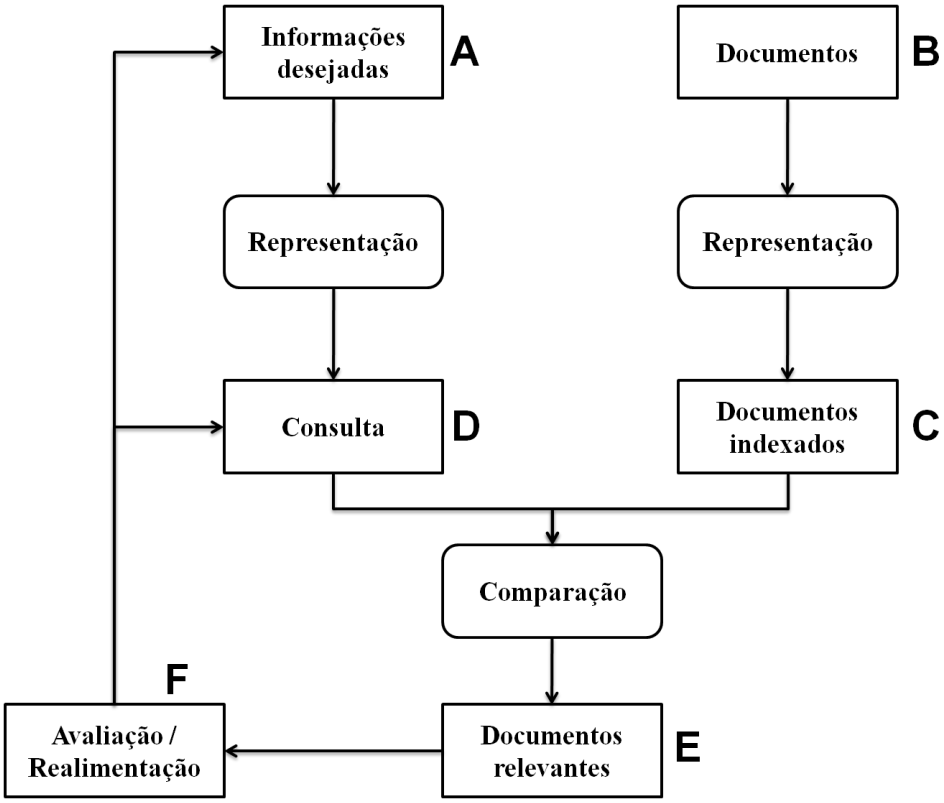

Figura 2: Processo de Recuperação da Informação (Modificado CROFT, 1993 *apud* SILVA, 2006, p. 30).

Conforme a Figura 2, o processo de recuperação da informação se dá a partir das informações desejadas pelo usuário (Figura 2A) e dos documentos disponíveis (Figura 2B). Os documentos (Figura 2B) são representados e é gerada a indexação dos mesmos (Figura 2C). Da mesma forma, as informações desejadas são representadas e é gerada uma expressão de consulta (Figura 2D). A partir de então, é realizada uma comparação entre os documentos indexados com a consulta gerada para que sejam apresentados os documentos julgados relevantes ao usuário (Figura 2E). A relevância dos documentos, Figura 2E, é calculada por meio do modelo de representação dos documentos, Figura 2C, e da consulta, Figura 2D, que é realizado no momento da comparação. Vale ressaltar que as informações desejadas (Figura 2A) são geradas a partir de uma requisição do usuário por meio de uma consulta (Figura 2D). Desta forma, essas informações podem ser informadas à medida que surge a necessidade de recuperar informações relevantes para o usuário, reiniciando todo o ciclo de recuperação (Figura 2F).

Segundo Cardoso (2002, p. 2-4), existem três modelos clássicos de Recuperação da Informação:

- Modelo Vetorial – considerado um modelo algébrico, utiliza vetores n-dimensional para representar os documentos, onde n indica a quantidade de termos únicos que ocorrem no interior de todos os documentos. A partir disso, busca-se encontrar os vetores mais próximos ao vetor equivalente à consulta submetida, ordenando o resultado conforme o grau de relevância do documento. Vale ressaltar que um documento é recuperado neste modelo, mesmo que este documento satisfaça a consulta de modo parcial.
- Modelo *Booleano* – baseado na teoria dos conjuntos, utiliza uma expressão lógica da consulta para recuperar os documentos, podendo essa expressão ser formada por elementos lógicos como *AND*, *OR* e *NOT*. A representação indica apenas se o termo está ou não presente no documento.
- Modelo Probabilístico – sendo considerada uma aplicação direta da teoria das probabilidades, utiliza pesos binários para representar os documentos, determinando a presença ou ausência de termos. Ou seja, a partir de uma consulta do usuário, há um conjunto de documentos que possui documentos relevantes e não-relevantes para o usuário.

Segundo Silva (2006, p. 31), "um dos maiores usos de recuperação de informação está associado à Internet.". Ainda segundo o autor, os mecanismos de buscas, responsáveis por tratar características específicas que a Web possui, buscam no ciberespaço e encontram documentos que possam ser indexados, utilizando *crawler*. A Figura 3 apresenta os processos de recuperação de informação na Web.

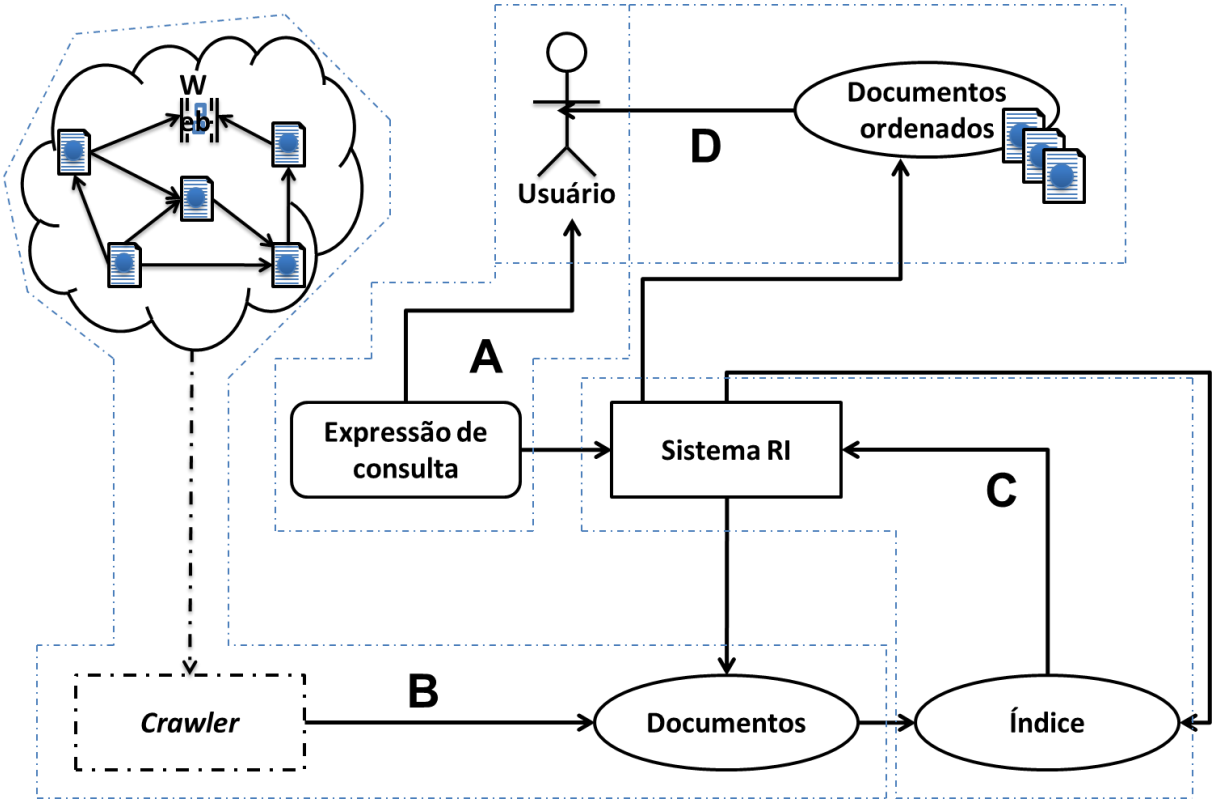

Figura 3: Processo de Recuperação de Informação na Web (Modificado SILVA, 2006, p. 31).

Como pode ser observado na Figura 3, o processo de recuperação de informação na Web se dá a partir de uma consulta realizada pelo usuário, resultando em uma expressão de consulta, a qual deve ser interpretada (Figura 3A). A coleção de documentos é utilizada para construir o sistema (Figura 3B), sendo estes representados de forma que possam ser "compreendidos" pelo sistema. Assim, o sistema RI produz um índice organizado (Figura 3C), que pode incluir o documento inteiro e seus metadados. Este índice é obtido por meio da comparação entre a expressão de consulta e o conjunto de documentos. Ao final do processo, tem-se uma lista de documentos ordenados pela sua relevância (Figura 3D), a qual é apresentada ao usuário de forma adequada com seu propósito, e na qual o usuário pode navegar e encontrar informações de que necessita.

De acordo com Baeza-Yates (1996, tradução nossa, p. 2), para que um sistema de RI possa ser desenvolvido, algumas etapas principais devem ser seguidas, são elas: Aquisição (seleção) dos documentos, Preparação dos documentos, Indexação dos documentos, Busca (casamento com a consulta do usuário) e Ordenação dos documentos recuperados. As próximas seções apresentarão detalhes do funcionamento de cada uma destas etapas.

2.2.1 Aquisição de Documentos

A premissa básica, para que um sistema de RI possa ser desenvolvido e implantado, é a existência de informações que possam ser analisadas. A aquisição (seleção) destes documentos é a primeira etapa de um sistema de RI, que visa a extrair dados para que os mesmos sejam tratados. A seleção de documentos pode ser realizada de forma manual ou automatizada. Direcionando para o contexto da Web, é imprescindível que esta seleção seja realizada de maneira automatizada, uma vez que a aquisição de documentos pode ser pertinente a um determinado contexto ou não. Neste sentido, existem os *Web Crawlers*, os quais serão apresentados na próxima seção juntamente com seus conceitos inerentes.

2.2.1.1. Web Crawler

Web Crawlers, também conhecidos como *Web Spider*, "são *softwares* que, uma vez alimentados por um conjunto inicial de URLs (sementes), iniciam o procedimento metódico de visitar um site, armazená-lo em disco e extrair deste os *hyperlinks*, que serão utilizados para as próximas visitas" (SOARES, 2008, p. 20). Ainda segundo o autor, "estes programas disparam, automaticamente, em intervalos de tempo, visitas às páginas Web, as lêem, copiam, e seguem os *hyperlinks* contidos nelas, construindo um repositório de informações, que é vinculado àquela pesquisa, e gerando a possibilidade de realizar a indexação dos documentos". A Figura 4 apresenta a arquitetura de um *Web Crawler*.

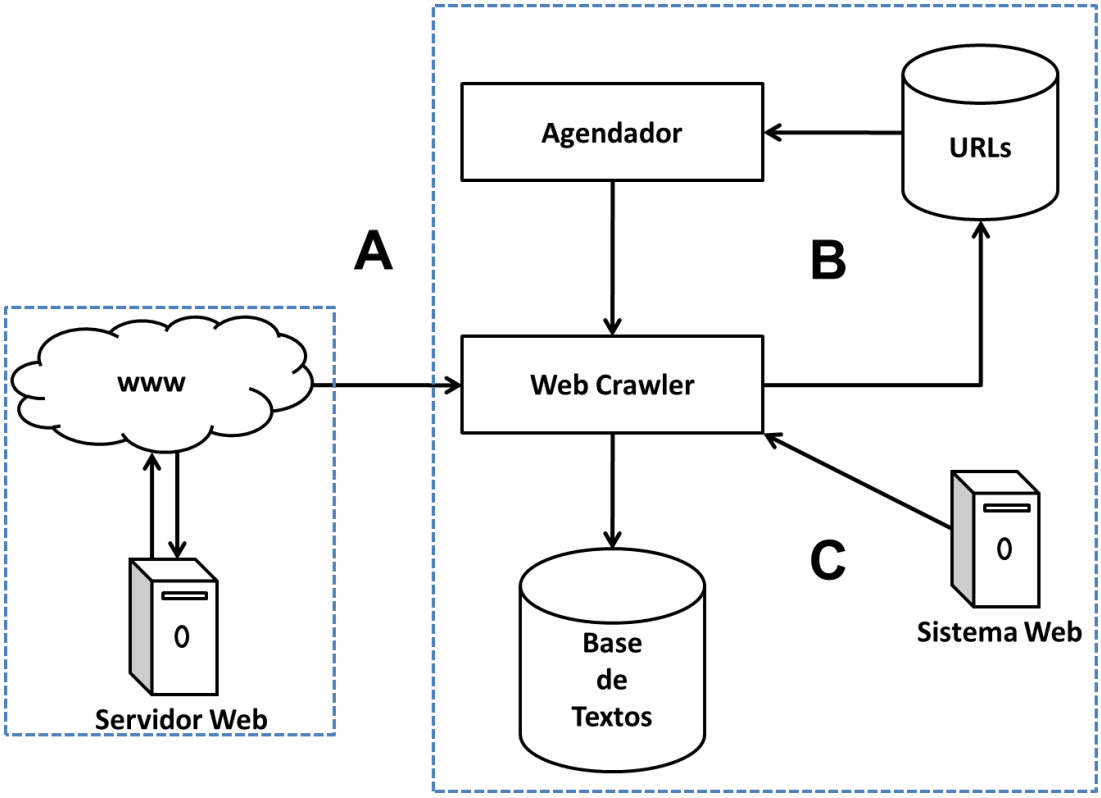

Figura 4: Arquitetura de um Web Crawler (Modificado CASTILHO & BAEZA-YATES, 2002 *apud* OLIVEIRA, 2008, p. 18).

De acordo com a Figura 4, os *web crawlers* iniciam a partir de um agendamento (Figura 4A). Neste agendamento é inserido um conjunto de URLs em uma fila, por onde todas estas URLs devem ser recuperadas, mantidas e priorizadas. A partir desta fila o *crawler* recupera a URL, faz o *download* da página que esta faz referência, extrai qualquer URL da página que foi baixada e coloca as novas URLs na fila (Figura 4B). Este processo é encerrado a partir do momento que o programa *crawler* encerra seu intervalo de frequência pré-determinado pelo mecanismo de busca. As páginas recuperadas são armazenadas em uma base de textos (Figura 4C), ficando disponíveis para o mecanismo de busca. Vale destacar que o processo inicial do *web crawler* é chamado de "sementes", que é uma lista de URLs a serem visitadas. Já o processo de visita do *crawler* às URLs é conhecido como fronteira de percorrimento. Critérios de seleção, revisita, cortesia e paralelização são obedecidos na visita recursiva às URLs que compõem a lista.

Segundo Silva (2006, p. 33), a política de seleção define quais páginas devem ser copiadas. É importante que o *crawler* recupere páginas relevantes, pois possui uma limitação quanto à quantidade de páginas que pode visitar. Para definir a relevância das páginas, faz-se necessária a adoção de métricas de importância para a priorização de páginas. Algumas das

métricas a serem consideradas estão relacionadas à popularidade de termos de *links* ou visitas a uma determinada página, bem como das URLs desta página.

Já a política de revisita, segundo Silva (2006, p. 33), é necessária devido à natureza dinâmica da Web, a qual enquanto o *crawler* percorre a Web, novos recursos são incluídos, bem como recursos existentes são atualizados e excluídos, ou seja, à medida que o *crawler* percorre a Web, os recursos existentes sofrem modificações. Devido a essa constante modificação de recursos da Web, são utilizadas métricas para indicar quão desatualizada está uma página do repositório do *crawler* e o quanto esta página do repositório é idêntica ou não à original. Essas métricas são chamadas de idade (*age*) e frescor (*freshness*), respectivamente (CHO & GARCIA-MOLINA, 2000, tradução nossa, p. 9).

Quando o *crawler* executa múltiplas requisições por segundo ou quando copia arquivos grandes, ocorre a sobrecarga no site que contem as páginas recuperadas pelo *crawler*. Neste momento, a política de cortesia é primordial, pois fornecerá indicadores necessários para evitar essa sobrecarga. Uma das alternativas para evitar o problema da sobrecarga é indicar quais partes dos servidores *Web* podem ser acessados. Essa indicação é realizada pelo protocolo de exclusão de robôs, que consiste em "programas que percorrem páginas web recursivamente recuperando o conteúdo das páginas" (SILVA, 2006, p. 29).

Por fim, para maximizar o número de recursos copiados e evitar que um recurso seja duplicado, a política de paralelização estabelece como coordenar múltiplos *clawers*. Cho (2002, tradução nossa, p. 1-3) propuseram duas políticas de paralelização: dinâmica e estática. Na política de paralelização dinâmica o servidor tem a responsabilidade de balancear a carga de cada *crawler*, pois um servidor central atribui novas URLs para diferentes *crawler* de forma dinâmica. Por outro lado, a política de paralelização estática utiliza uma regra fixa pré-estabelecida para definir como atribuir novas URLs aos *crawlers*.

Observa-se que os *crawlers* podem capturar informações de um ou mais servidores, sendo tais abordagens chamadas de *breadth-first* e *depth-first*, respectivamente. A forma mais simples dos *crawlers* iniciarem a captura de informações de uma página base e procurar *links* que direcionam para outras páginas é por meio da abordagem *depth-first*.

A próxima seção apresenta conceitos inerentes à próxima etapa de um sistema de RI, que é a Preparação de Documentos.

2.2.2 Preparação de Documentos

Uma vez que a fase de aquisição (seleção) dos documentos seja concluída, é essencial que estes documentos sejam tratados para, assim, serem processados e, posteriormente, indexados, armazenados e recuperados.

A fase de preparação de documentos tem como principal objetivo determinar quais termos do documento descrevem melhor o seu conteúdo, a fim de diminuir a complexidade da representação deste conteúdo. Tal preparação se deve ao fato de nem todas as palavras de um texto apresentarem realmente um significado relevante para representar a semântica de um documento, havendo no texto palavras que possuem mais significado do que outras.

A seleção de termos significativos para o conteúdo de um documento pode ser realizada tanto de forma manual, sendo executado por um especialista da área, quanto de forma automática, sendo esta seleção mais utilizada na maioria dos sistemas de RI.

Muitas são as operações realizadas sobre o conteúdo do documento para tratamento das informações nele contidas. Como algumas destas operações, destacam-se as principais existentes: *Case Folding*, *Stop Words* e *Stemming*, as quais serão apresentadas nas seções seguintes.

2.2.2.1. Case Folding

Segundo Silla & Kaestner (2002, p. 1), *case folding* é o procedimento de conversão de caracteres de um mesmo documento para um formato comum Este formato comum pode ser todo o texto em letras maiúsculas ou em letras minúsculas. A Figura 5 apresenta um exemplo de aplicação do *case folding*.

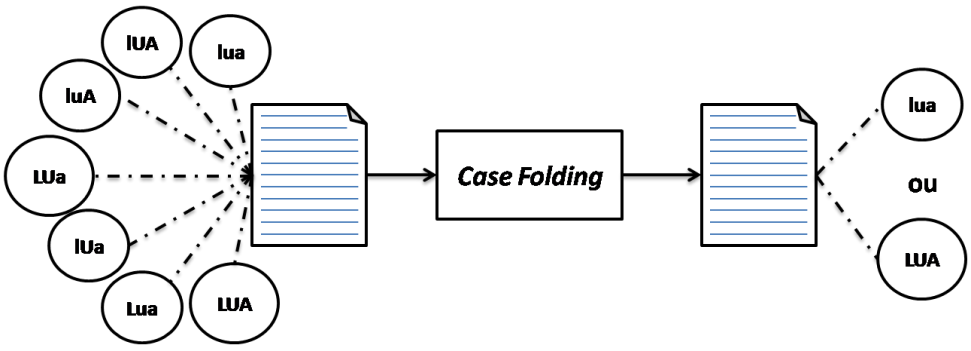

Figura 5: Exemplo de *Case Folding*.

Conforme a Figura 5, ao aplicar o *case folding* nas palavras “lua”, “lUA”, “luA”, “Lua”, “LUa“, “lUa”, “LUA”, poderiam todas ser transformadas para o formato comum em letras maiúsculas “LUA” ou em letras minúsculas “lua”. Desta forma, este processo padroniza as palavras para que estas possam ser entendidas como uma única palavra, fazendo com que o processo de comparação entre caracteres seja facilitado.

2.2.2.2. Stop Words

Stop words é o processo de remoção de termos que se repetem constantemente, como artigos, conjunções, preposições e pronomes. É a primeira identificação dos termos que serão descartados nos passos posteriores de processamento dos dados. Essa remoção é realizada para as palavras que aparecem em grande volume nos documentos, ou seja, palavras com baixo significado para corresponder à necessidade do usuário. Essas palavras são chamadas de *stop words* ou palavras de parada, o que permite um cálculo mais preciso da relevância das sentenças. Segundo Kongthon (2004, tradução nossa, p. 10-11) e Salton & McGill (1983, tradução nossa, p. 30), uma lista de *stop words* ou *stoplist*, que é uma lista de palavras que não possuem relevância para o documento, realiza a remoção de 40 a 50% do total de palavras de um texto.

2.2.2.3. Stemming

“*Stemming* é o processo de converter cada palavra para o seu radical, eliminando sufixos representados por flexões verbais e plurais” (SILLA & KAESTNER, 2002, p. 1-2). Assim, ao eliminar os sufixos das palavras, a quantidade de palavras diferentes a serem tratadas no texto será reduzida. A Figura 6 apresenta um exemplo de aplicação do processo *stemming*.

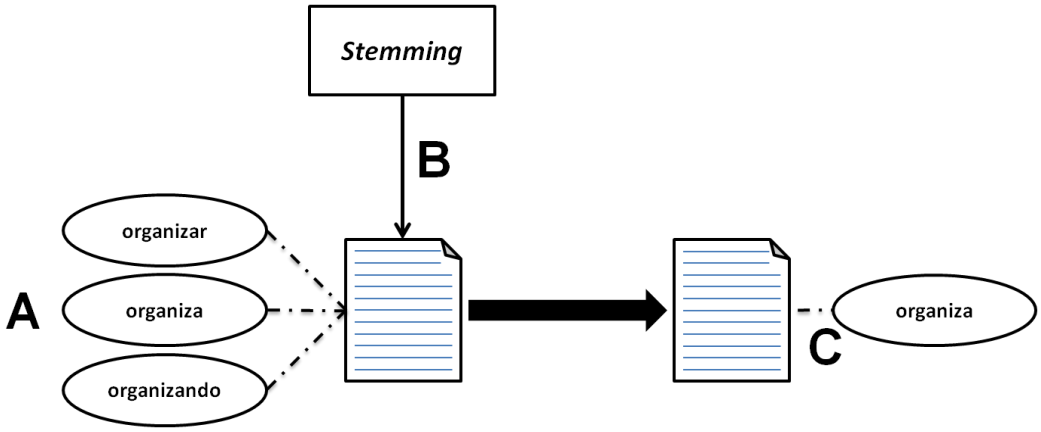

Figura 6: Exemplo de *Stemming*.

Como pode ser observado na Figura 6, tem-se um documento com as variações "organizar", "organiza" e "organizando" (Figura 6A). Ao aplicar o *stemming* neste documento (Figura 6B), o termo a ser considerado no mecanismo de busca seria o radical "organiza" (Figura 6C).

Os algoritmos de *stemming* são condicionados à linguagem por geralmente reunirem um conhecimento em lingüística abrangente. Segundo Mello (2007, p. 13), existem vários algoritmos de *stemming*, sendo os mais relevantes Método de Lovins, Método do Stemmer S, Método de Porter e Stemming RSLP, os quais serão brevemente apresentados a seguir.

- Método de Lovins – utiliza um algoritmo de combinação mais longa para remover cerca de 250 sufixos diferentes em um único processamento, removendo no máximo um sufixo por palavra. Embora vários sufixos não sejam removidos por este algoritmo, é considerado o mais agressivo dos algoritmos de *stemming* que serão apresentados, pelo fato da remoção acontecer em um único passo. Vale ressaltar que este algoritmo foi desenvolvido a partir de um conjunto exemplo de palavras na língua inglesa.
- Método do Stemmer S – reduz apenas alguns poucos sufixos do inglês, sendo eles "ies", "es" e "s". É considerado o algoritmo mais simples dos algoritmos de *stemming*. Apesar de não descobrir muitas variações, este algoritmo é utilizado pela sua característica conservadora.
- Método de Porter – considerado atualmente o algoritmo mais popular, remove sufixos a partir de critérios que não envolvem diretamente aspectos lingüísticos. Este algoritmo incide na assimilação e substituição das diversas inflexões e derivações de uma mesma palavra por um mesmo radical. Com a aplicação deste algoritmo, cerca de 60 sufixos diferentes são removidos.

- Stemming RSLP – chamado de Removedor de Sufixo da Língua Portuguesa (RSLP), remove sufixos da língua portuguesa a partir de regras.

Apesar de existirem vários algoritmos de *stemming*, de acordo com Mello (2007, p. 16), muitos desses algoritmos podem trazer alguns malefícios, sendo eles:

- *Under-stemming* – quando o algoritmo de *stemming* não remove um sufixo ou retira um sufixo menor do que poderia.
- *Over-stemming* – quando o algoritmo de *stemming* remove mais sufixo do que deveria. Quando isso acontece, conseqüentemente uma parte do radical é removida e uma nova palavra sem relação com o texto é gerada.
- *Mis-stemming* – quando o algoritmo de *stemming* remove uma parte da palavra ao interpretar como um sufixo e o trecho removido não é um sufixo.

Desta forma, assim que os documentos são selecionados e preparados, têm-se a etapa de indexação dos mesmos, a qual será apresentada na próxima seção.

2.2.3 Indexação de Documentos

Uma vez que a preparação de documentos tenha sido concluída, pode-se realizar o processo de indexação. Esta indexação garante que as informações estejam disponíveis aos usuários. Esse processo é feito a partir de um programa chamado indexador, o qual tem como funcionalidade extrair o conteúdo dos *links* descobertos e armazenar na base de dados informações como título, endereço e palavras-chave relacionadas àqueles recursos (PAIVA, 2007, p. 27). É nesta etapa da recuperação da informação que as estruturas de dados relacionadas aos textos disponíveis são criadas.

Segundo Baeza-Yates (1996, tradução nossa, p. 3), há três tipos de indexação:

- Indexação tradicional – os termos que caracterizam os documentos são determinados pelo usuário, sendo tais termos utilizados no índice de busca;
- Indexação do texto todo – o índice é constituído por todos os termos que compõem o documento, sem haver estrutura hierárquica entre tais termos;
- Indexação de *tags* – consiste na indexação das partes relevantes do documento, sendo tais partes selecionadas de forma automática para gerar as entradas no índice.

Para que seja possível organizar as páginas Web no processo de indexação, são retiradas algumas informações do corpo da página e de *metatags*[1], sendo elas (SILVA, 2006, p. 35-36):

- Conteúdo da página – informações mais concisas e texto completo;
- Descrição da página – informações sucintas que apresentam o conteúdo da página;
- Texto dos *hyperlinks* – disponibiliza índices semânticos do tópico para o qual apontam;
- Palavras-chave – informações que caracterizam o conteúdo de uma página;
- Textos destacados – textos que provavelmente possuem maior importância na descrição da página.

O procedimento de indexação é caracterizado pela constante consulta a um dicionário de termos. Este dicionário é conhecido como *Thesaurus*. O *Thesaurus* informa para os processos de indexação de documentos que termos índices devem ser utilizados para descrever cada conceito.

Thesaurus é "um dicionário de dados que correlacionam palavras diferentes e comuns a uma única palavra em todo o texto" (MELLO, 2007, p. 17). A idéia do dicionário de dados é fazer com que várias palavras sejam relacionadas para uma única palavra que possa substituí-las sem modificar o contexto. Ou seja, *thesaurus* é um dicionário que contém uma lista de termos sinônimos e/ou hierárquicos, que auxiliam o usuário a encontrar a informação desejada. Um exemplo típico desse algoritmo são as palavras "rua", "avenida" e "estrada", que poderiam ser reduzidas a uma única palavra como, por exemplo, "rua".

Segundo Fonsêca (2002, p. 22), a semântica entre os termos de um *thesaurus* está definida em três relacionamentos principais: o de equivalência, o de hierarquia e o de associação. Nos relacionamentos de equivalência são apresentados os termos diferentes que têm a mesma definição. Já o relacionamento hierárquico estabelece uma ligação vertical entre os termos que podem ser conceituados dentro de uma mesma divisão, onde uns são mais abrangentes e outros mais peculiares. Por fim, na associação são realizadas ligações entre os termos que representam conceitos únicos que estão, de certa forma, relacionados.

2.2.4 Armazenamento de Documentos

[1] *Metatags* são linhas de código HTML ou "etiquetas" que, dentre outras coisas, descrevem o conteúdo do site para os buscadores.

O armazenamento tem como principal objetivo utilizar repositórios para armazenar as páginas manipuladas. Segundo Arasu et al. (2001, tradução nossa, p. 16-17), esses repositórios devem possuir as seguintes características:

- Método duplo de acesso às informações armazenadasi – o armazenamento deve suportar dois diferentes modos de acesso com a mesma eficácia. O acesso randômico é utilizado para recuperar rapidamente uma página Web específica, atribuindo à página um identificador único. Esse tipo de acesso é utilizado pelo mecanismo de busca para apresentar cópias em *cache* para o usuário final. Já *Stemming* é o acesso usado para receber a coleção inteira, ou algum subconjunto significativo, como um fluxo de páginas. Esse tipo de acesso é usado pelo indexador e pelos módulos de análise para processar e analisar páginas em grande volume.
- Manipulação de grande volume de atualizações – esses repositórios devem ser capazes de adicionar, atualizar e reorganizar facilmente informações enviadas por *crawlers*. Como o *crawler* recebe novas versões de páginas Web, o espaço ocupado por versões antigas deve ser recuperado por meio da compactação e reorganização desse espaço.
- Controle de páginas obsoletas – deve ser capaz de identificar e retirar páginas que não são utilizadas. Essa necessidade se dá devido ao fato de não haver notificação ao repositório quando uma página Web é removida, o que acontece na maioria dos sistemas de arquivos ou dados, onde há a exclusão explícita de objetos não necessários.

Após os documentos terem sido selecionados, tratados, indexados e armazenados, estes estão prontos para serem recuperados a partir de consultas formalizadas por usuários.

2.2.5 Recuperação de Documentos

Após a análise do conteúdo de cada página armazenada no repositório e a criação de um conjunto de palavras-chave (índice) que identifica o conteúdo da página e associa a URL na qual cada palavra-chave ocorre, finalmente, inicia-se o módulo de consulta e ordenação. Este módulo está diretamente relacionado com o modo que as páginas foram indexadas, devido ao fato de nem todos os tipos de busca poderem ser usados em qualquer sistema.

O módulo de busca e ordenação inicia a partir de requisições de consultas dos usuários. A partir de tais requisições, o módulo de busca processa tais requisições, retornando, de maneira ordenada, pelo módulo de ordenação, os documentos que melhor atendem essas.

Uma consulta pode ser baseada em palavras-chave, casamento de padrão ou estruturada, as quais serão apresentadas de forma detalhada a seguir (RODRIGUES, 2009, p. 22-23).

- Consulta baseada em palavras-chave – consiste em classificar as respostas segundo a função de relevância, seguida pelo mecanismo de busca. Pode ser estabelecida a partir de palavras isoladas, fundamentada no contexto de documentos sobre o tema da consulta ou com junções *booleanas*. A idéia deste tipo de consulta é recuperar todos os documentos que contêm ao menos uma palavra da consulta e, em seguida, ordenar os documentos recuperados para que possam ser retornados ao usuário.
- Consulta baseada no casamento de padrão – tem como objetivo encontrar documentos que apresentam segmentos de texto que casam com o padrão da consulta. Para realizar este tipo de busca, tal busca pode ter um padrão simples ou complexo. O padrão simples diz respeito a apenas uma palavra, um sufixo, substring ou intervalo. Já o padrão complexo pode ser uma expressão regular.
- Consulta baseada na estrutura – permite que sejam realizadas buscas a campos específicos das páginas. Esse tipo de consulta permite que um usuário busque por uma palavra apenas nos títulos dos documentos, por exemplo.

As expressões de consultas podem ser compostas por operadores lógicos, como: “e”, “ou” e “não”. O processo de busca é apresentado a seguir:

1. O usuário formula uma pesquisa através da interface do sistema;
2. A consulta é encaminhada para o sistema;
3. O sistema analisa as palavras ou expressões fornecidas pelo usuário e as compara com os índices.

Antes que os documentos selecionados no processo de recuperação da informação sejam apresentados para o usuário, estes passam por um processo de ordenação. A ordenação desses resultados obedece critério de relevância de cada item retornado em relação à consulta submetida. A frequência dos termos e a proximidade entre os mesmos são fatores utilizados para determinar a relevância, que diz respeito à medida de quão bem um objeto atende à expectativa do usuário. Outra forma de determinar a relevância de um documento é a partir de cálculos realizados a partir de um modelo de representação de documentos e consultas, sendo que esses modelos foram apresentados na Seção 2.2. O processo de ordenação é estruturado da seguinte forma:

1. Após realizar a pesquisa em sua base de dados, o sistema identifica a relevância de cada um dos documentos retornados;

2. Identificada a relevância de cada um dos documentos, o sistema prioriza os documentos que apresentaram maior relevância para a consulta, até chegar ao menos relevante, finalizando assim o processo de ordenação;
3. O sistema retorna os documentos para sua interface, possibilitando que o usuário tenha acesso a tais documentos.

2.3 Clustering de Documentos

Clustering é um método de descoberta de conhecimento que identifica agregações ou relações entre objetos, sendo um método útil para o agrupamento de documentos similares (WIVES, 2004, p. 27). Esses objetos similares são distribuídos em grupos relativamente homogêneos, conhecidos como *clusters* (JAIN, MURTY & FLYNN, 1999, tradução nossa, p. 306).

A preparação e a transformação dos objetos são realizadas a partir de técnicas de seleção e construção de atributos. Em seguida, determinam-se as métricas de similaridade a serem utilizadas para o agrupamento de dados e se submete o conjunto de exemplo a um algoritmo. Por fim, analisa-se o grau de significância dos resultados obtidos a partir do algoritmo de *clustering*. O grau de significância dos resultados obtidos indica a medida de importância relativa de cada tema ou palavra-chave do documento. A presença de um especialista da área do domínio onde o sistema é construído é de grande importância nessa etapa de análise para evitar que o *clustering* obtido produzida resultados de má qualidade ao usuário final. . Esse especialista será responsável por verificar quais dos termos retornados possui relevância para a classificação dos objetos. Quando a análise do grau de significância não é realizada pelo especialista, o próprio usuário final exercerá o papel de avaliador, verificando quanto o resultado apresentado do *clustering* é relevante para suas necessidades. A Figura 7 apresenta o processo de *clustering* de documentos.

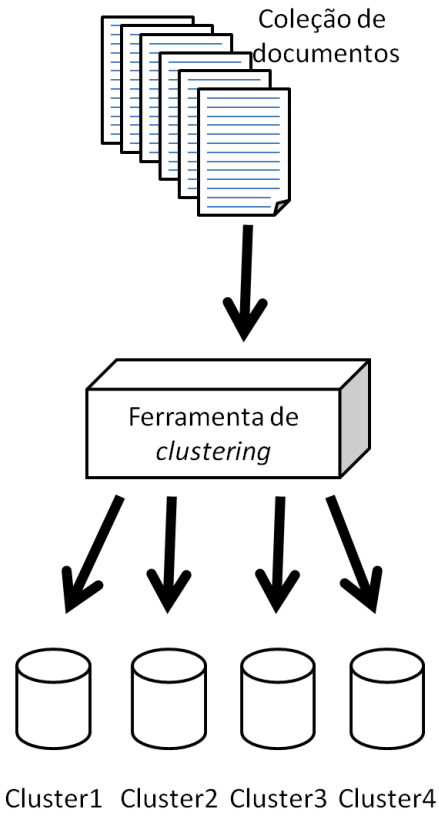

Figura 7: Processo básico de *Clustering* (Modificado LOPES, 2004, p. 49).

Como observado na Figura 7, o processo básico de *clustering* se dá a partir da criação de uma descrição simplificada do documento para cada texto a ser adicionado aos *clusters*. Essa descrição é normalmente representada por um vetor de características, ou uma lista de temas dominantes ou palavras-chave e uma medida de importância relativa de cada tema ou palavra-chave do documento (HATZIVASSILOGLOU, GRAVANO & MAGANTI, 2000, tradução nossa, p. 1). É a partir do vetor de características que a distância entre os documentos é calculada, podendo tais documentos serem agrupados logo em seguida. O cálculo de distância entre documentos pode ser realizado através da medida de similaridade padrão ou conceitual. A medida de similaridade padrão consiste em métricas de distâncias convencionais já conhecidas. Por outro lado, a medida de similaridade conceitual se dá a partir da função de distância entre tópicos na hierarquia de assuntos e os pesos desses tópicos nos documentos, medindo, desta forma, a distância em uma hierarquia de assunto. A Figura 8 apresenta um exemplo de cálculo de grau de similaridade.

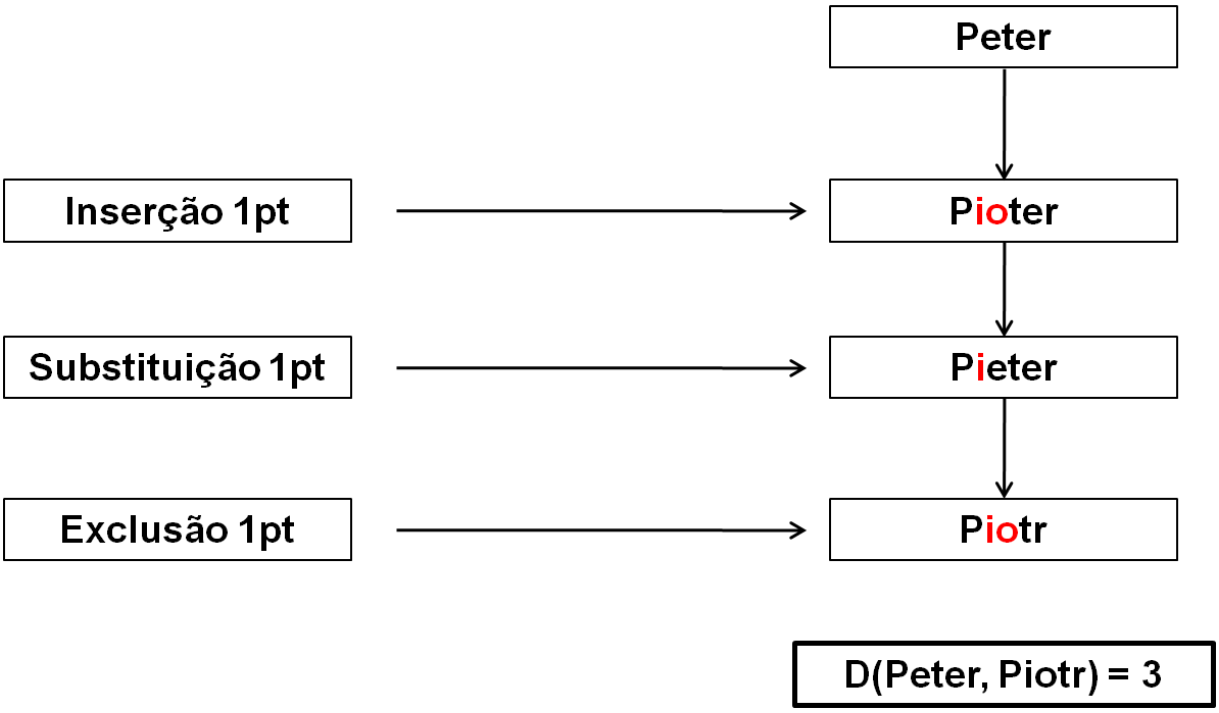

Figura 8: Exemplo de Cálculo de Grau de Similaridade.

Como observado na Figura 8, o exemplo mostra o cálculo de grau de similaridade entre “Peter” e “Piotr”. Inicialmente, pelo processo de inserção, a letra “i” é adicionada entre as letras “P” e “e”. Utilizando a substituição de caracteres, no lugar da terceira letra “e” aparece a letra “o”. Por último, o quinto caractere “e” é excluído da palavra. Após as três etapas, sendo cada etapa contabilizada com uma unidade, a similaridade entre as palavras “Peter” e “Piotr” é resultante do cálculo da distância entre a estrutura das duas palavras, neste caso, igual a três.

Segundo Wives (2004, p. 28), o processo de agrupamento de documentos similares é empregado a partir de dados não rotulados (*unlabeled data*), isto é, não conhecidos e sem modelos estatísticos que os descrevam, a fim de obter um maior conhecimento sobre tais dados e suas relações. Sendo assim, uma coleção de padrões desconhecidos (não classificados) é agrupada em *clusters* que possuem algum significado para o usuário. Ao contrário da classificação propriamente dita, não há a possibilidade de saber se o processamento está sendo feito de forma apropriada ou não, uma vez que neste tipo de análise não supervisionada não existe a possibilidade de estabelecer uma comparação entre os resultados e os modelos conhecidos. Desta forma, determina-se as métricas de similaridade que servirão como base para o agrupamento de dados. A partir de então, é obtida a coleção de padrões desconhecidos e agrupada em *clusters* que apresentam algum significado para o

usuário, ou seja, estabelece-se o agrupamento de documentos similares. Uma vez realizado o agrupamento de documentos, segue-se para a análise do grau de significância dos resultados obtidos a partir do algoritmo de *clustering*.

Com o intuito de facilitar a definição de grupos, o processo de agrupamento é na maioria das vezes realizado antes de um processo de classificação. Desta forma, as relações entre os objetos podem ser analisadas por um especialista, assim como também a melhor distribuição ou configuração de classes para os objetos pode ser identificada.

Uma vez identificada a similaridade entre os objetos, estes são atribuídos a um *cluster* de objetos que possuem alguma relação de similaridade. Esta similaridade é obtida através da análise de todas as características que descrevem os objetos. A partir desta análise, são identificas as relações de interdependência entre os objetos, sendo estes classificados em um mesmo grupo.

De forma genérica, o processo de agrupamento possui as seguintes etapas (JAIN, MURTY & FLYNN, 1999, tradução nossa, p. 306):

- representação dos padrões, que se dá a partir da extração e ou seleção de características;
- definição de uma medida de distância apropriada ao domínio e ao esquema de representação dos dados;
- descoberta ou identificação de *clusters*;
- abstração dos dados, a fim de representar o conjunto de padrões pertencentes a cada *cluster* de forma simples e compacta;
- validação ou avaliação do resultado.

Desse modo, levando em consideração as etapas do processo de agrupamento apresentadas, as próximas seções apresentarão, de forma detalhada, cada uma dessas etapas.

2.3.1 Representação dos Padrões (Objetos)

A representação dos padrões (objetos) se dá a partir da descrição ou representação do objeto por algum modelo. Tal descrição ou representação do objeto é de grande importância para que haja a possibilidade de realizar uma análise ou um processamento do objeto. A descrição do objeto diz respeito ao seu tamanho, tipo, quantidade, forma e características disponíveis. No caso de documentos textuais, os objetos são geralmente representados por vetores de palavras, sendo tais palavras identificadas a partir da seleção ou extração de características. Esse vetor de palavras é composto por características do objeto, podendo essas características ser do tipo

numérico (inteiro, real) e categórico (booleano, conjunto de valores). A Figura 9 apresenta um exemplo de representação dos objetos.

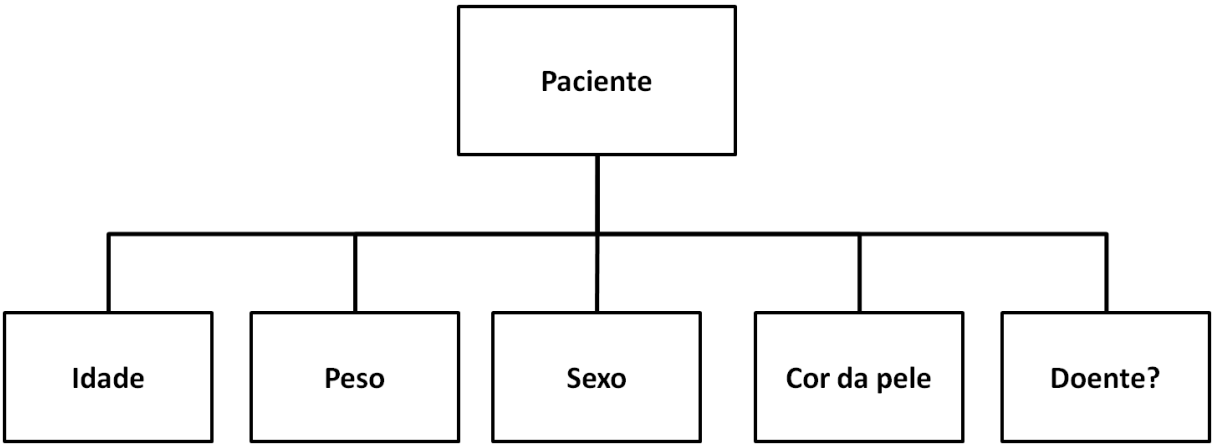

Figura 9: Representação dos objetos.

Como observado na Figura 9, tem-se como contexto uma amostra de dados clínicos, tendo como objeto "Paciente". Ao representar esse objeto por um vetor de características, foram selecionadas: idade, característica numérica (inteira); peso, característica numérica (real); sexo, característica categórica (masculino, feminino); cor da pele, característica categórica (branca, marrom, amarela, preta); doente, característica booleana (sim, não).

A seleção de características tem como objetivo identificar um subconjunto de palavras mais adequado no que tange a representatividade, efetividade e relevância dessas palavras dentro do contexto do documento. Basicamente, a seleção de características diminui o conjunto de palavras utilizado para representar um documento, utilizando somente algumas palavras ao invés de todas.

Segundo Wives (2004, p. 33), na extração de características (*feature extraction*), novos descritores para os documentos são criados tendo como base as palavras neles contidas. Neste contexto, o conjunto inicial de palavras passa por um processo de transformação, objetivando produzir novas características mais descritivas e discriminantes, melhorando desta forma a representação do documento.

2.3.2 Mediação da Proximidade entre Padrões

Os documentos textuais a serem agrupados são descritos por características, sendo tais características tanto qualitativas quanto quantitativas (WIVES, 2004, p. 34). As características

qualitativas constituem valores nominais ou não ordenados como, por exemplo, cores e palavras; ou ordinais como, por exemplo, frio ou quente. Já as características quantitativas correspondem a valores numéricos contínuos, discretos ou intervalares como, por exemplo, peso, tempo e idade.

Desta forma, as medidas de proximidade entre os objetos devem ser apropriadas ao modelo escolhido, uma vez que dependendo do tipo de objeto a ser manejado, este apresenta uma grande variedade e variabilidade de tipos de características.

A medida de proximidade entre objetos pode ser avaliado por meio de quatro grupos:

- Medidas de distância – a similaridade é determinada pela proximidade dos objetos em um espaço euclidiano, onde cada ordenada corresponde a uma característica.
- Coeficientes de correlação – a similaridade é obtida através da identificação de correlações entre variáveis.
- Coeficientes de associação – a similaridade é estabelecida por variáveis binárias, isto é, variáveis que possuem apenas dois estados ou valores. Os estados de uma variável binária podem ser exemplificados por zero ou um, presentes ou ausentes, ativas ou inativas.
- Medidas probabilísticas de similaridade – a similaridade é constituída pela média realizada diretamente no dado bruto, sendo representado em valores binários. O objetivo da medida probabilística de similaridade é analisar o ganho de informação que a combinação de dois ou mais objetos proporciona e reunir aqueles que apresentam o maior ganho.

2.3.3 Identificação de Clusters

Considerada a principal etapa do processo de descoberta e análise de *clusters*, a identificação de *clusters* tem como objetivo identificar grupos de objetos similares (WIVES, 2004, p. 39). Existem diversas técnicas de identificação de *clusters*, que se diferem na forma como os grupos de documentos são visualizados e na precisão com que os grupos são definidos.

Como uma das principais técnicas de *clustering*, tem-se o *clustering* hierárquico. As técnicas hierárquicas de *clustering* produzem uma seqüência aninhada de partições, de forma a juntar os objetos em *clusters* cada vez maiores, incluindo elementos e os próprios *clusters* já identificados. Graficamente, o resultado final deste algoritmo de agrupamento hierárquico pode ser representado por uma árvore. Segundo Wives (2004, p. 45), esta representação é

denominada dendograma (*dendrogram*), a qual pode ser vista na **Erro! Fonte de referência não encontrada.**.

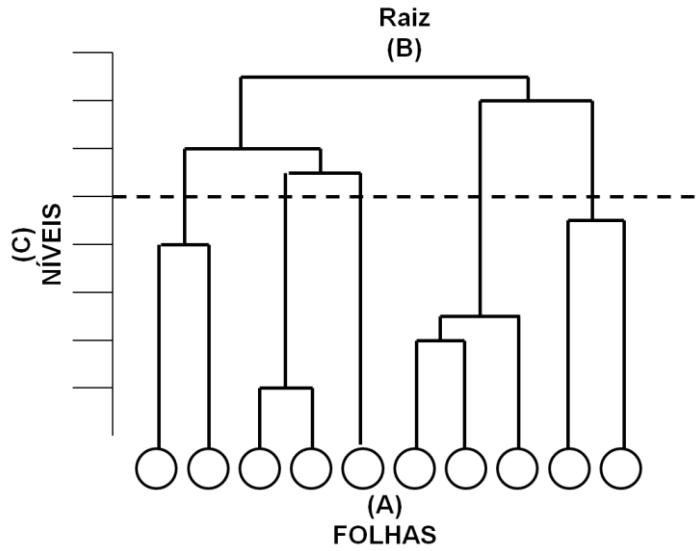

Figura 10: Representação de um dendograma ou árvore de classificação.

No dendograma representado na Figura 10, as folhas (Figura 10A) representam elementos completamente isolados. Ao caminhar em direção à raiz (Figura 10B), cada bifurcação representa uma união entre elementos. Cada nível do dendograma (Figura 10C) descreve o grau de similaridades, ou seja, indica em que nível de similaridade os elementos foram reunidos.

Segundo Coelho (2010, p. 45) os métodos de *clustering* hierárquico podem ser agrupados em duas classes:

- **Método Ascendente ou Aglomerativo (*bottom-up*) –** inicia-se com n *clusters* e se agrupam consecutivamente até obter um único *cluster*. A Figura 11 apresenta um exemplo do método ascendente.

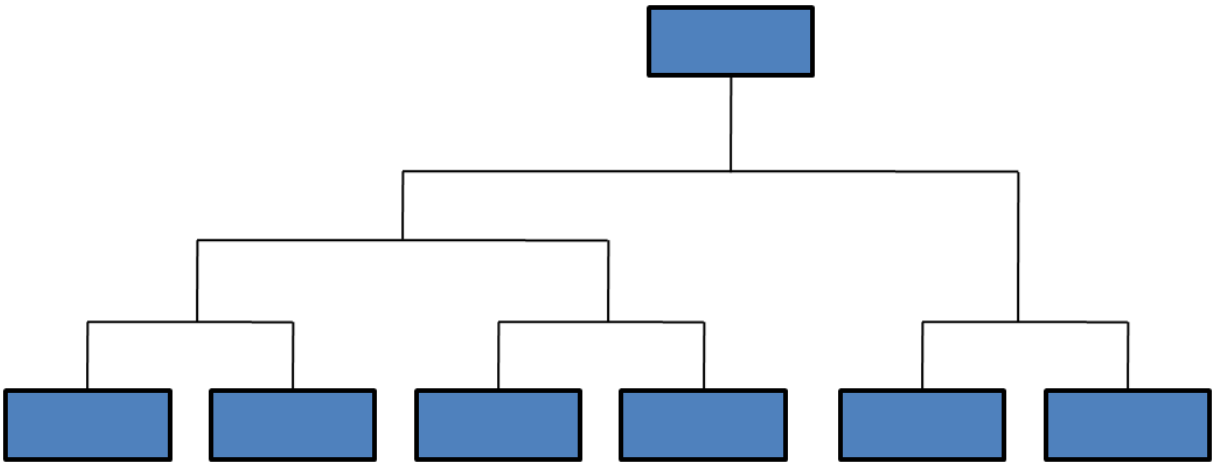

Figura 11: Exemplo do Método Ascendente ou Aglomerativo (*bottom-up*).

Como observado na Figura 11, o método ascendente se resume em atribuir um padrão por *cluster* (n *clusters*), encontrar o par de *clusters* mais semelhantes, juntar os dois objetos em um único *cluster* e, se o número de *clusters* for maior que um, repete-se o ciclo a partir da busca pelo par de *clusters* mais semelhantes.

- **Método Descendente ou Divisivo (*top-down*) –** inicia-se com um *cluster* obtendo todos os dados e se dividem continuamente até obter n *clusters*. A Figura 12 apresenta um exemplo do método descendente.

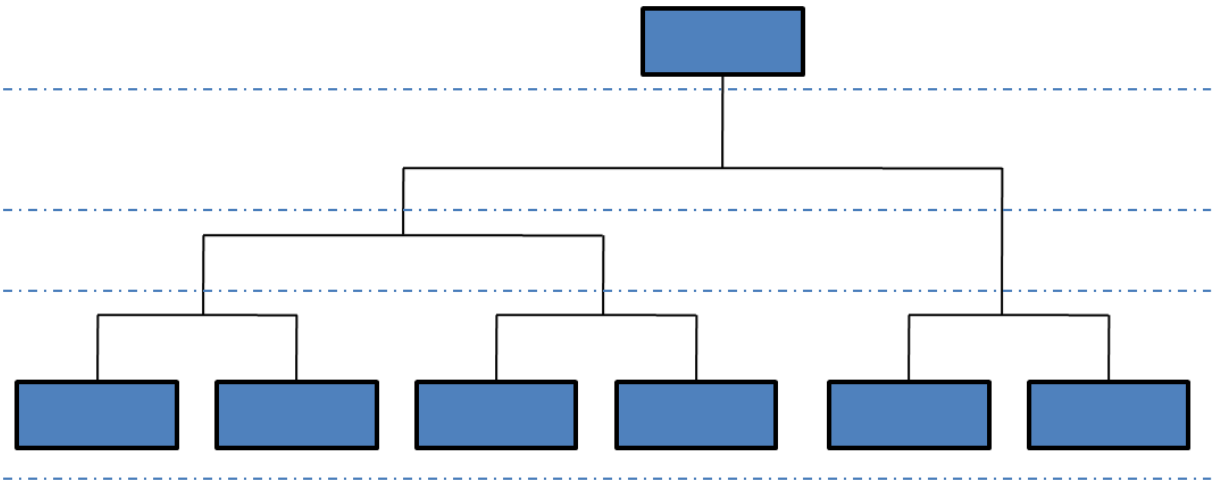

Figura 12: Exemplo do Método Descendente ou Divisivo (*top-down*).

Como observado na Figura 12, o método descendente se resume em atribuir todos os objetos (n) a um *cluster*, encontrar o "pior" *cluster*, dividi-lo em dois e, se o número de *clusters* for menor que os objetos padrões, repete-se o ciclo a partir da busca pelo "pior"

cluster. Vale ressaltar que o "pior" *cluster* se refere ao maior número de amostras, maior variância, dentre outros.

Outra das principais técnicas de *clustering* são os algoritmos particionais. Esses algoritmos dividem o conjunto de dados, ou seja, criam aglomerados, fazendo várias iterações nesse conjunto. O algoritmo mais conhecido é o *k-means*.

Segundo Wives (2004, p. 46), no *k-means* "o usuário indica o número de conglomerados desejado e o algoritmo de particionamento cria (de forma aleatória ou por outro processo) um conjunto inicial de partições (conglomerados)". A Figura 13 apresenta o algoritmo particional *k-means*.

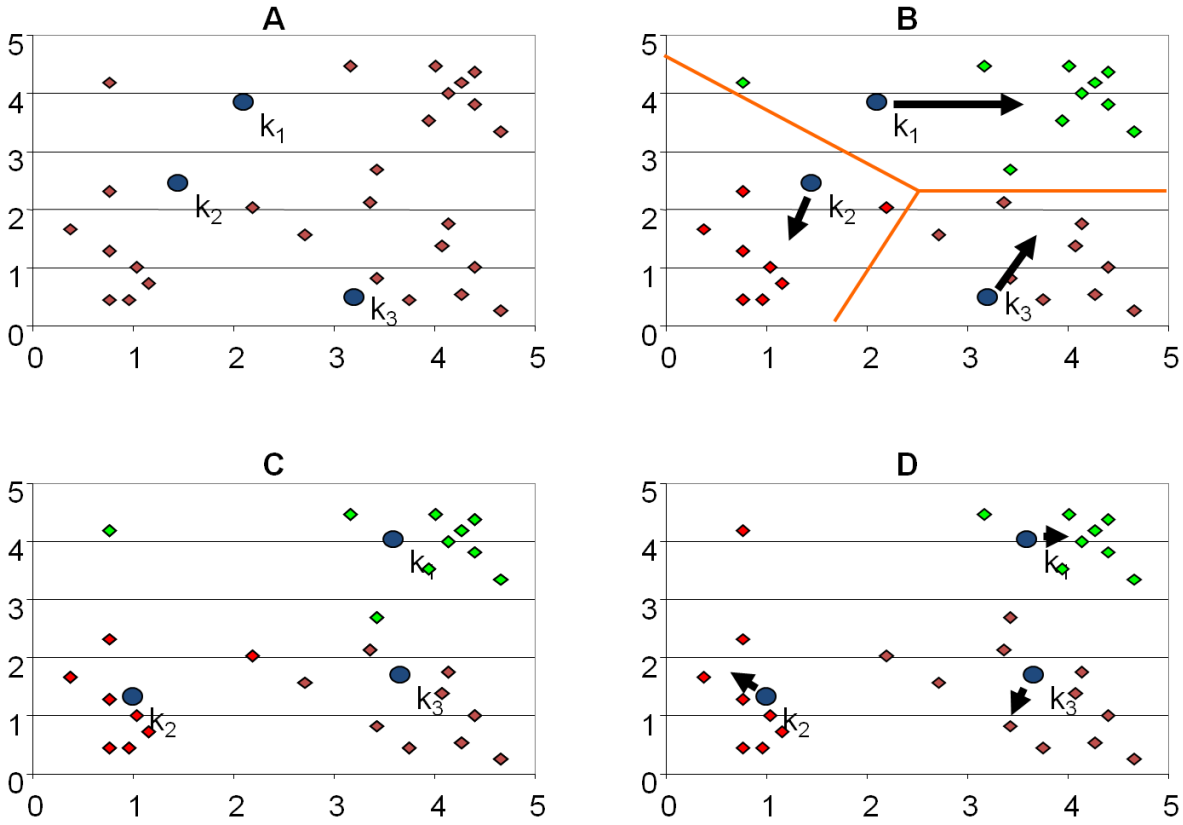

Figura 13: Algoritmo particional *k-means*.

Como observado na Figura 13, adicionar explicação feita na apresentação para a banca. A forma como o *k-means* trabalha é representada na Figura 14.

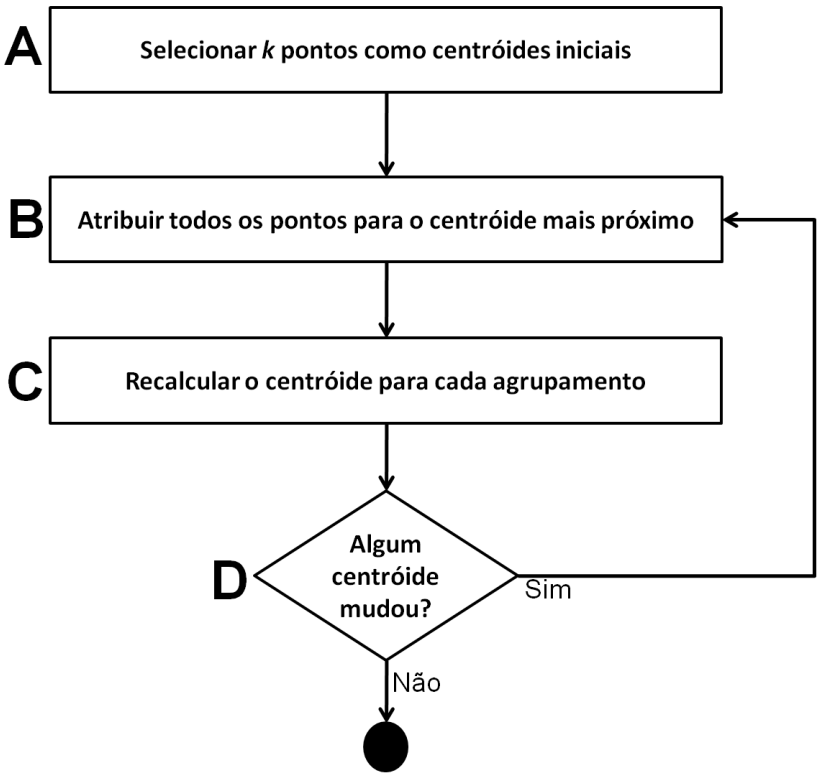

Figura 14: Funcionamento do algoritmo *clustering k-means.*

Como pode ser observado na Figura 14, o algoritmo *k-means* inicia seu processamento a partir da seleção de um conjunto de características mais representativas de cada um dos conglomerados, denominados centróides (Figura 14A). O centróide é obtido pela média de todos os vetores do *cluster*. A partir de então o algoritmo analisa a distância ou similaridade dos pontos selecionados com todos os objetos a serem agrupados. Assim, cada objeto é alocado ao conglomerado de cujo centróide estiver mais próximo (Figura 14B). Atribuído todos os pontos para o centróide mais próximo, este centróide é recalculado para refletir (e representar) esse novo cluster (Figura 14C). Enquanto os centróides não mudarem de posição (verificação representada na Figura 14D), o processo é repetido (JAIN, MURTY & FLYNN, 1999, tradução nossa, p. 279).

O fato dos métodos de particionamento fazerem várias passagens (iterações) no conjunto de dados, é considerado sua maior vantagem. Tal vantagem se dá ao fato da possibilidade de correção de eventuais problemas de alocação inadequada, muito comum nos algoritmos hierárquicos por agrupamento.

Clustering k-means biseccionado é uma variante do *k-means*. Este algoritmo inicia com um único *cluster* de todos os documentos e trabalha da forma apresentada na Figura 15.

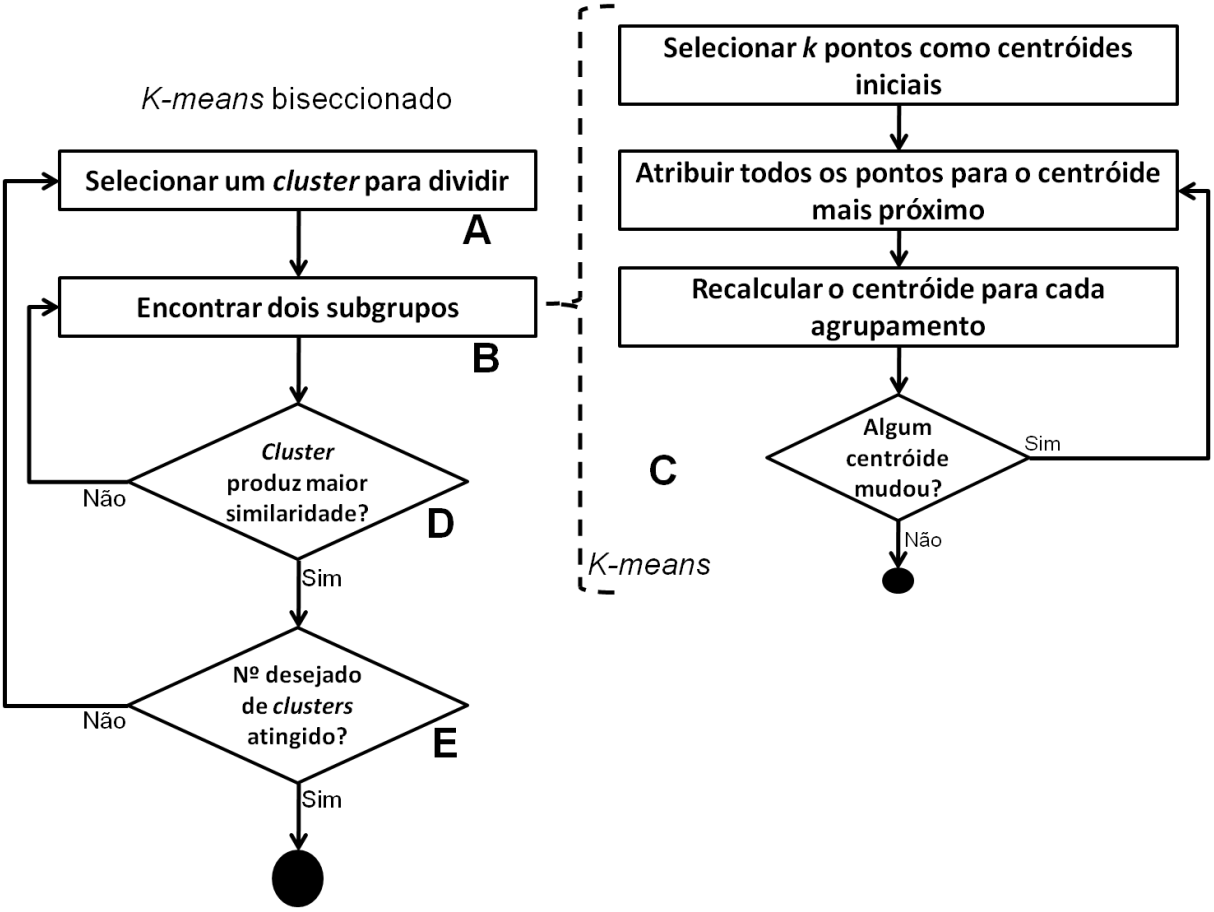

Figura 15: Funcionamento do algoritmo *clustering k-means* biseccionado.

O *k-means* biseccionado inicia com um único *cluster* de todos os documentos e trabalha da forma como é apresentada na Figura 15. Como observado na Figura 15, nota-se que um *cluster* deve ser selecionado para ser dividido (Figura 15A). Em seguida, executa-se a técnica tradicional de *k-means* (Figura 15C) para cada um dos dois subgrupos encontrados (Figura 15B). Este processo é executado até que se obtenha o *cluster* de elementos de maior similaridade (Figura 15D). Obtendo-se o *cluster* de maior similaridade, o algoritmo verifica se a quantidade de agrupamentos desejada foi alcançada (Figura 15E). Caso tenha sido atingido, o processamento do algoritmo é encerrado. Caso contrário, repete-se os passos executados pelo algoritmo desde o início (Figura 15A).

Há diversas formas diferentes de selecionar qual *cluster* será dividido. Por exemplo, pode-se escolher o maior *cluster* em cada etapa, o *cluster* com similaridade geral ou utilizar o critério baseado em ambos descritos anteriormente, ou seja, baseado tanto no tamanho quanto na similaridade geral. Segundo Steinbach, Karypis & Kumar (2000, tradução nossa, p. 11), a diferença entre as formas de selecionar o *cluster* a ser dividido é pequena.

Por fim, observa-se que o *k-means* biseccionado apresenta uma complexidade de tempo que é linear ao número de documentos. Steinbach, Karypis & Kumar (2000, tradução nossa, p. 13) aponta que se o número de *clusters* é amplo e o refinamento não é utilizado, então *k-means* biseccionado é mais eficiente que o algoritmo *k-means* tradicional. Ainda segundo os autores, essa maior eficiência se dá ao fato de no *k-means* biseccionado não possuir a necessidade de comparar cada ponto a cada centróide, mas sim aos pontos do *cluster* e sua distância até os dois pontos centrais. Ao realizar um paralelo entre o *clustering* hierárquico e o *k-means* biseccionado, nota-se que ambos analisam os pontos do *cluster*.

Como a terceira das principais técnicas de *clustering*, tem-se o algoritmo *density-based*. Segundo Wiley & Sons (2000, tradução nossa), o *density-based* é baseado em um critério local, tais como pontos de ligações de densidade. Ainda segundo o autor, as principais características deste algoritmo dizem respeito à capacidade de encontrar *clusters* de forma arbitrária e lidar com ruídos. Sendo assim, necessita de apenas uma varredura por meio do banco de dados e de parâmetros de densidade para estabelecer o critério de finalização. Segundo Wives (2004, p. 40), os parâmetros de densidade "define o conglomerado como sendo uma densa agregação de pontos no espaço quando comparado a outras áreas que possuam poucos pontos ou nenhum. Pode ser compreendida como a quantidade de pontos do conglomerado".

Density-based Spatial Clustering of Applications with Noise (DBSCAN) encontra *clusters* de forma arbitrária em bases de dados espaciais na presença de ruídos com o intuito de determinar o conjunto de pontos de máxima densidade de ligações. Os *clusters* são considerados regiões mais densas dos objetos espaciais de dados, sendo separados por regiões de baixa densidade ou ruído.

A técnica *density-based* está representada na Figura 16.

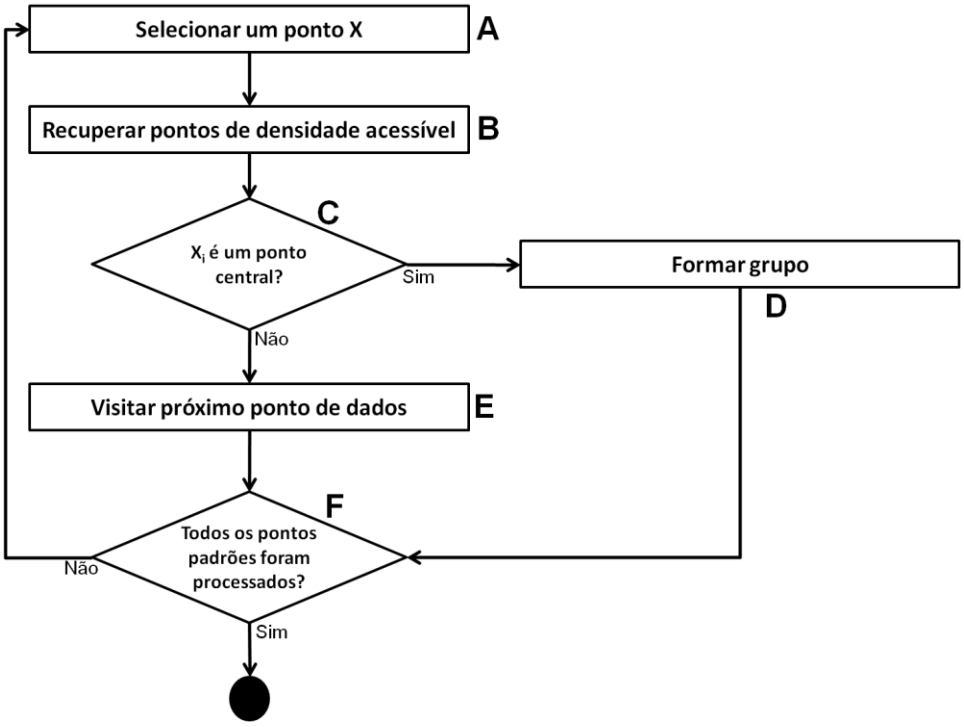

Figura 16: Funcionamento do algoritmo *Density-based*.

Como pode ser observado na Figura 16, o algoritmo *density-based* inicia seu processamento a partir da seleção arbitrária de um ponto X (Figura 16A). Em seguida, todos os pontos de densidade acessível são recuperados (Figura 16B) e é feita a verificação se o ponto X_i é um ponto central ou da fronteira (Figura 16C). Caso X_i seja um ponto central, forma-se um grupo (Figura 16D). Caso X_i seja um ponto da fronteira, nenhum ponto é acessível a partir da densidade X. Sendo assim, o DBSCAN visita o próximo ponto de dados do banco de dados (Figura 16E). Até que todos os objetos de dados (pontos padrão) tenham sido processados (Figura 16F) o algoritmo reinicia seu processamento (Figura 16A). vale ressaltar que o DBSCAN é considerado um método eficaz no que tange à descoberta de *clusters*, mesmo para grandes bases de dados espaciais.

Por fim, a última das principais técnicas de *clustering* é o algoritmo *model-based*, baseado no modelo de *clustering* hierárquico. Esse algoritmo tem como objetivo otimizar a correlação entre os dados de entrada com alguma função matemática. Os dados muitas vezes são assumidos para serem gerados a partir de distribuições de probabilidade *k*. O algoritmo *model-based* pode ser considerado como a versão probabilística do *clustering k-means*. No entanto, uma de suas necessidades é encontrar os parâmetros de distribuição (WILEY & SONS, 2000, tradução nossa). Ao aplicar o algoritmo *model-based*, o resultado obtido pelo

clustering é considerado razoavelmente bom, mesmo quando sua inicialização se dá sem algum tipo de informação referente aos *clusters.*

Expectation Maximization (EM) é um dos mais conhecidos algoritmos de refinamento iterativo que pertence à categoria de *cluster model-based*. A diferença entre o algoritmo tradicional *k-means* é que cada ponto padrão pertence a um *cluster* conforme algum peso, ou seja, probabilidade de adesão. Desta forma, pode-se dizer que o *model-based* fornece um modelo estatístico de dados capaz de lidar com incertezas associadas.

2.3.4 Abstração de Dados e Compreensão dos Clusters

Segundo Wives (2004, p. 50), "Abstração de Dados e Compreensão dos Clusters consiste na extração de uma representação simplificada e compacta de um *cluster*, a fim de descrevê-lo ou representá-lo". A criação de um modelo de representação de *clusters* intuitivo e compreensível dá ao usuário a possibilidade de ter maior facilidade em compreender o *cluster*. Além disso, a representação de *clusters* permite que o *cluster* seja analisado automaticamente por outro processo computacional, pelo fato de existir uma representação que prezou pela interoperabilidade e processamento eficiente.

A representação de *clusters* pode ser feita através de centróides ou de protótipos. Como dito anteriormente, o centroide, neste caso, consiste na média de todos os vetores de um *cluster*, sendo tais vetores compostos por palavras, pelo fato do *cluster* tratar da representação de documentos. O protótipo possui o mesmo objetivo do centróide, diferenciando-se pelo fato de abranger qualquer modelo de representação, inclusive o próprio centróide. Apesar de serem tratados como duas formas distintas de representação de *clusters*, vários pesquisadores consideram os dois termos como sinônimos (WIVES, 2004, p. 50).

Existem diversas medidas de seleção de características, que podem ser utilizadas no auxílio para criar os centróides ou protótipos, através da identificação das características do *cluster* consideradas mais relevantes. Dentre as diversas medidas de seleção de características, têm-se o coeficiente X^2 de Schtutze, o coeficiente de correlação de Hwee Ng e o escore de relevância de Wiener, as quais serão detalhadas a seguir (WIVES, 2004, p. 50-51):

- Coeficiente X^2 – este coeficiente consiste na redução de dimensionalidade, por meio da aplicação de uma medida de dependência estatística denominada X^2. Sendo assim, a medida é determinada pelo número de documentos relevantes (pertencentes a uma categoria) e não relevantes (não pertencentes a uma categoria) em que um termo aparece e no número de documentos relevantes e não relevantes em que o mesmo termo não aparece

(WIVES, 2004, p. 51). Schutze et al. (1995, tradução nossa, p. 2) desenvolveu essa técnica a partir do desejo que possuía em selecionar as palavras mais relevantes para seus documentos.

- Coeficiente de Correlação – indica o grau de semelhança entre uma palavra e um documento. Sendo assim, esse coeficiente leva em consideração a quantidade total de documentos de uma coleção, a quantidade de documentos em que a palavra aparece e a quantidade de documentos em que a palavra não aparece (WIVES, 2004, p. 51).

Escore de Relevância – tem como objetivo construir protótipos capazes de representar categorias de documentos em sistemas de classificação de textos. Seu funcionamento se dá a partir do número de documentos da categoria que contem o termo que está sendo analisado, do número total de documentos da categoria, do número de documentos fora da categoria que contem o termo que está sendo analisado e o número total de documentos da coleção que está sendo analisada. A partir do escore obtido, atribui-se pontos positivos para as palavras exclusivas dos documentos pertencentes a uma categoria e pontos negativos para as palavras de documentos de outras categorias. Essa atribuição de valores é executada até que todas as categorias existentes sejam analisadas. No final do processo, as palavras que possuem valores mais altos são classificadas como mais adequadas para representar as categorias analisadas (WIVES, 2004, p. 51-52).

2.3.5 Avaliação e Validação de Cluters

Assim que o processo de identificação de *clusters* é finalizado, inicia-se a análise dos *clusters* resultantes. Entretanto, o usuário precisa ter certeza de que os *clusters* resultantes são válidos para que, então, realizem a análise destes.

Normalmente, a avaliação de resultados é feita pela comparação de um resultado certo com o resultado obtido ou pela verificação dos padrões estabelecidos estarem sendo satisfeitos. Como na avaliação de *clusters* o usuário, na maioria das vezes, não possui nenhuma informação acerca dos dados, é difícil identificar se o resultado está correto ou não.

“Uma das formas de avaliação consiste em analisar se o resultado obtido teve alguma influência do método utilizado ou se a configuração de *clusters* resultante se dá devido a alguma relação natural entre os objetos” (WIVES, 2004, p. 52). Esta forma de avaliação é de grande importância devido ao fato de nem todos os conjuntos de dados possuírem relações válidas entre seus elementos.

Outra forma de avaliação se baseia na idéia de que um processo de agrupamento "ótimo" seria aquele em que o resultado melhor se aproxima das partições essenciais ao conjunto de dados, e um conjunto de partições diferentes do ideal poderia levar a decisões incorretas (HALKIDI, BATISTAKIS & VAZIRGIANNIS, 2002, tradução nossa, p. 1). Da mesma forma, há a necessidade do usuário conhecer o conjunto de dados previamente para que ele possa comparar o resultado obtido com o conjunto ideal de partições. O usuário pode também realizar a comparação a partir da identificação e criação de um conjunto, que possua as mesmas características do conjunto ideal, obtido por meio da utilização de métodos estatísticos de geração de dados.

Wives (2004, p. 53) aponta três categorias principais de formas de validação e avaliação do processo de identificação de *clusters*, sendo elas:

- Validação baseada em critérios externos – o critério externo, denominado estrutura de teste, consiste na avaliação dos resultados a partir de uma estrutura pré-especificada (HALKIDI, BATISTAKIS & VAZIRGIANNIS, 2002, tradução nossa, p. 7). Essa estrutura deve ser criada por algum especialista da área, uma vez que usualmente reflete alguma intuição sobre a estrutura do *cluster* do conjunto de dados.
- Validação baseada em critérios internos – os métodos de validação baseados em critérios internos realizam a validação dos resultados levando em consideração os próprios dados (HALKIDI, BATISTAKIS & VAZIRGIANNIS, 2002, tradução nossa, p. 7). Para isso, gera-se um conjunto de dados (artificial) que possuam as mesmas características dos objetos originais (reais), sem a presença de *clusters*. Os resultados são obtidos pela submissão dos conjuntos de dados (artificial e reais) ao algoritmo de identificação de *clusters*. Tais resultados são analisados por métodos estatísticos de validação apropriados ao tipo de dado que está sendo processado.

Validação baseada em critérios relativos – o critério relativo tem como objetivo "identificar os melhores valores de parâmetro para o conjunto de dados em questão" (WIVES, 2004, p. 55). Sendo assim, compara-se a estrutura de *clusters* resultantes com outras estruturas ou esquemas gerados pelo mesmo algoritmo com o fim de avaliar e validar os *clusters*. (HALKIDI, BATISTAKIS & VAZIRGIANNIS, 2002, tradução nossa, p. 7).

2.3.6 Aplicações da Descoberta e Análise de Clusters

A descoberta e a análise de *clusters* possuem diversas aplicações. Dentre elas, destaca-se a área da recuperação de informações e banco de dados (CUTTING, KARGER & PEDERSEN,

1993, tradução nossa, p. 2; SALTON & MCGILL, 1983, tradução nossa, p. 30). Este processo promove a organização e a recuperação de informações ou em outros processos de análise textual, que objetivam a descoberta de conhecimento a partir de textos.

A aplicabilidade da descoberta e análise de *clusters* na área da recuperação de informações se dá ao fato da possibilidade de processamento de uma grande quantidade de documentos, os quais são agrupados em *clusters* de documentos de assuntos similares. Segundo Wives (2004, p. 56), esses "grupos de documentos similares são armazenados em um mesmo local no arquivo de dados indexados de forma que todo um *cluster* seja recuperado, quando um dos documentos que faz parte dele for considerado relevante a uma consulta".

Já na área de descoberta de conhecimento em textos, a descoberta e análise de *clusters* são aplicadas no processo de descoberta de associações entre palavras, o que facilita o desenvolvimento de dicionários e *thesaurus.*

Aldenderfer & Blashfield (1984, tradução nossa *apud* WIVES, 2004, p. 57) destaca que é importante levar em consideração alguns aspectos antes de utilizar a descoberta e análise de *clusters*, sendo tais aspectos:

- Os métodos de análise de *clusters*, em sua grande maioria, são procedimentos relativamente simples. Na maioria das vezes, tais métodos não são suportados por um extenso corpo de raciocínio estatístico (WIVES, 2004, p. 57);
- Os métodos de análise de *clusters* herdam algumas tendências e procedimentos de várias disciplinas (WIVES, 2004, p. 58);
- Podem-se obter diferentes soluções para o mesmo conjunto de dados, por haver a possibilidade de utilizar diferentes métodos de *clusters* (WIVES, 2004, p. 58);

A busca de estrutura é fundamentada na estratégia da análise de *clusters*, embora sua de sua operação impor estruturas (WIVES, 2004, p. 58).

3 MATERIAIS E MÉTODOS

Esta seção apresentará a metodologia e os materiais utilizados para desenvolver o presente trabalho.

3.1 Materiais

Os materiais utilizados no desenvolvimento deste trabalho foram: referências bibliográficas e *softwares.*

As referências bibliográficas utilizadas nesse projeto foram obtidas através de consultas à Internet. Entre tais materiais estão inclusos: monografias de graduação, teses de doutorado, dissertações de mestrado, livros, publicações científicas, artigos dentre outros.

Para o desenvolvimento deste projeto foi utilizado computador pessoal, com a seguinte configuração: Intel Core i5 2.67Ghz, 8Gb de Ram. A conexão ADSL utilizada para conexão com a internet foi de 5Mb.

Para a implementação do sistema foi utilizado o ambiente de desenvolvimento Microsoft Visual Studio 2010 Ultimate, além do ambiente de desenvolvimento NetBeans IDE 7.0.1.

3.2 Metodologia

A metodologia adotada para o desenvolvimento deste trabalho iniciou a partir da realização de pesquisas em diversos sites para encontrar materiais que servissem como referência bibliográfica. Assim, o primeiro passo foi estudar conceitos relacionados a Sistemas de Recomendação e entender a arquitetura do Konnen, que é um Projeto de Pesquisa registrado na COPPEX (Coordenação de Pesquisa, Pós-Graduação e Extensão) do CEULP/ULBRA (Centro Universitário Luterano de Palmas). Dentre tais conceitos destacam-se os tipos de filtragens realizadas em Sistemas de Recomendação que podem ser tanto baseadas em conteúdo quanto colaborativa.

Após a fase de compreensão dos Sistemas de Recomendação, buscou-se na literatura referências sobre o processo de Recuperação da Informação. Compreenderam-se as etapas envolvidas neste processo que consistem na aquisição, preparação, indexação, armazenamento e recuperação da informação.

Concluída a fase de estudo sobre Recuperação da Informação, foi analisa a metodologia de *Clustering* de Documentos. Tal metodologia utiliza algumas técnicas, sendo elas *Clustering* Hierárquico, Algoritmos Particionais, *Density-based* e *Mode-based*.

Foi feita a escolha do algoritmo de *Clustering k-means* Biseccinado, para ser utilizado na organização dos documentos que serão utilizados para a busca de publicações científicas relacionadas na Web, assim como na categorização do retorno destas buscas. Foi utilizada a implementação do algoritmo disponível na biblioteca open-source Air-Head Research[2]

Na aquisição dos dados foi utilizada a Plataforma Lattes[3], que é a base de dados de currículos, instituições e grupos de pesquisa das áreas de Ciência e Tecnologia. Também foi utilizado o repositório público do Bibsonomy[4], que é um sistema de compartilhamento de *bookmarks* e listas de publicações. Por fim, foi utilizada a base de dados de publicações científicas da Microsoft, o Microsoft Academic Search[5].

A partir de um registro da Plataforma Lattes, cujo identificador é "K4775706D6", pertencente ao usuário Edeílson Milhomem da Silva, foram submetidas à recomendação todas as publicações disponíveis em seu perfil. O próprio usuário avaliou as recomendações em um formulário eletrônico disponibilizado por e-mail.

Como parte da preparação dos dados, obtidos no processo de aquisição de dados, foi aplicado o processo de *stop words* para eliminação de palavras com pouca relevância semântica. A interoperabilidade do sistema foi garantida pelo Apache Thrift[6], cujo propósito é tornar possível a comunicação entre várias linguagens de programação. Neste projeto, as linguagens que demandaram a utilização deste framework foram C# e Java.

[2] http://code.google.com/p/airhead-research/
[3] http://lattes.cnpq.br
[4] http://www.bibsonomy.com
[5] http://academic.research.microsoft.com
[6] http://thrift.apache.org

4 RESULTADOS E DISCUSSÕES

Nesta seção, será demonstrada uma abordagem de gerar a recomendação de trabalhos científicos e enriquecer o perfil dos usuários do *Konnen*. Para tanto, será apresentado o contexto do sistema e, logo em seguida, será especificada a estrutura do trabalho que consiste na implantação de um mecanismo de Enriquecimento do Perfil do Usuário e Recomendação de Trabalhos Científicos, o qual será implantado no *Konnen*.

4.1 Visão Geral

Com o fim de possibilitar que uma organização aprenda a partir do desenvolvimento de suas atividades, Souza et al. (2010) propuseram o desenvolvimento de uma plataforma para aprendizagem organizacional através de uma rede de conhecimento. A Figura 6 apresenta a arquitetura e os módulos do *software* em questão, denominado *Konnen*.

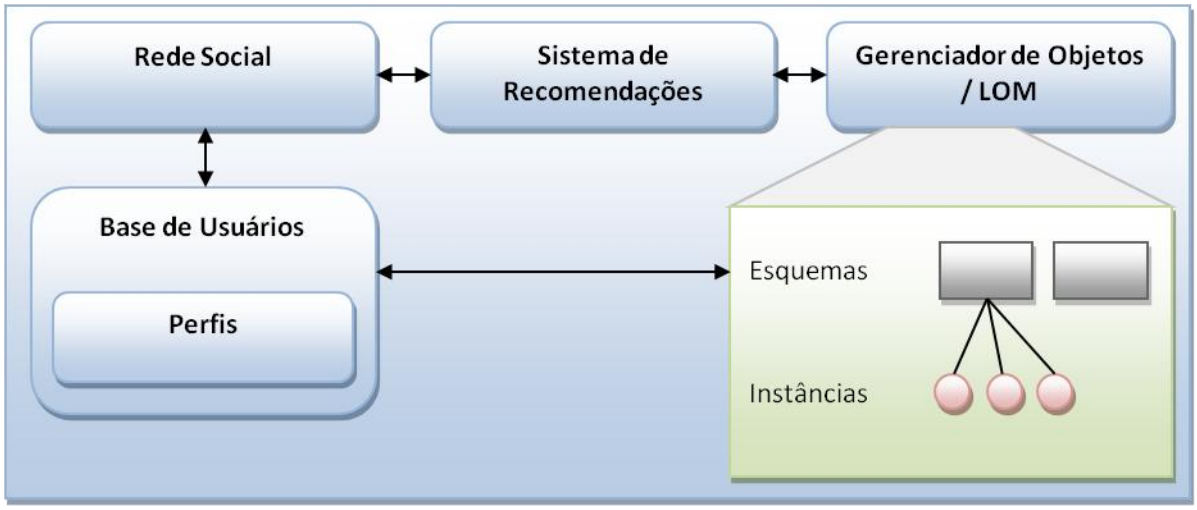

Figura 17: Arquitetura da Plataforma Konnen (SOUZA et al., 2010).

Como pode ser observada na Figura 17, a arquitetura da plataforma *Konnen* está dividida em quatro módulos, sendo eles: base de usuários, rede social, sistema de recomendações e gerenciador de objetos. A base de usuários é responsável por armazenar os dados referentes aos usuários e seus respectivos perfis. O módulo sobre a rede social objetiva

implementar as relações entre os usuários, fornecendo ferramentas para que isso ocorra através de ferramentas de comunicação, como mensagens, chat e fórum. Já o módulo de sistemas de recomendações visa recomendar conteúdos aos usuários a partir do conhecimento obtido na rede social. Por fim, o gerenciador de objetos tem como finalidade gerenciar esquemas (descrições de dados que podem ser criadas pelo usuário) de instâncias (objetos criados pelo usuário ou sistema tendo como base os esquemas).

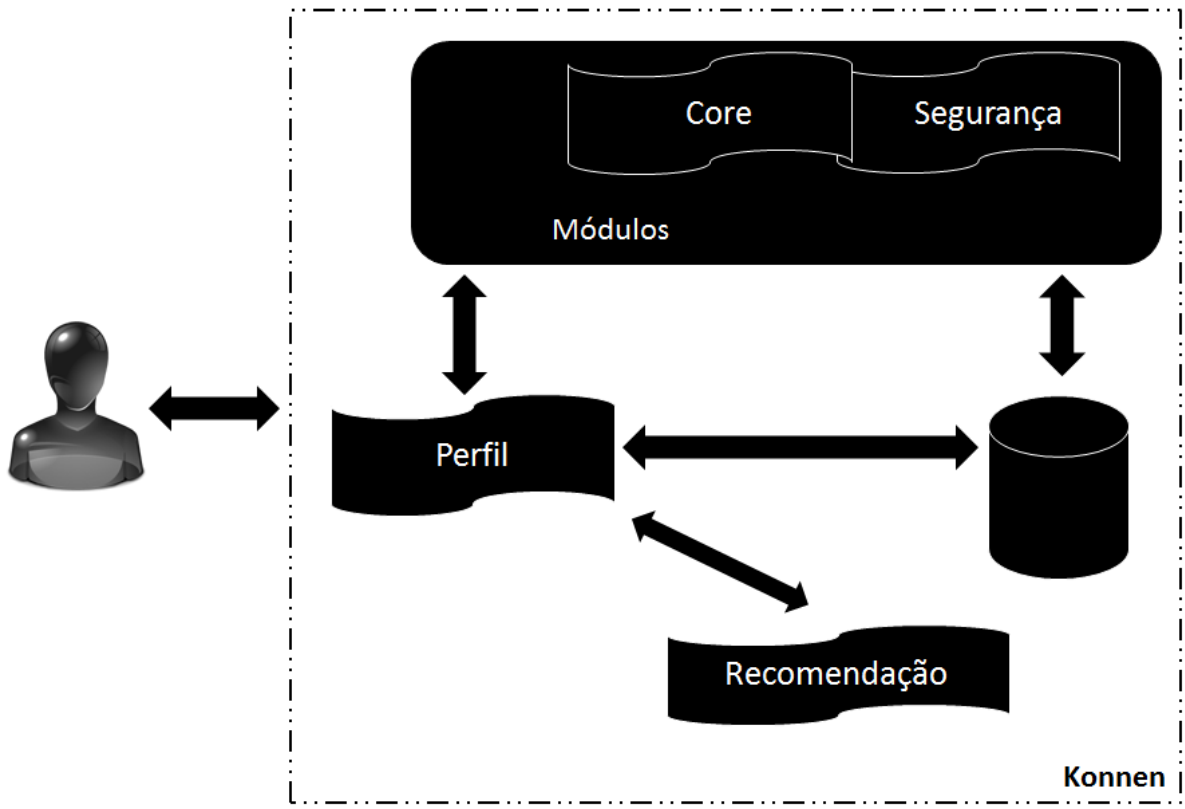

Figura 18: Relação entre o *Konnen* e o Sistema de Recomendação.

A Figura 18 apresenta a relação entre o *Konnen* e o Sistema de Recomendação de *Trabalhos Científicos*. O usuário irá utilizar o módulo de perfil, que recuperará os dados a partir da base de dados e irá interagir com outros módulos do sistema como o *Core*, responsável pelos métodos de acesso a dados e o Módulo de Segurança, que controla a autenticação e autorização do usuário. Por fim, o Módulo de Recomendação enviará para o Módulo de Perfil as informações necessárias para enriquecimento e geração da recomendação de trabalhos científicos.

4.2 Mecanismo de Recomendação de Trabalhos Científicos e sua Implantação no *Konnen*

Esta seção apresentará a aplicação dos conceitos de Recuperação da Informação e *Clustering* para o Mecanismo de Recomendação de Trabalhos Científicos para o *Konnen*.

4.2.1 *WebCrawlers*

O processo de aquisição de dados foi feito a partir de conteúdo disponibilizado no *Currículo Lattes*, *Bibsonomy* e *Microsoft Academic Search*. Os dados foram obtidos via consultas a *web services*, ou diretamente a páginas *web*. O funcionamento dos *WebCrawlers* está descrito nos parágrafos seguintes.

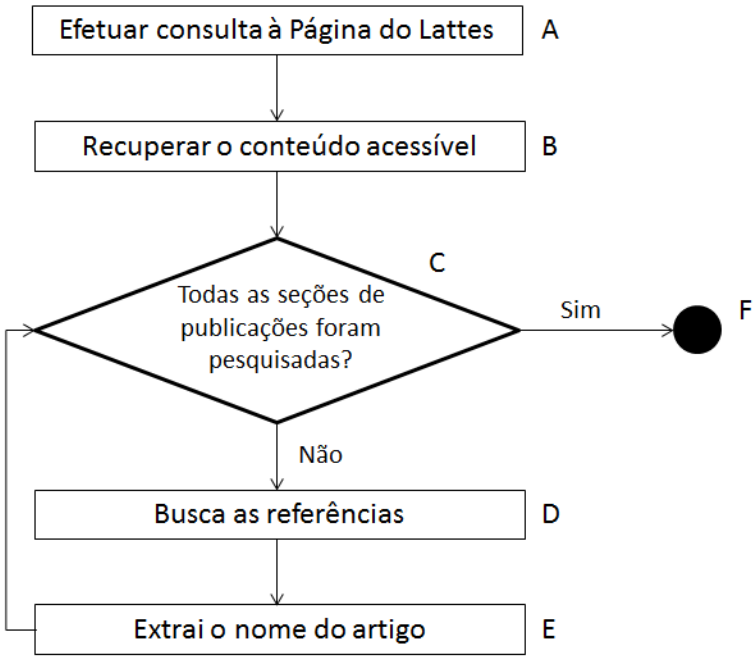

Figura 19: Funcionamento do *WebCrawler* do Currículo *Lattes*.

Como pode ser observado na Figura 19, o algoritmo de *web crawler* responsável pela aquisição da lista de publicações do usuário inicia seu processamento a partir da execução de uma consulta direta à página do Currículo Lattes, passando como parâmetro de URL o código correspondente ao usuário (Figura 19-A), por exemplo, o endereço http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4775706D6. Em seguida, o conteúdo da página é recuperado (Figura 19-B). Por padrão, o conteúdo textual HTML

retornado deve ser convertido em um formato que facilite a navegação pelas marcações do código pela linguagem C#, visto a falta de navegabilidade do conteúdo recuperado. Uma lista de seções marca o trecho das publicações. São executadas consultas a todas estas seções (Figura 19-C) e, como resultado, obtém-se uma lista de referências (Figura 19-D), com a extração do nome do artigo (Figura 19-E) e todas as seções pesquisadas. O *crawler*, então, finaliza sua execução (Figura 19-F).

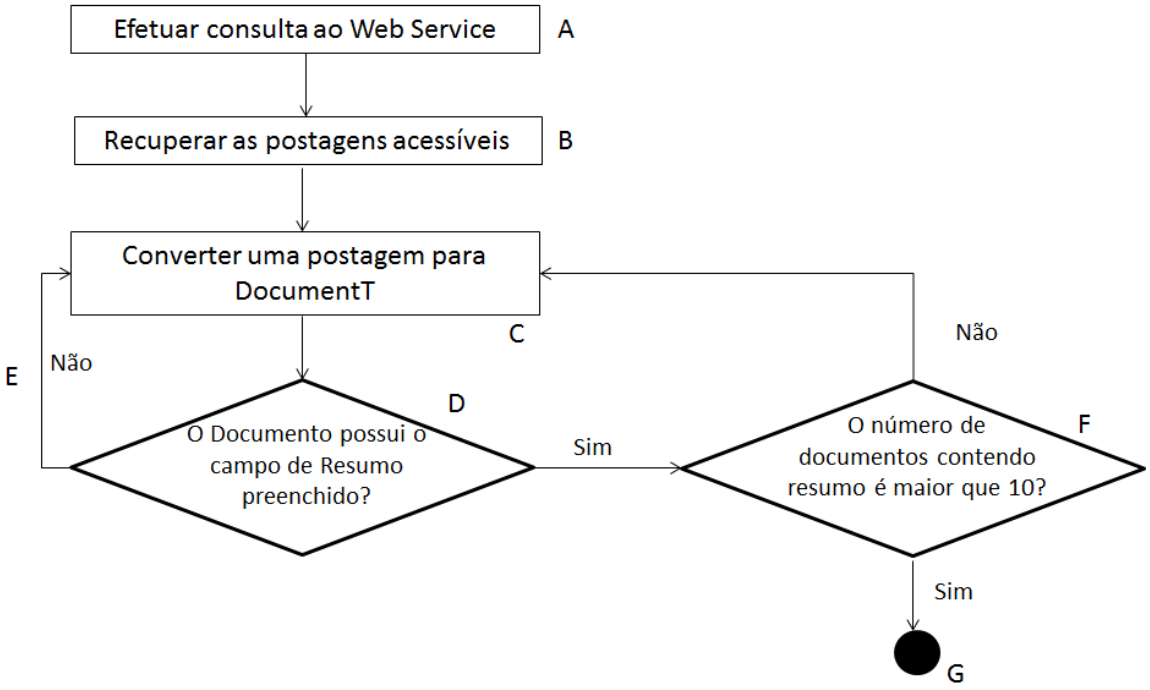

Figura 20: Funcionamento do *WebCrawler* do *Bibsonomy* e *Microsoft Academic Search*.

A pesquisa de conteúdo em repositórios *online* é parte do processo de recomendação. A Figura 20 mostra o processo utilizado para buscas nos repositórios do *Bibsonomy* e do *Microsoft Academic Search*. Inicialmente é feita uma consulta ao *web service* do repositório (Figura 20-A), utilizando como parâmetros: os termos de busca, seja palavra-chave ou trechos de frases e a quantidade máxima de documentos, aqui definido em quinhentos para o *Bibsonomy*) e indefinido para o *Microsoft Academic Search*. Após recuperar as postagens (Figura 20-B), o sistema converte cada registro em um formato padronizado (Figura 20-C). Após a conversão é verificado se o objeto possui o campo resumo (*Abstract*) preenchido (Figura 20-D). Se não estiver preenchido, o objeto é descartado e o próximo registro é analisado. Caso contrário, é adicionado à lista de documentos. Se a quantidade de documentos

não for superior a dez (Figura 20-F), o ciclo de execução continua (Figura 20-C). Sendo superior a 10 documentos, a execução é encerrada (Figura 20-G).

4.2.2 Preparação dos Dados

O processo de aquisição de dados utiliza os fluxos apresentados na seção anterior. Cada conjunto de dados resultante da execução dos *web crawlers* deve passar por um procedimento de preparação, para ser utilizado em algum momento, por exemplo, na execução da técnica *clustering* ou como parâmetro de entrada em algum processo de aquisição de dados. Uma estatística do número de registros obtidos em cada uma das fontes de dados (*WebCrawlers*) é listada abaixo:

Tabela 1: Estatísticas do número de registros obtidos em cada fonte de dados.

	Mínimo	***Máximo***	***Média***
Currículo *Lattes*	**21**	**21**	**21**
Bibsonomy	**17**	**255**	**84,57**
Microsoft Academic Search	**0**	**4**	**0,29**

O vocabulário para a representação vetorial dos termos é formado pelos campos título e resumo, aumentado com a aplicação da expansão de termos. Foram removidas palavras consideradas comuns e de pouca relevância, com a aplicação do *stop-words*. A Figura 21 apresenta o número de *tags* resultante da expansão de termos pelo sistema. O eixo x corresponde a cada artigo e o eixo y corresponde ao número de tags associadas pelo processo de expansão de termos.

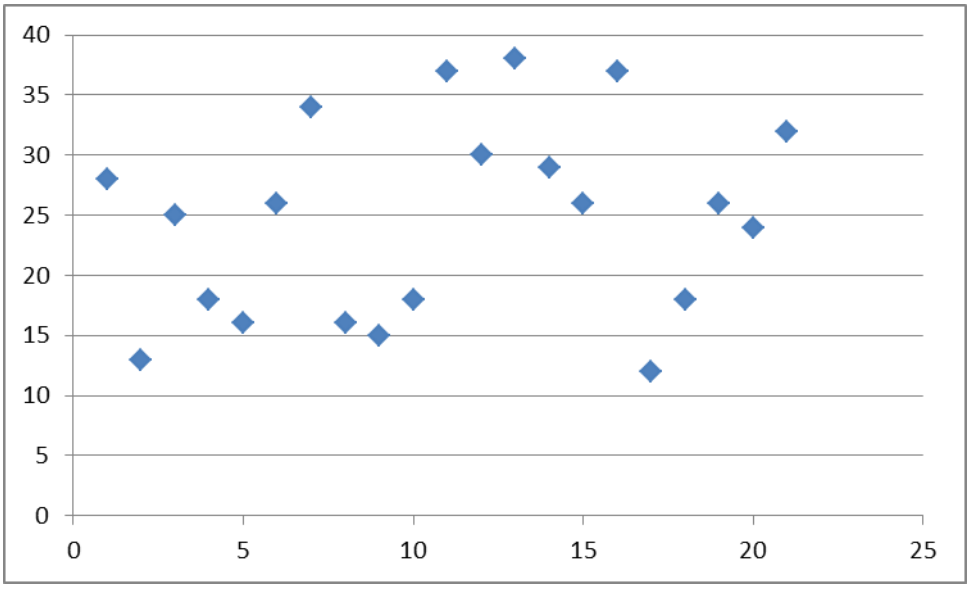

Figura 21: Número de *tags* resultante da expansão de termos.

O processo de expansão de termos é auxiliado pelo uso de *stop-words*, eliminando palavras comuns e com isso excluindo do vocabulário termos que não sejam relevantes para descrever o documento. O processo de aquisição e preparação dos dados pode ser visualizado na figura abaixo:

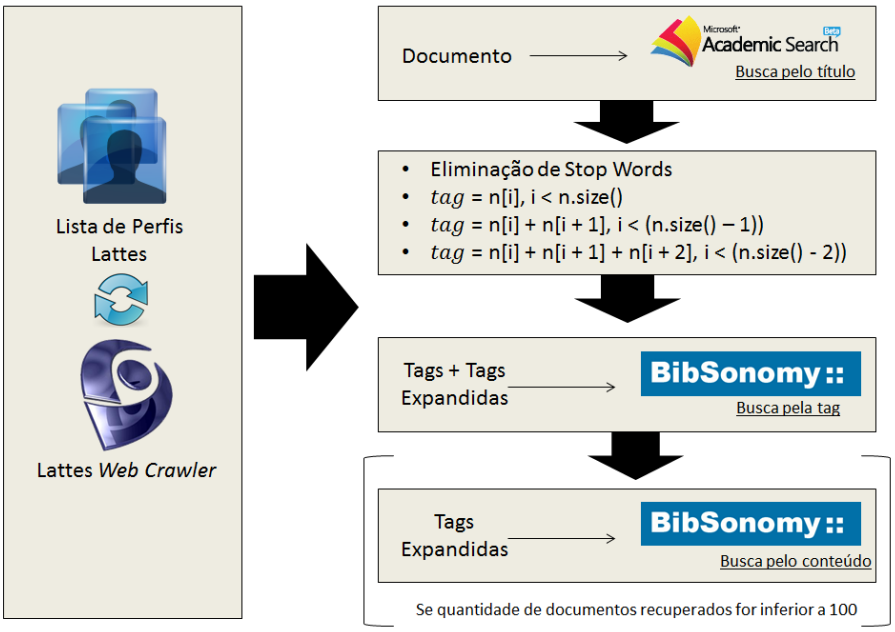

Figura 22: Processo de Aquisição de Preparação dos Dados.

A Figura 22 apresenta o processo global de aquisição e preparação dos dados. A partir de uma lista de perfis da plataforma *Lattes*, para cada item da lista, o Lattes *WebCrawler* responsável pela captura dos dados do *Lattes* recupera as informações inerentes àquele perfil e busca o artigo através da base de dados do *Microsoft Academic Search*. O resultado da busca é encapsulado em um objeto do tipo *DocumentT*, caso o retorno tenha sido nulo, um objeto do mesmo tipo é criado com o campo *Title* preenchido.

Com o intuito de aumentar a quantidade de termos disponíveis para serem pesquisados, o objeto possui o método *ProcessDocument*, este por sua vez, faz chamada ao método *Generate* da classe *TagExpansion*. A Figura 23 mostra o código responsável pela expansão de termos. O próximo passo corresponde à busca, no *Bibsnomy*, de documentos que possuam palavras-chave iguais às *tags* retornadas durante a consulta à base do *Microsoft Academic Search*, e dos termos resultantes do método *ProcessDocument*. Caso o número total de documentos recuperados seja inferior a 100, uma nova busca é feita no *Bibsnomy*, bastando assim, que cada termo esteja presente em qualquer campo do documento.

```
1  public static List<string> Generate(string sentence)
2  {
3      sentence = sentence.ToLower();
4      List<string> returnObj = new List<string>();
5      //1 termo
6      string[] wordsSplit = sentence.Split(
7      new char[] { ' ', ',', '.', ')', '(', ':', '!', '?', ';', '[', ']', '{', '}', '-', '_', '=', '+', '\"', '<', '>' }, StringSplitOptions.RemoveEmptyEntries);
8
9      StopWords sw = new StopWords();
10     string[] swPortugues = sw.GetWords(Broker.Language.PORTUGUES);
11     string[] swIngles = sw.GetWords(Broker.Language.INGLES);
12     foreach (string s in wordsSplit)
13     {
14         if (!(swPortugues.Any(o => o.Equals(s)) || swIngles.Any(o => o.Equals(s))))
15             returnObj.Add(s);
16     }
17     //2 termos e 3 termos
18     string[] wordsSplit2 = sentence.Split(
19     new char[] { ',', '.', ')', '(', ':', '!', '?', ';', '[', ']', '{', '}', '=', '+', '<', '>' }, StringSplitOptions.RemoveEmptyEntries);
20     foreach (string s in wordsSplit2)
21     {
22         //2 termos
23         string[] p1 = s.Split(new char[] { ' ' });
24         for (int i = 0; i < p1.Count() - 1; i++)
25         {
26             string temp = p1[i] + " " + p1[i + 1];
27             returnObj.Add(temp);
28         }
29         //3 termos
30         for (int i = 0; i < p1.Count() - 2; i++)
31         {
32             string temp = p1[i] + " " + p1[i + 1] + " " + p1[i + 2];
33             returnObj.Add(temp);
34         }
35     }
36     return returnObj;
37 }
```

Figura 23: Processo de Expansão de Termos.

O código da Figura 23 retorna uma lista de termos extraídos do parâmetro de entrada *sentence*. A execução do código compreende três etapas principais: termos com uma palavra, termos com duas palavras e termos com três palavras. A linha 5 apresenta o início do processamento dos termos com uma palavra, para estes são eliminadas palavras presentes na lista de *StopWords*, visto que sozinhas, não apresentam resultado semântico relevante. Entre

as linhas 18 e 35, são gerados os termos com duas e três palavras. Cada palavra é concatenada a sua consecutiva, agrupando-se em pares e trios. O próximo passo, visando a alcançar a recomendação e o enriquecimento do perfil do usuário, é a aplicação da técnica de *clustering*.

4.2.3 *Clustering* e Recomendação

A partir da entrada e tratamento de dados no sistema pelos processos de aquisição e preparação, os dados são convertidos em uma estrutura, que será repassada via chamada de serviço para o método responsável pelo *clustering*. A execução da técnica de *clustering* retorna uma matriz frequência, isto é, linhas correspondendo termos, colunas referenciando documentos e os itens contendo a frequência de cada termo no respectivo documento. Importante dizer que cada termo está vinculado a um cluster. Este procedimento está exemplificado na figura abaixo.

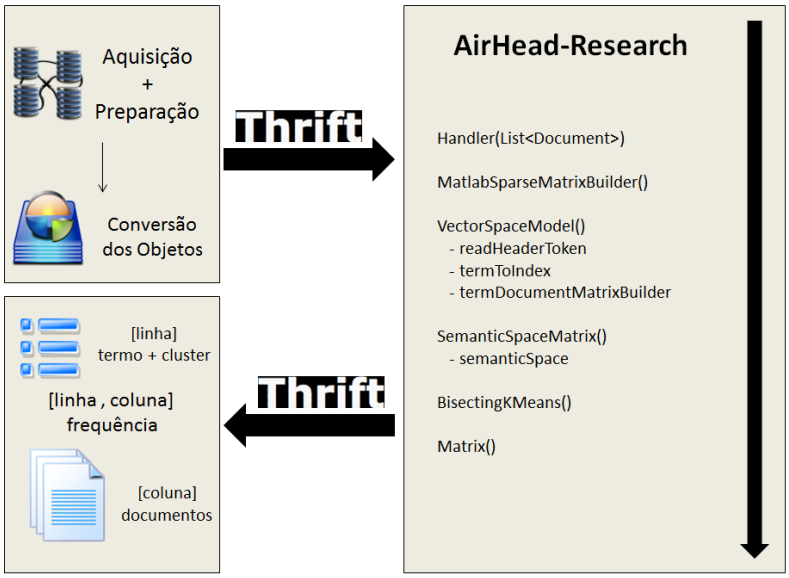

Figura 24: Representação do processo de *clustering*.

A partir da Figura 24 deve-se destacar que a camada de negócios que pertence ao processo de *clustering* está implementada em duas linguagens de programação distintas: C# e Java. A comunicação entre as duas linguagens é feita utilizando um *middleware* chamado *Apache Thrift*. Esta arquitetura permite que máquinas distintas dividam a carga na execução de tarefas, garantindo a interoperabilidade e escalabilidade do sistema.

Dado um documento *docTemp* que representa uma instância da classe *DocumentT*, é gerada uma lista de documentos (*DocumentT*) *docListTemp* a partir da aplicação dos

procedimentos descritos nas seções 4.2.1 e 4.2.2. No processo de construção da lista *docListTemp* é garantido que os documentos possuam os atributos *Title* e *Abstract* preenchidos.

```
1   Document document = new Document();
2
3   docTemp.ProcessDocumentTitle(true);
4   docTemp.FillTermSplit();
5
6   document.Id = docTemp.Id.ToString();
7
8   List<string> terms = new List<string>();
9
10  foreach (Tag t in docTemp.TagExpansion)
11  {
12      int count = docTemp.TermSplit.Count(o => o.Equals(t.Name));
13
14      for (int i = 0; i < count; i++)
15      {
16          terms.Add(t.Name);
17      }
18  }
19
20  document.Terms = terms;
```

Figura 25: Código da conversão de objetos.

A chamada ao serviço provido pelo *middleware* requer uma conversão do tipo *TothLibrary.Object.DocumentT* para o *TothLibrary.Broker.Document*. A Figura 25 apresenta o processo de conversão completo para cada objeto. A linha 6 registra o vínculo entre os objetos dos dois tipos, o que é feito pelo atributo *Id*. A conversão é iniciada com o objeto do tipo *DocumentT*, que realiza uma chamada ao método *ProcessDocumentTitle* e passa como parâmetro *true*. Isso permite que as *tags* originais do documento sejam adicionadas aos termos expandidos.

Uma lista intermediária, *TermSplit*, é preenchida com a concatenação dos atributos *Title* e *Abstract* e consequente divisão dos termos concatenados. Esta lista será utilizada para cálculo da frequência dos termos presentes na propriedade *TagExpansion*, já que estes termos são todos distintos. A lista *docList* do tipo *TothLibrary.Broker.Document* armazenará os respectivos objetos convertidos de *docTemp* e os presentes na lista *docListTemp*. Efetuada a conversão de todos os documentos, é feita a chamada ao código Java.

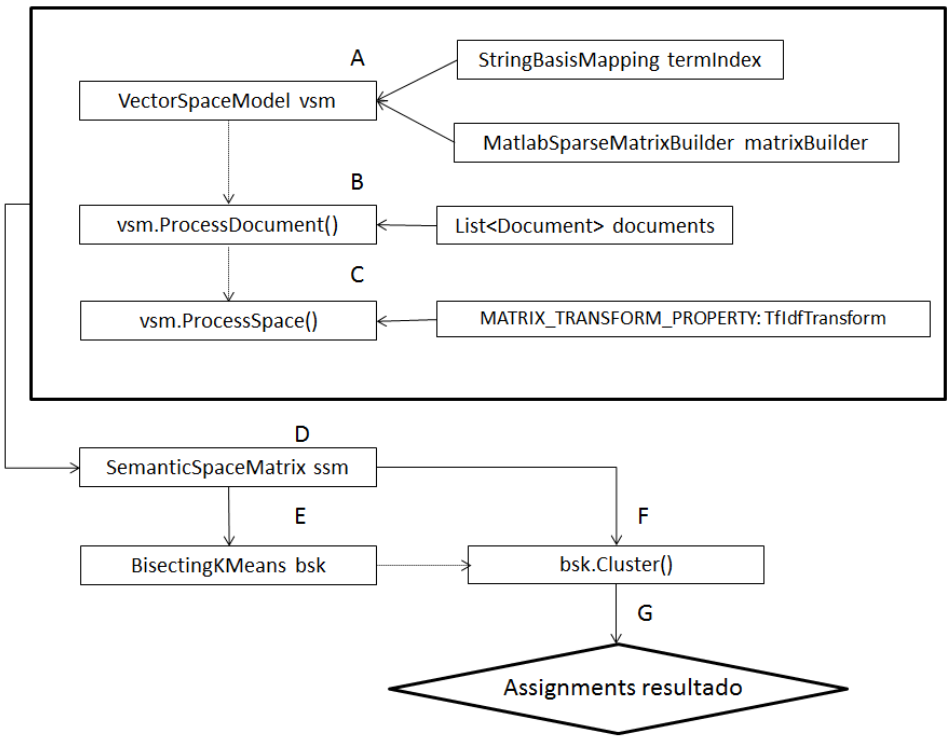

Figura 26: Procedimento para execução da técnica de *clustering* no Java.

A Figura 26 demonstra o procedimento para execução da técnica de *clustering*. O modelo do vetor espacial deve ser definido logo a princípio (Figura 26-A). É composto de um construtor de matrizes e de um indexador de termos. Definidos os parâmetros de instanciação, todos os documentos são adicionados para que seus termos sejam indexados (Figura 26-B). A seguir, é escolhido o tipo de métrica a ser aplicada futuramente para avaliação do grau de relevância de cada termo (Figura 26-C). A matriz semântica é criada a partir do Modelo Vetorial Espacial (Figura 26-D). Definida a técnica de *clustering* (Figura 26-E), a execução inicia-se com os dados provenientes da matriz (Figura 26-E). Ao final, uma lista de atribuições (Figura 26-F) apresenta o resultado final do processo de *clustering*. Uma última conversão é feita com a finalidade de retornar ao *C#* o resultado do método (Figura 24).

A lista final de documentos a serem recomendados para *docTemp* é gerada por regras que combinam informações de três listas. Inicialmente, são selecionados os três termos mais frequentes de *docTemp*. A partir da matriz retornada pelo Java, buscam-se os clusters correspondentes a cada um destes termos e recupera-se a respectiva lista de documentos. A lista de documentos é ordenada em ordem decrescente de frequência pelo termo analisado, e são selecionados respectivamente, 6, 4 e 2 termos superiores de cada lista formada pelos termos mais comuns de *docTemp*.

4.2.4 Resultados

Nesta seção são apresentados os resultados da abordagem *clustering* biseccionado baseado em frequência de termos. Primeiramente é validada a qualidade da recomendação a partir da avaliação dos artigos recomendados por parte do autor.

Tabela 2: Estatísticas dos artigos analisados.

	Mínimo	***Máximo***	***Média***
Duração	**01 min. 41 seg.**	**12 min. 11 seg.**	**04 min. 56 seg.**
Tags	**0**	**5**	**0,62**
Expansão semântica	**12**	**38**	**24,67**

Após a execução do mecanismo, foram processados 21 artigos, durante 1 hora 43 minutos e 45 segundos. Os resultados individuais para cada documento serão apresentados logo abaixo:

- **Artigo [1]**: A Process to Manage Corporate Knowledge Using Social Networks: A Case Study.

O tempo de execução para este artigo foi de **12 minutos e 11 segundos**, o conteúdo do artigo foi encontrado online e resultou na obtenção de **4 tags**, a expansão de termos resultou em **28 registros**; a quantidade de documentos distintos recuperados para análise foi de **255 publicações**; o tempo de execução do *clustering* foi de **03 minutos e 27 segundos**, em que foi analisado um total de **1549 termos**.

- **Artigo [2]**: A.M.I.G.O.S: Knowledge Management and Social Networks.

O tempo de execução para este artigo foi de **12 minutos e 32 segundos**, o conteúdo do artigo foi encontrado online e resultou na obtenção de **5 tags**, a expansão de termos resultou em **13 registros**; a quantidade de documentos distintos recuperados para análise foi de **214 publicações**; o tempo de execução do *clustering* foi de **02 minutos e 02 segundos**, em que foram analisados um total de **1263 termos**.

- **Artigo [3]**: A.M.I.G.O.S: Uma plataforma para Gestão de Conhecimento através de Redes Sociais.

O tempo de execução para este artigo foi de **03 minutos e 24 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos

resultou em **25 registros**; a quantidade de documentos distintos recuperados para análise foi de **34 publicações**; o tempo de execução do *clustering* foi de **04 segundos**, em que foram analisados um total de **251 termos**.

- **Artigo [4]**: A.M.I.G.O.S: Using Social Networks to Manage Corporate Knowledge.

O tempo de execução para este artigo foi de **08 minutos e 37 segundos**, o conteúdo do artigo foi encontrado online e resultou na obtenção de **4 tags**, a expansão de termos resultou em **18 registros**; a quantidade de documentos distintos recuperados para análise foi de **173 publicações**; o tempo de execução do *clustering* foi de **01 minutos e 01 segundo**, em que foram analisados um total de **1008 termos**.

- **Artigo [5]**: Construção de um Sistema para Gerenciamento de Eventos.

O tempo de execução para este artigo foi de **02 minutos e 01 segundo**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **16 registros**; a quantidade de documentos distintos recuperados para análise foi de **18 publicações**; o tempo de execução do *clustering* foi de **14 segundos**, em que foram analisados um total de **157 termos**.

- **Artigo [6]**: EP-RDF: SISTEMA PARA ARMAZENAMENTO E RECUPERAÇÃO DE IMAGENS BASEADO EM ONTOLOGIA.

O tempo de execução para este artigo foi de **05 minutos e 58 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **26 registros**; a quantidade de documentos distintos recuperados para análise foi de **79 publicações**; o tempo de execução do *clustering* foi de **28 segundos**, em que foram analisados um total de **481 termos**.

- **Artigo [7]**: Formalização da Base de Conhecimento para um Sistema Especialista em Diagnóstico de Defeitos em Hardware.

O tempo de execução para este artigo foi de **05 minutos e 04 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **34 registros**; a quantidade de documentos distintos recuperados para análise foi de **58 publicações**; o tempo de execução do *clustering* foi de **17 segundos**, em que foram analisados um total de **501 termos**.

- **Artigo [8]**: Improving Communication and Cooperation Through Recommendation System.

O tempo de execução para este artigo foi de **06 minutos e 48 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos

resultou em **16 registros**; a quantidade de documentos distintos recuperados para análise foi de **123 publicações**; o tempo de execução do *clustering* foi de **47 segundos**, em que foram analisados um total de **836 termos**.

- **Artigo [9]**: O padrão RDF na descrição de Imagen

O tempo de execução para este artigo foi de **02 minutos e 48 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **15 registros**; a quantidade de documentos distintos recuperados para análise foi de **37 publicações**; o tempo de execução do *clustering* foi de **04 segundos**, em que foram analisados um total de **219 termos**.

- **Artigo [10]**: OPENUSER: Sistema Unificado Aberto de Usuários na Web

O tempo de execução para este artigo foi de **01 minutos e 21 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **18 registros**; a quantidade de documentos distintos recuperados para análise foi de **53 publicações**; o tempo de execução do *clustering* foi de **08 segundos**, em que foram analisados um total de **339 termos**.

- **Artigo [11]**: ProGerWMI: Utilizando WMI para a Construção do Protótipo de uma Ferramenta para o Gerenciamento de Redes

O tempo de execução para este artigo foi de **05 minutos e 02 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **37 registros**; a quantidade de documentos distintos recuperados para análise foi de **53 publicações**; o tempo de execução do *clustering* foi de **22 segundos**, em que foram analisados um total de **411 termos**.

- **Artigo [12]**: Programação Baseada em Camadas através de Recursos do ASP.NET 2.0: uma Aplicação Prática

O tempo de execução para este artigo foi de **06 minutos e 30 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **30 registros**; a quantidade de documentos distintos recuperados para análise foi de **85 publicações**; o tempo de execução do *clustering* foi de **37 segundos**, em que foram analisados um total de **643 termos**.

- **Artigo [13]**: Promovendo Melhorias na Comunicação e Colaboração em uma Plataforma de Gestão de Conhecimento Através de Recomendações

O tempo de execução para este artigo foi de **03 minutos e 02 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos

resultou em **38 registros**; a quantidade de documentos distintos recuperados para análise foi de **17 publicações**; o tempo de execução do *clustering* foi de **02 segundos**, em que foram analisados um total de **153 termos**.

- **Artigo [14]**: RDF na Definição de um Ambiente para o Armazenamento e Recuperação de Monografias

O tempo de execução para este artigo foi de **04 minutos e 51 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **29 registros**; a quantidade de documentos distintos recuperados para análise foi de **64 publicações**; o tempo de execução do *clustering* foi de **23 segundos**, em que foram analisados um total de **440 termos**.

- **Artigo [15]**: Sistema de Correção Automática de Endereços Digitados Incorretamente em Browsers Web

O tempo de execução para este artigo foi de **04 minutos e 25 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **26 registros**; a quantidade de documentos distintos recuperados para análise foi de **56 publicações**; o tempo de execução do *clustering* foi de **10 segundos**, em que foram analisados um total de **357 termos**.

- **Artigo [16]**: Sistema de Recomendação como Mecanismo para Promover Melhorias no Processo de Comunicação e Colaboração nas Organizações

O tempo de execução para este artigo foi de **03 minutos e 29 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **37 registros**; a quantidade de documentos distintos recuperados para análise foi de **32 publicações**; o tempo de execução do *clustering* foi de **05 segundos**, em que foram analisados um total de **236 termos**.

- **Artigo [17]**: Sistema para Geração de Documentos RDF

O tempo de execução para este artigo foi de **02 minutos e 52 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **12 registros**; a quantidade de documentos distintos recuperados para análise foi de **46 publicações**; o tempo de execução do *clustering* foi de **16 segundos**, em que foram analisados um total de **261 termos**.

- **Artigo [18]**: Social Knowledge Management in Practice: A Case Study

O tempo de execução para este artigo foi de **11 minutos e 46 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos

resultou em **18 registros**; a quantidade de documentos distintos recuperados para análise foi de **186 publicações**; o tempo de execução do *clustering* foi de **01 minuto e 37 segundos**, em que foram analisados um total de **1331 termos**.

- **Artigo [19]**: Um Processo para Gestão do Conhecimento Organizacional através de Redes Sociais

O tempo de execução para este artigo foi de **02 minutos e 49 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **26 registros**; a quantidade de documentos distintos recuperados para análise foi de **33 publicações**; o tempo de execução do *clustering* foi de **05 segundos**, em que foram analisados um total de **261 termos**.

- **Artigo [20]**: Utilizando Redes Sociais e Folksonomy para Localizar Especialistas de Domínio

O tempo de execução para este artigo foi de **04 minutos e 03 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **24 registros**; a quantidade de documentos distintos recuperados para análise foi de **56 publicações**; o tempo de execução do *clustering* foi de **12 segundos**, em que foram analisados um total de **398 termos**.

- **Artigo [21]**: XML e Java no Desenvolvimento de um Editor de Fórmulas do Cálculo de Predicados

O tempo de execução para este artigo foi de **06 minutos e 19 segundos**, o conteúdo do artigo não foi encontrado online e resultou na obtenção de **0 tags**, a expansão de termos resultou em **32 registros**; a quantidade de documentos distintos recuperados para análise foi de **104 publicações**; o tempo de execução do *clustering* foi de **43 segundos**, em que foram analisados um total de **724 termos**.

Tabela 3: Resultado das avaliações dos artigos recomendados.

	Mínimo	***Máximo***	***Média***
Artigo[1]	**1**	**5**	**3,33**
Artigo[2]	**2**	**5**	**3,36**
Artigo[3]	**1**	**5**	**2,2**
Artigo[4]	**1**	**5**	**2,54**
Artigo[5]	**1**	**3**	**1,33**
Artigo[6]	**1**	**5**	**2,11**

Artigo[7]	**1**	**4**	**2,72**
Artigo[8]	**1**	**5**	**2,18**
Artigo[9]	**3**	**5**	**3,9**
Artigo[10]	**1**	**4**	**2,27**
Artigo[11]	**1**	**2**	**1,11**
Artigo[12]	**2**	**4**	**2,6**
Artigo[13]	**1**	**4**	**3**
Artigo[14]	**1**	**5**	**2,33**
Artigo[15]	**1**	**5**	**1,81**
Artigo[16]	**1**	**4**	**2**
Artigo[17]	**2**	**5**	**3,63**
Artigo[18]	**2**	**5**	**3,63**
Artigo[19]	**1**	**5**	**2,36**
Artigo[20]	**3**	**5**	**4**
Artigo[21]	**1**	**3**	**2,45**
Média Geral	**1,38**	**4,43**	**2,61**

Como consequência da Tabela 3, a média das avaliações máximas atribuídas ficou acima de “bom”, considerando o intervalo de avaliação (1 – muito fraco, 2 – fraco, 3 – regular, 4 – bom, 5 muito bom). Em contrapartida, a média das avaliações mínimas ficou ligeiramente acima da pontuação “muito fraco”.

Em um cenário ideal, a execução do algoritmo seria capaz de recuperar os documentos oriundos da Plataforma Lattes na Web, atribuindo-lhe uma lista inicial de palavras-chave. Os artigos [1] e [2] foram encontrados através da base de dados do *Microsoft Academic Search*, e obtiveram avaliações mínimas “muito fracas”, e avaliações máximas como “muito bom”. Um fato a se observar é que a média de ambos foi registrada entre “regular” e “bom” e apesar de haver outros casos com avaliações mínimas e máximas iguais, as médias diferem-se consideravelmente, ficando a maioria abaixo do “regular”.

As maiores médias são marcadas pela estrutura de títulos compostas por termos-chave simples ou compostos, esta composição ameniza a necessidade de encontrar o documento on-line, tornando a expansão de termos mais eficientes para a busca de documentos candidatos a recomendação.

Na tentativa de evitar esses desvios de resultados, a aplicação do processo de *tagging* (atribuição de palavras-chave) e o armazenamento local dos documentos, melhoram o

desempenho do sistema, pois menos consultas serão feitas para preenchimento da lista que deve ser analisada no processo de *clustering*.

5 CONSIDERAÇÕES FINAIS

Nesse trabalho foram abordados os conceitos sobre Sistemas de Recomendação, Recuperação da Informação e *Clustering*. As compreensões destes conceitos foram de grande importância para que fosse possível a definição e o desenvolvimento do mecanismo de enriquecimento do perfil do usuário e recomendação de trabalhos científicos. Com base nestes estudos foi definido que a utilização de *WebCrawlers* para a aquisição dos dados representaria uma forma automatizada para busca de informações facilitando a vida do usuário. A técnica de *clustering* utilizada foi o K-Means biseccionado.

Como etapa do processo de aquisição da informação foi necessária a implementação de três *WebCrawlers*. Um responsável por buscar as informações do usuário na Plataforma Lattes e outros dois visando buscar informações a respeito destas publicações na *Web*. Na etapa de aquisição da informação encontrou-se dificuldade em encontrar as publicações listadas na Plataforma Lattes nos repositórios de dados. Na etapa de *clustering* foi utilizada uma biblioteca em Java, pois esta implementa a técnicas de *clustering* escolhida. Para executar os algoritmos presentes na biblioteca, houve certo atraso, pois foram necessárias alterações no código da mesma, para a correta execução do K-Means biseccionado, tendo como parâmetro, conteúdo semântico (palavras) e não apenas valores flutuantes. A comunicação da biblioteca, escrita em Java, com os demais módulos, escritos em C# foi auxiliado com o Apache Thrift.

Após realizar a implementação da ferramenta de recomendação, pode-se perceber que o processamento do mecanismo deve ocorrer *off-line*, ou seja, como um serviço executando a partir de uma fila de solicitações, pois o tempo necessário para gerar a recomendação é alto, além de depender de fatores externos, como a disponibilidade de dados relevantes nos repositórios pesquisados.

A partir da metodologia definida, utilizou-se um perfil específico da Plataforma Lattes, devido ao tempo disponível para testes e ajustes de código. Dessa forma, ao realizar os testes, o usuário preencheu um questionário avaliando os conteúdos recomendados. Pode-se comprovar a funcionalidade da ferramenta. Assim, também verificou-se que a codificação de

caracteres pode ser um problema na análise dos dados, visto que o sistema não reconhece determinados caracteres, ocasionando erros de comparação de *string*.

Por fim, vale ressaltar que o resultado do presente trabalho será implantado no Konnen, que ainda está em estágio de desenvolvimento. Além disso, a arquitetura utilizada torna o sistema baixo acoplado, permitindo que o mesmo seja facilmente integrado a outras ferramentas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALDENDERFER, M. S.; BLASHFIELD, R. K. **Cluster Analysis.** Beverly Hills, CA: Sage, 1984. 88 p. *apud* WIVES, L. K. **Utilizando conceitos como descritores de textos para o processo de identificação de conglomerados (clustering) de documentos.** Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. Disponível em: <http://www.leandro.wives.nom.br/pt-br/publicacoes/Tese.pdf>. Acessado em: abril 2011.

ARASU, A. et al. **Searching the *web*.** ACM Transaction on Internet Technology, v. 1, n. 1, p. 2-43, Ago 2001. Disponível em: <https://*web*archive.jira.com/wiki/*download*/attachments/5441/2001-Arasu-search.pdf>. Acessado em: abril 2011.

BAEZA-YATES, R. **An extended model for full text databases.** Journal of Brazilian Computer Society, v. 2, n. 3, abril, 1996. Disponível em: <http://www.pms.ifi.lmu.de/publikationen/projektarbeiten/Felix.Weigel/xmlindex/material/baeza94hybrid.pdf>. Acessado em: abril 2011.

BALABANOVIC, M.; SHOHAM, Y. **Fab:** content-based, collaborative recommendation. In: Commun. ACM, New York, 40(3), p. 66-72, 1997. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.118.2150&rep=rep1&type=pdf>. Acessado em: abril 2011.

CARDOSO, O. N. P. **Recuperação de Informação.**6 p. Departamento de Ciência da Computação – Universidade Federal de Lavras, Lavras, 2002. Disponível em: <http://www.dcc.ufla.br/infocomp/artigos/v2.1/art07.pdf>. Acessado em: abril 2011.

CASTILHO, C.; BAEZA-YATES, R. **A new model for *web* crawling.** In: International World Wide *Web* Conference, 11, 2002, Honolulu *apud* OLIVEIRA, B. G. **Sistema de Recuperação de Informação para a Busca de Notícias na Área Tecnológica.** Monografia Ciência da Computação) – Universidade do Vale do Itajaí, Itajaí, 2008. Disponível em: <http://siaibib01.univali.br/pdf/Bruno%20Goulart%20de%20Oliveira.pdf>. Acessado em: abril 2011.

CAZELLA, S. C. **Aplicando a Relevância da Opinião de Usuários em Sistema de Recomendação para Pesquisadores.** Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <http://www.dcomp.ufs.br/~gutanunes/hp/TCI/teseSilvio.pdf>. Acessado em: abril 2011.

CHO, J.; GARCIA-MOLINA, H. **Synchronizing a database to Improve Freshness.** ACM SIGMOD Record, v. 29, n. 2, p. 117-128, June 2000. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.4718&rank=1>. Acessado em: abril 2011.

COELHO, J. A. F. **Levantamento de Demanda para Implantação de um Laboratório Colaborativo de Teste de Software no APL de TI de Fortaleza-CE.** Monografia (Pós-Graduação em Agentes Gestores de APLs) – Universidade de Fortaleza, Fortaleza, 2010. Disponível em: <http://testeplone.cidades.ce.gov.br/categoria4/Jose%20Antonio%20Farias%20Coelho.pdf/>. Acessado em: abril 2011.

CROFT, W. B. **Knowledge-based and statistical approaches to text retrieval.** IEEE Intelligent Systems and Their Applications, v.8, n. 2, p. 8-12, April 1993 *apud* SILVA, F. R.G. **Geodiscover:** mecanismo de busca especializado em dados geográficos. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisa Espaciais, São José dos Campos, 2006. Disponível em: <http://urlib.net/rep/sid.inpe.br/MTC-m13@80/2006/11.07.13.25?languagebutton=en>. Acessado em: abril 2011.

CUTTING, D. R.; KARGER, D. R.; PEDERSEN, J. O. **Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections.** In: Annual International ACM-SIGIR Conference on Research and Development in Information retrieval, SIGIR, New York: ACM Press, 1993, p. 126-134. Disponível em: <http://www.jopedersen.com/Publications/cutting93constant.pdf>. Acessado em: abril 2011.

FONSÊCA, J. N. **Um Modelo de Indexação Semântica para Intranet.** Dissertação (Mestrado em Informática Aplicada) – Universidade de Fortaleza, Fortaleza, 2002. Disponível em: <https://uol01.unifor.br/oul/conteudosite/F8464526323/Dissertacao%20-%20MIA%20Jansley%20Nobre%20da%20Fonseca.pdf>. Acessado em: abril 2011.

GODOY NETO, M. **Rave au Disco:** Recomendação automática de vídeo como auxílio no processo de disseminação do conhecimento. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2009. Disponível em: <http://www.bdtd.ufpe.br/tedeSimplificado//tde_busca/arquivo.php?codArquivo=6959>. Acessado em: abril 2011.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. **Cluster Validity Methods:** Part I. ACM SIGMOD Record, New York, v. 31, n. 2, p. 40-45, 2002. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.19.1874&rep=rep1&type=pdf>. Acessado em: abril 2011.

HATZIVASSILOGLOU, V.; GRAVANO, L.; MAGANTI, A. **An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering.** Proceedings of the 23º Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000. Disponível em: <http://citeseer.ist.psu.edu/viewdoc/*download*;jsessionid=0DEFFD89260CAEEA49AABC7C2B646EF8?doi=10.1.1.34.1259&rep=rep1&type=pdf>. Acessado em: abril 2011.

HERLOCKER, J. L. **Understanding and Improving Automated Collaborative Filtering Systems.** Tese (Doutorado em Ciência da Computação) – University of Minnesota, Minnesota, 2000. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.86.5842&rep=rep1&type=pdf>. Acessado em: abril 2011.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. **Data Clustering:** A Review. ACM Computing Surveys, New York, v. 31, n. 3, p. 264-323, 1999. Disponível em: <http://citeseer.ist.psu.edu/viewdoc/*download*;jsessionid=8AFDD6A36E6C771C71513A639BE2357C?doi=10.1.1.18.2720&rep=rep1&type=pdf>. Acessado em: abril 2011.

KONGTHON, A. **A Text Mining Framework for Discovering Technological Intelligence to Support Science and Technology Management.** Ph.D. Thesis, Georgia Institute of Technology, Atlanta, USA, 2004. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.136.9605&rep=rep1&type=pdf>. Acessado em: abril 2011.

LICHTNOW, D. et al. **O uso de técnicas de recomendação em um sistema para apoio à aprendizagem colaborativa.** Revista Brasileira de Informática na Educação, Porto Alegre: Sociedade Brasileira de Computação, v. 14, n. 3, p. 49-59, Set./Dez. 2006. Disponível em: <http://www.br-ie.org/pub/index.php/rbie/article/viewFile/46/40>. Acessado em: abril 2011.

LOPES, M. C. S. **Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português.** Tese (Doutorado em Ciências em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004. Disponível em: <http://200.201.81.50/~jorge/ARTIGOS%20INTERESSANTES/LING%DC%CDSTICA%20e%20MECANISMOS%20DA%20LINGUAGEM/Tese_coc_ufrj.pdf>. Acessado em: abril 2011.

MELLO, A. A. **Aplicação das Técnicas de Mineração de Textos e Sistemas Especialistas na Liquidação de Processos Trabalhistas.** Dissertação (Mestrado em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007. Disponível em: <http://www.coc.ufrj.br/index.php?option=com_docman&task=doc_*download*&gid=1663>. Acessado em: abril 2011.

METZ, J.; MONARD, M. C. **Utilizando uma Abordagem Híbrida para Interpretação do Clustering Hierárquico.** In: Workshop em Algoritmos e Aplicações de Mineração de Dados, João Pessoa, 2007, p. 5-12. Disponível em: <http://www.lbd.dcc.ufmg.br:8080/colecoes/waamd/2007/001.pdf>. Acessado em: abril 2011.

MOTTA, C. L. R.; BORGES, M. R. S. **TeamWorks:** teamwork collaborative environment. In Proceedings of Sixth Brazilian Symposium of Multimedia and Hypermedia – SBMIDIA'2000, p. 14-16, June, Natal, RN, Brazil, p. 259-272, June, Natal, RN, Brazil, p. 259-272, In Portuguese *apud* MOTTA, C. L. R.; LOPES, L. M. C. **Sistema de Recomendação apoiando a TV Escola.** Anais do XIII Simpósio Brasileiro de Informática na Educação – SBIE 2002, p. 377-384. Editora Unisinos, São Leopoldo, RS. Disponível em <http://www.br-ie.org/pub/index.php/sbie/article/viewFile/199/185>. Acessado em: abril 2011.

PAIVA, F. A. P. de. **Especificação e implementação de um protótipo de serviço *web* para buscas baseadas em contextos compartilhados definidos a partir de sintagmas e relacionamentos.** Tese (Pós-Graduação em Engenharia Elétrica e de Computação) –

Universidade Federal do Rio Grande do Norte, Natal, 2007. Disponível em: <ftp://ftp.ppgee.ufrn.br/Mestrado/M196.pdf>. Acessado em: abril 2011.

REATEGUI, E; BOFF, E.; VICCARI, R. M. **Proposta e Avaliação Preliminar de um Assistente Virtual para Recomendação de Conteúdos.** In: Simpósio Brasileiro de Informática na Educação, 2005, Juiz de Fora. Proceedings XVI Simpósio Brasileiro de Informática na Educação, 2005. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/viewFile/432/418>. Acessado em: abril 2011.

RICOTTA, F. C. M. **Como os search engines funcionam?** Projeto Final de Graduação – Universidade Federal de Itajubá, Itajubá, 2007. Disponível em: <http://www.fabioricotta.com/wp-content/uploads/2007/11/tcc_fabio_ricotta.PDF>. Acessado em: abril 2011.

RODRIGUES, T. G. **Um Sistema de Apoio à Recuperação de Informação na *Web* voltado à Segurança de Redes e Sistemas.** Monografia (Graduação em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2009. Disponível em: <http://www.cin.ufpe.br/~tg/2009-2/tgr.pdf>. Acessado em: abril 2011.

SALTON, G.; McGILL, M. J. **Introduction to modern information retrieval.** New York, McGraw-Hill, 1983. Disponível em: <http://lyle.smu.edu/~mhd/8337sp07/salton.pdf>. Acessado em: abril 2011.

SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. **E-Commerce Recommendation Applications:** Data Mining e Knowledge Discovery, Massachusetts, n. 5, jan. 2001. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.43.1280&rep=rep1&type=pdf>. Acessado em: abril 2011.

SCHUTZE, H. et al. **A Comparison of Classifiers and Document Representations for the Routing Problem.** In: Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, SIGIR, 1995, Nwe York: ACM Press, 1995. Disponível em: <http://lvk.cs.msu.su/~bruzz/articles/relevance_feedback/schutze95comparison.pdf>. Acessado em: abril 2011.

SHARDANAND U.; MAES, P. **Social Information Filtering:** Algorithms for Automating "Word of Mouth". In: ACM CHI'95 Conference on Human Factor in Computing Systems, Proceedings, (1), p. 210-217, 1995. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/*download*?doi=10.1.1.95.586&rep=rep1&type=pdf>. Acessado em: abril 2011.

SILLA JR., C. N.; KAESTNER, C. A. A. **Estudo de Métodos Automáticos para Sumarização de Textos.** In: Simpósio de Tecnologias de Documentos, 2002, São Paulo. Anais do STD 2002. São Paulo: ITS, 2002, v. 1, p. 45-49. Disponível em: <http://sites.google.com/site/carlossillajr/files/2002-STD.pdf?attredirects=0>. Acessado em: abril 2011.

SILVA, A. B. M., PORTUGAL, M. S., CHECHIN, A. L.**Redes Neurais Artificiais e Análise de Sensibilidade:** Uma aplicação à demanda de importações brasileira. Revista de Economia Aplicada. São Paulo, v. 5, n. 4, p. 645-693, 2001. Disponível em: <http://www.ufrgs.br/ppge/pcientifica/2000_11.pdf>. Acessado em: abril 2011.

SILVA, F. R.G. **Geodiscover:** mecanismo de busca especializado em dados geográficos. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisa Espaciais, São José dos Campos, 2006. Disponível em: <http://urlib.net/rep/sid.inpe.br/MTC-m13@80/2006/11.07.13.25?languagebutton=en>. Acessado em: abril 2011.

SOARES, F. A. **Mineração de Textos na Coleta Inteligente de Dados da Web.** Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em: <http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0621324_08_pretextual.pdf>. Acessado em: abril 2011.

STEINBACH, M.; KARYPIS, G.; KUMAR, V. **A Comparison of Document Clustering Techniques.** In KDD Workshop on TextMining, 2000. Disponível em: <http://rakaposhi.eas.asu.edu/cse494/notes/clustering-doccluster.pdf>. Acessado em: abril 2011.

TICOM, A. A. M. **Aplicação das técnicas de mineração de textos e sistemas especialistas na liquidação de processos trabalhistas.** Dissertação (Mestrado em Ciências em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007. Disponível em:

<http://www.coc.ufrj.br/index.php?option=com_docman&task=doc_*download*&gid=1663>. Acessado em: abril 2011.

WILEY, J.; SONS. **Data Mining:** Concepts, Models, Methods, and Algorithms. Wiley-IEEE Press. 2000.

WIVES, L. K. **Utilizando conceitos como descritores de textos para o processo de identificação de conglomerados (clustering) de documentos.** Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. Disponível em: <http://www.leandro.wives.nom.br/pt-br/publicacoes/Tese.pdf>. Acessado em: abril 2011.