6
Metodologia Proposta

Neste capítulo são apresentados a metodologia proposta, a implementação e o desenvolvimento de um sistema para a coleta de dados inteligente na *Web*, seguido de aplicação prática em um estudo de caso.

6.1.
Proposta

Atualmente, encontrar informação relevante na *Web* não é uma tarefa trivial. Avanços tecnológicos constantes das máquinas de buscas permitem que cada vez mais informação esteja disponível. Entretanto, disponibilidade não é garantia de relevância. Além disso, os mecanismos atuais de pesquisa na *Web* funcionam baseados na consulta por palavras-chave. Expressar por termos a necessidade de informação nem sempre é algo possível para todos os usuários.

Uma situação muito comum, nos dias de hoje, é possuir um ou mais documentos textuais sobre determinados tópicos e desejar encontrar na grande rede outros documentos semelhantes. Em qualquer máquina de busca, encontrar *sites* sobre o assunto não é problema. O problema é encontrar a informação desejada em uma lista com mais de **576 milhões**[24] de *web sites* disponíveis. Refinar os critérios de busca através do uso de termos mais restritivos auxilia a reduzir este número (**320 mil**[25] *web sites*), mas, não o torna humanamente tratável.

Algumas abordagens para a coleta específica de documentos na *Web* recorrem às técnicas de rastreamento propostas para um *crawler* focado (ver capítulo "5"). Um *crawler* focado é altamente efetivo na construção de coleções de documentos de qualidade sobre tópicos específicos e oriundos da *Web*. Porém,

[24] Consulta realizada pelo termo "casa" no *search engine* Google, em 20/07/2008.

[25] Consulta realizada pelos termos "casa", "madeira" e "rj" no *search engine* Google, em 20/07/2008.

a política de rastreamento utilizada em um *crawler* focado é baseada principalmente no treinamento de um classificador que irá julgar a que classe ou tópico as características daquele documento pertencem.

Técnicas de Aprendizado de Máquina, tais como um classificador, necessitam de treinamento. E para que o aprendizado seja efetivo é necessário um bom conjunto de exemplos. Um bom conjunto de exemplos deve ser composto de um elevado número de elementos, muitas vezes balanceados, o que nem sempre é possível. Além disso, consideram apenas os aspectos estatísticos de um documento, desprezando a vasta informação semântica que pode auxiliar no processo de tratamento textual.

Somado a isto há o fato de *crawlers* focados não oferecerem suporte a métodos que auxiliem o tratamento da informação que venha a ser recuperada da *Web*, como por exemplo, indicação dos documentos mais representativos daquela coleção de textos.

Portanto, dado um documento exemplo e a necessidade de se obter da *Web* informação relevante sobre o mesmo, propõe-se uma metodologia de coleta inteligente de dados na *Web*, baseada em Mineração de Textos, capaz de atender a necessidade de informação do usuário sem o esforço de treinamento e ajustes requeridos por um *crawler* focado.

6.2. Metodologia Proposta

A metodologia proposta neste estudo tem como finalidade captar e analisar as características dos vários métodos disponíveis e avaliar suas potencialidades e limitações na coleta inteligente de dados.

Uma vez idealizado o objetivo de uma solução, uma técnica muito comum para a elaboração de uma metodologia é a de "dividir para conquistar". Esta técnica, empregada até mesmo em **algoritmos da Ciência da Computação**[26], consiste em dividir um grande problema em instâncias menores, tornando mais

[26] Um bom exemplo é o algoritmo de *merge sort* empregado na ordenação de grandes conjuntos de dados.

fácil a visualização de soluções para estas partes, e que unidas, solucionam o problema global apresentado inicialmente.

Ao modularizar o problema exposto neste trabalho em duas partes, chega-se a duas necessidades: a primeira delas é buscar informações específicas na *Web*, e a segunda, analisar e extrair as principais características de um documento que será apresentado ao sistema.

Obrigatoriamente, todo processo que se destine a realizar a coleta de dados no ambiente da *Web* faz uso de um *crawler* (HEATON, 2002). Não diferente, a abordagem proposta neste trabalho inicia-se com o planejamento de um *crawler*. Em razão da grande heterogeneidade deste ambiente, diferentes estratégias podem ser empregadas no processo de construção de um *crawler* e na definição da política de seleção de conteúdo que será utilizada por este (item "4.4"). Dentre estas, quando se deseja obter informações específicas sobre um assunto, emprega-se a estratégia de *crawling* focado (capítulo "5"). Em razão das grandes demandas de um *crawler* deste tipo, a sua utilização torna-se restrita, mas, as técnicas de rastreamento nele empregadas constituem importantes heurísticas a serem utilizadas.

Para analisar e extrair as principais características de um documento podem ser empregadas, principalmente, dois tipos de análise: Análise Estatística e Análise Semântica (item "2.5"). Em razão de considerar a linguagem com maior centralidade, a Análise Semântica obtém melhores resultados (ARANHA C. N., 2007), apesar das exigências de conhecimento lingüístico e maior esforço computacional, e será empregada neste estudo.

Após o levantamento destas duas principais necessidades, chega-se a metodologia ilustrada na Figura 40. Como notado, esta metodologia de coleta inteligente de dados na *Web* possui dois módulos: um módulo *off-line* e outro *on-line*. Estes dois módulos serão abordados em detalhes nos próximos itens.

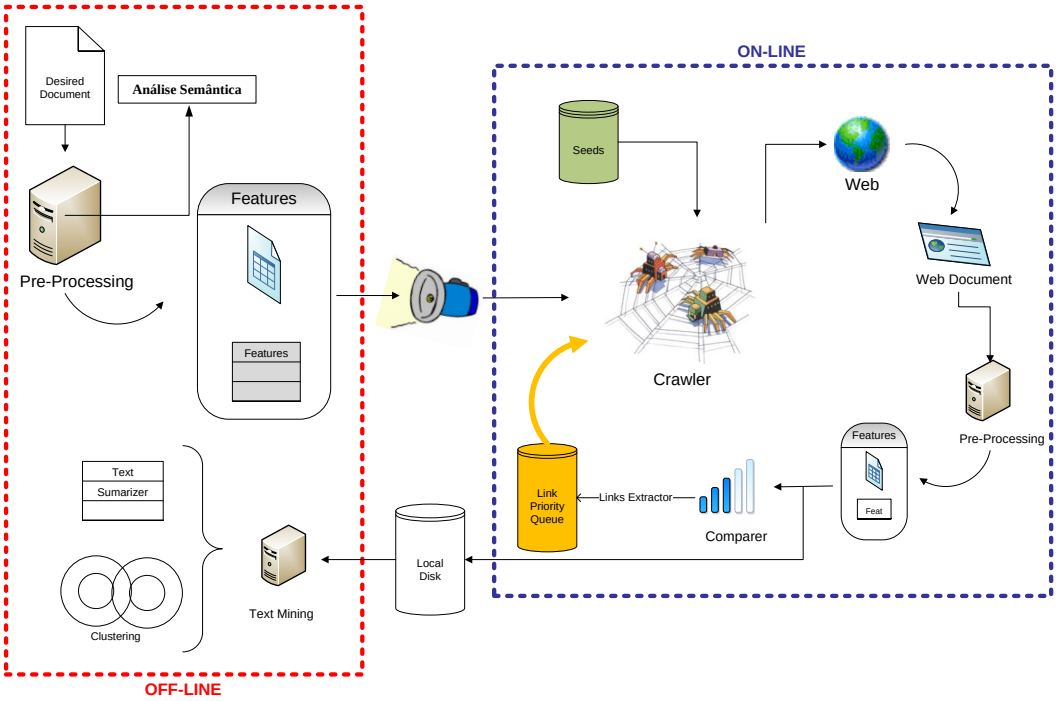

Figura 40 - Metodologia proposta para a Coleta Inteligente de Dados

6.2.1. Módulo *Off-Line*

Neste módulo são realizadas as principais operações de Mineração de Textos propriamente ditas. Engloba dois grupos de operações principais: Pré-Processamento de documentos (parte de um processo de Mineração de Textos) e Mineração de Textos para Clusterização de Documentos. Recebe a denominação de *off-line* por realizar tarefas que não necessitam de conexão com a Internet. A Figura 41 ilustra este processo em maiores detalhes.

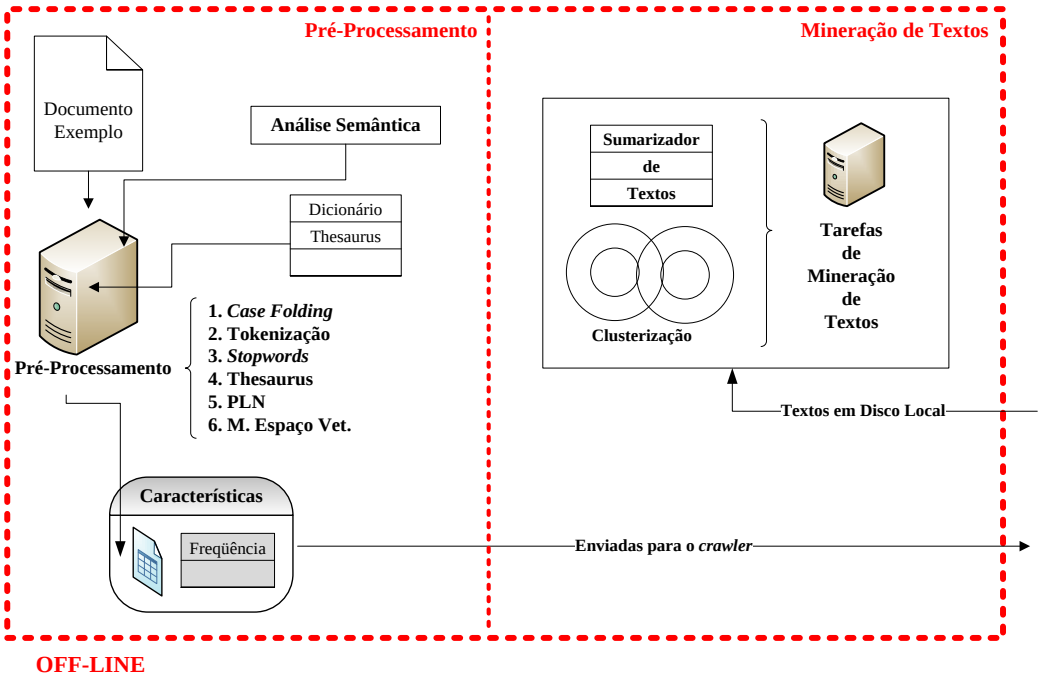

Figura 41 - Módulo Off-Line da metodologia proposta

6.2.1.1. Pré-Processamento

Inicialmente, as operações de Pré-Processamento são responsáveis por analisar semanticamente o documento exemplo. Este processo também ocorre igualmente no módulo *On-Line*, porém, os documentos analisados são aqueles recuperados da *Web* pelo *crawler*. Ao submeter um documento para análise, as seguintes operações de pré-processamento (item "3.2") são realizadas: *Case Folding*, Tokenização, Remoção de *Stopwords*, substituição de termos por consulta ao Dicionário *Thesaurus*, Processamento de Linguagem Natural e criação do Modelo de Espaço Vetorial (item "4.3.2") representativo do documento.

O uso da técnica de *Case Folding* consiste em converter todas as letras de um documento em maiúsculas ou minúsculas. Na metodologia proposta neste trabalho, a utilização desta técnica é justificada pelo fato de não ser objetivo deste sistema de Mineração de Textos o reconhecimento de entidades no documento. Desta forma, a presença de termos ora, em letras maiúsculas, ora, em letras minúsculas, só iria acrescentar maior custo ao processo de comparação de termos com o Dicionário *Thesaurus* e não acrescentaria qualquer ganho de informação relevante sobre os termos encontrados no texto.

A etapa de Tokenização segue a metodologia de geração de *tokens* (ver Figura 14) proposta em (KONCHADY, 2006). Esta metodologia, fortemente apoiada na utilização de um dicionário de palavras, obtém *tokens* com alto valor semântico, conforme resultados obtidos nos experimentos realizados no Estudo de Casos. Esta metodologia é composta dos seguintes passos:

1. Geração simples de *tokens:* baseada no conjunto de *tokens* delimitadores, como espaço e fim de linha.
2. Identificação de abreviações: realizada com auxílio de dicionários desenvolvidos para este propósito. Todas as abreviações encontradas são substituídas pelos seus termos não contraídos.
3. Identificação de palavras combinadas: muito comum em nomes próprios de organizações. Em geral, são palavras separadas por símbolos como o "&": "Casa & Vídeo".
4. Identificação de símbolos de Internet: geralmente, símbolos de internet atendem às normas regras estabelecidas no momento de criação do serviço. Desta forma, o uso de expressões regulares auxilia na identificação de tais símbolos (JARGAS, 2006). Alguns exemplos de símbolos de Internet são IP, *e-mails* e *URLs*. Detalhe importante é sobre a distinção entre *URLs* no texto e *links*. Quando na forma de *links*, *URLs* são recuperados na etapa de *Parsing*, uma das fases do processo de *crawling* e que será abordada em detalhes no item.
5. Identificação de números: este processo é realizado pela verificação de conteúdo numérico em meio aos *tokens*. Números na forma extensa, quando identificados, são convertidos em formato numérico.
6. Identificação de *tokens* multi-vocabulares: realizado também com forte apoio de um dicionário de termos. Busca reunir em um único *token* palavras que, quando utilizadas em conjunto, transmitem idéias diferentes ou incompletas quando utilizadas separadas. Exemplos comuns são "bolsa de valores", "casa da moeda", "cachorro de rua".

Em seguida, inicia-se o processo de remoção de *stopwords* com o objetivo de reduzir a grande dimensionalidade de dados textuais através da remoção de

palavras de baixo poder discriminatório. Uma lista de aproximadamente cem *stopwords* da Língua Portuguesa pode ser encontrada em: "http://linguateca.di.uminho.pt/Paulo/stopwords/folha.MF100.txt".

a	da	em	já	nos	quando	Sua
à	das	entre	local	o	que	Também
ainda	de	era	maior	ontem	quem	Tem
ano	depois	está	mais	os	r	Ter
anos	deve	estado	mas	ou	rio	Todos
ao	dia	estão	mercado	país	são	Três
aos	disse	eu	mesmo	para	se	Um
apenas	diz	foi	mil	paulo	segundo	Uma
as	do	folha	milhões	pela	sem	Us
às	dois	foram	muito	pelo	ser	Vai
até	dos	governo	mundo	pessoas	será	
brasil	e	grande	na	pode	seu	
com	é	há	não	por	seus	
como	ela	hoje	nas	porque	só	
contra	ele	isso	no	presidente	sobre	

Tabela 6 - Lista de stopwords utilizadas na etapa de Pré-processamento

Outro detalhe relacionado ao processo de remoção de *stopwords* é que a *stoplist* ideal deve ser construída levando em consideração o domínio do problema. Por exemplo, no caso de um sistema de Mineração de Textos para a área médica, termos como "exame" e "medicamento" poderiam ser incluídos na lista de *stopwords*, visto que, estas palavras, provavelmente, possuirão pouco poder discriminatório em relação a outros termos.

Portanto, assim que for obtida a coleção de documentos sob a qual será realizado o processo de Mineração de Textos, mesmo após a remoção das *stopwords* consideradas padrão, é interessante que seja analisada a distribuição das freqüências de ocorrência dos termos que estão presentes na coleção. De acordo com (SALTON & BUCKLEY, 1988), termos com elevada freqüência em muitos documentos constituem elementos de pouco poder discriminatório.

Após a identificação de *tokens*, foi utilizado o algoritmo de Distância de Edição para realizar possíveis correções ortográficas nos documentos de exemplo

e nos documentos recuperados na *Web*. Para maiores detalhes sobre este algoritmo, consulte (FONSECA & REIS, 2002).

O passo seguinte é a utilização do dicionário *Thesaurus* (item "3.3.2") para substituir termos que possuam valores semânticos semelhantes ou relacionados, mas, são grafados de maneira distinta, sejam estes sinônimos ou exemplos de sinonímia. A utilização deste dicionário evita equívocos ao realizar o cálculo de freqüência destes termos. Para exemplificar, podemos supor que em determinado documento sobre animais, encontramos dois termos de grande freqüência: cão e cachorro. Porém, como possuem grafias distintas, estes termos serão computados separadamente, ainda que pelo fato de transmitirem a mesma idéia, deveriam ser considerados como somente um termo e ter freqüências e outras métricas somadas. Outro benefício proporcionado por estes dicionários é relação entre termos. Retornando ao exemplo acima, com a ajuda do dicionário, percebe que cachorro e cão possuem valores semânticos aproximados e pertencem a uma categoria superior: a de animais de estimação que também pertence à categoria de animais, e esta, a de mamíferos. O dicionário *Thesaurus* construído para este estudo de casos possui cerca de três mil termos preferenciais e dez mil termos não-preferenciais, todos classificados em dezenas de categorias. A estrutura de dados utilizada na construção deste dicionário é ilustrada na Figura 42.

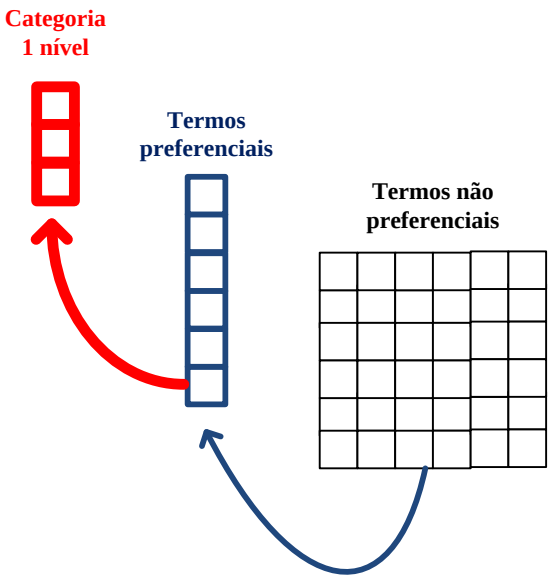

Figura 42 - Estrutura do dicionário Thesaurus utilizado no Sistema de MT

Além desta técnica de Processamento de Linguagem Natural oferecida pelo uso de *Thesaurus*, também é aplicada a técnica de *stemming*. O algoritmo de *stemming* utilizado foi o de (PORTER, 1980) adaptado para a Língua Portuguesa. Para efeitos comparativos, também é apresentado um estudo de casos sem a utilização da técnica de *stemming.*

Em seguida, inicia-se o processo de construção da modelo de representação do documento. O modelo utilizado é o de Espaço Vetorial, em razão das diversas técnicas disponíveis para a comparação de similaridade entre documentos. Outra vantagem que justificou a escolha deste modelo é o fato deste modelo expressar a freqüência de ocorrência dos termos representados.

6.2.1.2. Mineração de Textos

As operações de Mineração de Textos são executadas, assim que é finalizado o processo de rastreamento, sobre os documentos recuperados pelo *crawler*. Todo documento recuperado pelo *crawler* é armazenado em disco já Pré-processado. Dentre a lista de tarefas possíveis para um sistema de Mineração de Textos, duas delas destacam-se como de grande utilidade ao problema proposto e são listadas no esquema gráfico da metodologia: Clusterização e Sumarização. Apenas a tarefa de Clusterização é implementada em razão das dificuldades existentes para realizar a avaliação da qualidade de um resumo (RINO & PARDO, 2003). A tarefa de Sumarização é indicada no esquema gráfico para demonstrar a possibilidade de realização desta tarefa.

O algoritmo de Clusterização utilizado é baseado no *K-means* (HARTIGAN & WONG, 1979), porém, com uma pequena adaptação necessária para a tarefa de agrupamento de textos: o cálculo de distância, tradicionalmente realizado pela Distância Euclidiana, é realizado pelo Método do Cosseno, apresentado no item "4.3.2.2".

A execução deste algoritmo foi realizada na plataforma MatLab. MatLab é um software confiável e de alta performance voltado para o cálculo numérico e possui ferramentas para análise visual de *clusters*. Para a utilização do MatLab neste processo, fez-se necessário incluir no sistema proposto um processo para

exportação dos documentos que serão clusterizados. Por ser apenas uma funcionalidade adicional e que não interfere nos resultados obtidos, esta não é considerada parte da metodologia proposta.

6.2.2. Módulo *On-Line*

Este módulo é composto basicamente pela operação de *crawling* inteligente. Recebe a denominação de *on-line* por realizar tarefas que possuem como ambiente de trabalho a Internet, especificamente, a *Web*. A Figura 43 ilustra este processo em maiores detalhes.

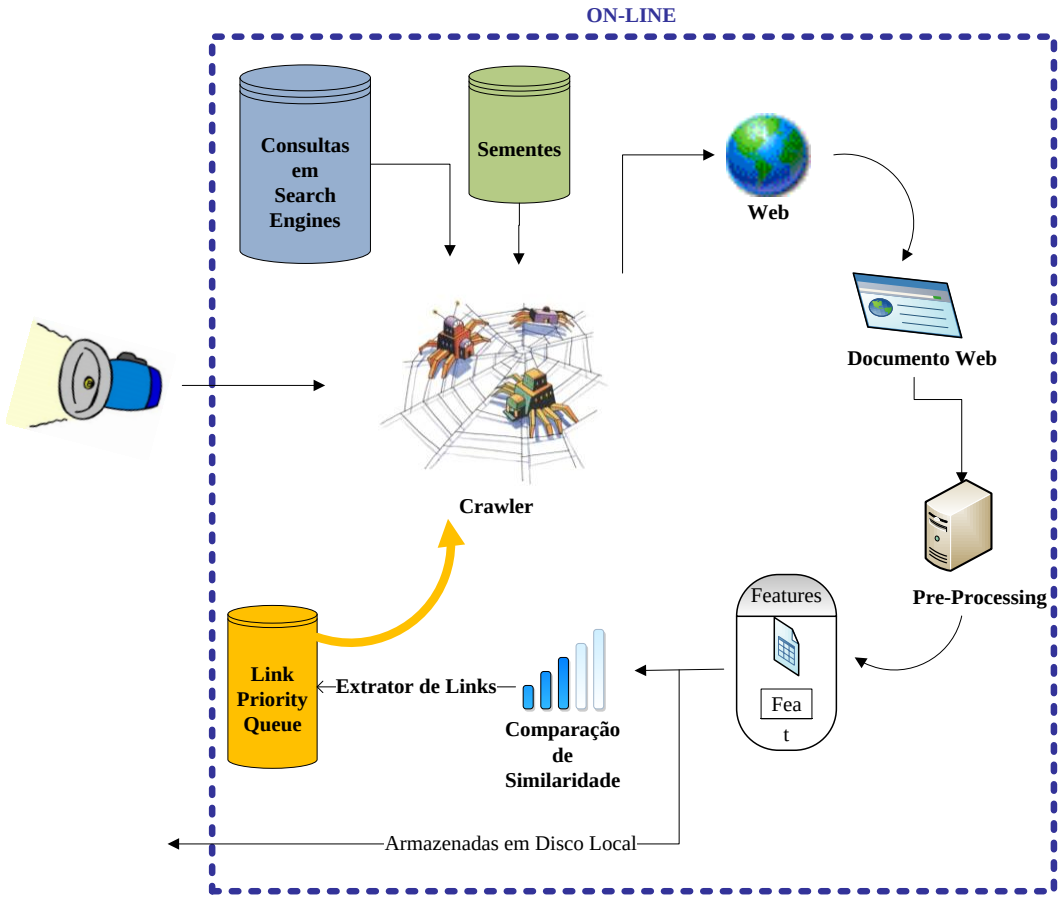

Figura 43 - Módulo On-Line da metodologia proposta

O processo é iniciado com o envio das informações obtidas pela operação de pré-processamento do documento exemplo executada fase anterior. Uma vez

recebida estas informações, o processo de *crawling* pode ser iniciado. Para iniciar um processo de rastreamento na *Web*, todo *crawler* precisa de um ponto de partida. Nesta abordagem, o ponto de partida para o início do processo de rastreamento pode ser fornecido pelo usuário através de *URLs* relacionadas ao assunto desejado (sementes) ou, caso isto não seja possível, o próprio *crawler* pode encontrar um ou mais pontos de partidas através da realização de consultas nas máquinas de busca disponíveis.

Para que seja realizado um procedimento de consulta em uma máquina de busca (item "4.4.5") é necessário que a necessidade de informação do usuário seja expressa por meio de palavras-chaves. O *crawler* proposto neste trabalho, assim que analisa as informações obtidas pela etapa de Pré-processamento do documento exemplo, seleciona os termos de maior freqüência deste documento, e então, submete estes termos para as máquinas de busca. A quantidade de termos submetidos é alterada durante a execução do processo de rastreamento. Busca-se a relação ideal entre quantidade de documentos retornados pela máquina de busca e relevância ao documento exemplo.

Após iniciado o processo de rastreamento, todo documento *web* recuperado é analisado (pré-processado) da mesma forma que o documento exemplo. Ao final deste processo, o documento recuperado é submetido a um processo de comparação de similaridade com o documento exemplo.

Em seguida, são extraídos os *hyperlinks* do documento recuperado da *web*. Estes *hyperlinks* são analisados por outro componente do *crawler*, que possui função semelhante a do destilador proposto em (DOM, CHAKRABARTI, & BERG, 1999), e abordado no capítulo "5": atribuir um grau de importância a cada *hyperlink*. Este grau de importância é resultado da média aritmética entre o grau de similaridade do documento recuperado e as características individuais de cada *hyperlink*. Cada *hyperlink* é enfileirado de acordo com o seu grau de importância: quanto mais importante (relevante) for um *hyperlink*, mais rápido ele será retirado da fila e visitado. Assim que são extraídos e graduados os *hyperlinks* de um documento, o *crawler* retira da fila outro *hyperlink* (o mais importante) e inicia novamente o processo iterativo de recuperação (visita), pré-processamento, comparação e atribuição de importância aos *hyperlinks* deste outro documento.

O processo de rastreamento será finalizado pelo usuário ou quando algum critério de parada definido pelo mesmo for alcançado.

6.3. Implementação

6.3.1. Ambiente de Desenvolvimento

A metodologia proposta neste trabalho foi implementada na linguagem de programação **C#**[27]**.** C# é uma linguagem de programação orientada a objetos criada pela empresa Microsoft e faz parte da plataforma de desenvolvimento ".NET". Embora existam mais de vinte linguagens de programação suportadas pela tecnologia ".NET", a linguagem C#, baseada na linguagem C++ e Java, é considerada a linguagem símbolo da plataforma ".NET".

Com uma idéia semelhante a da plataforma Java, o ambiente de desenvolvimento de aplicações ".NET" permite que *softwares* sejam desenvolvidos para qualquer sistema operacional que suporte o .NET *Framework*.

Portanto, constituem justificativas para a escolha deste Ambiente de Desenvolvimento e Linguagem de Programação a orientação a objetos que permite fácil reuso, bem como grande organização, do código e a alta portabilidade para diversos Sistemas Operacionais.

6.3.2. Arquitetura Geral do Sistema

No sistema desenvolvido para aplicação da metodologia proposta nesta Dissertação, todas as funcionalidades de maior nível hierárquico foram implementadas em módulos.

Na Figura 44 está ilustrada a seqüência das funções executadas na análise do documento exemplo (módulo *off-line*). Na Figura 45 estão ilustradas as atividades desempenhadas pelo *crawler* para iniciar o processo de rastreamento.

---

[27] Pronuncia-se C Sharp.

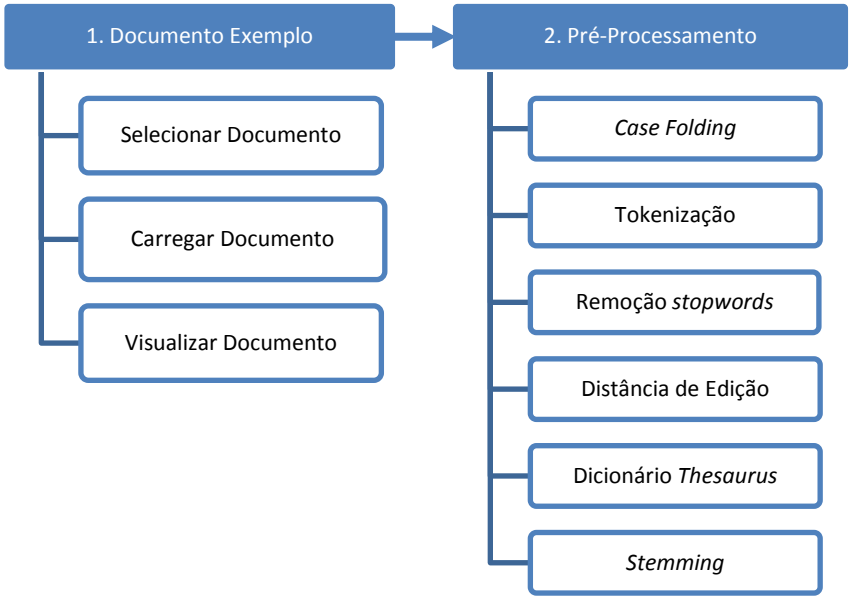

Figura 44 – Seqüência de ações executadas na análise do documento exemplo

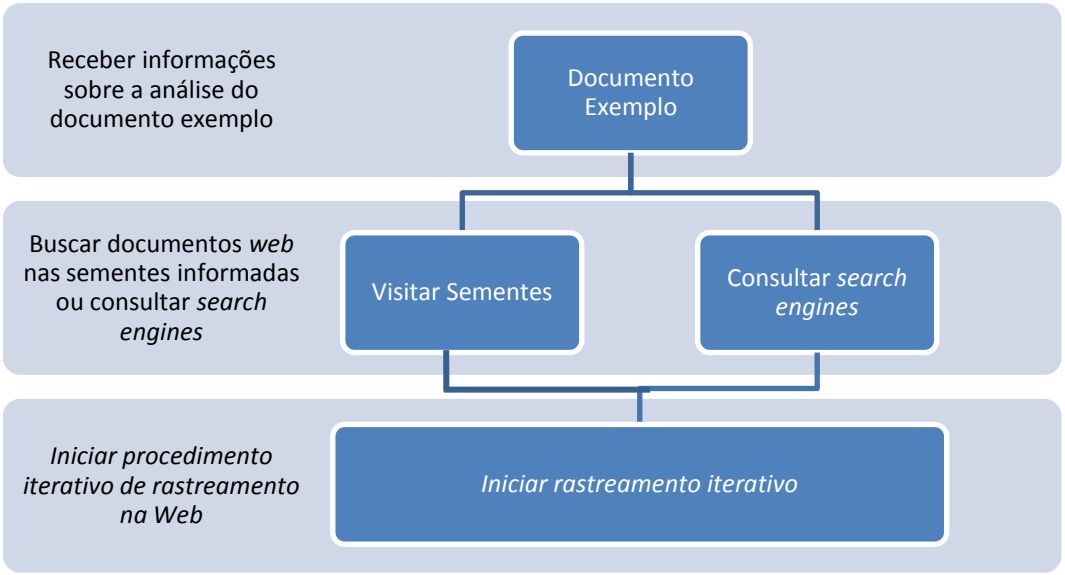

Figura 45 – Atividades desempenhadas pelo *crawler* para iniciar o rastreamento

Na Figura 46 são detalhas as atividades executadas no procedimento de rastreamento propriamente dito.

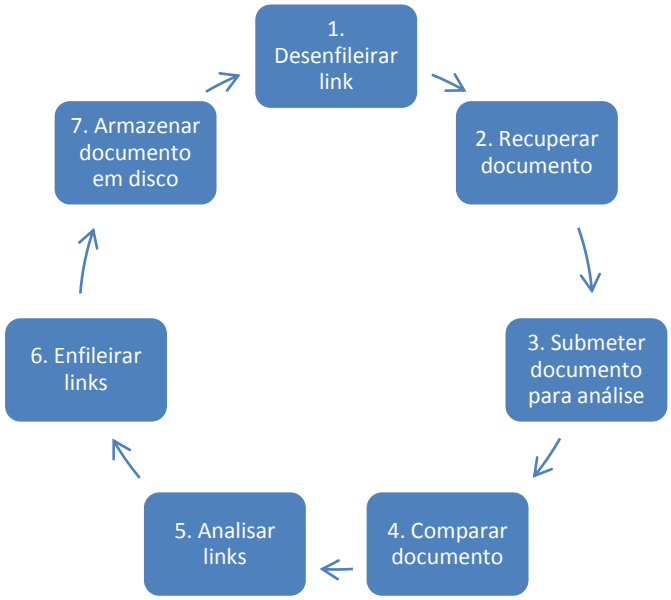

Figura 46 - Procedimento iterativo de rastreamento do crawler

Na Figura 47 são exibidos os processos necessários para a realização da atividade de Mineração de Textos quando o processo de rastreamento é finalizado.

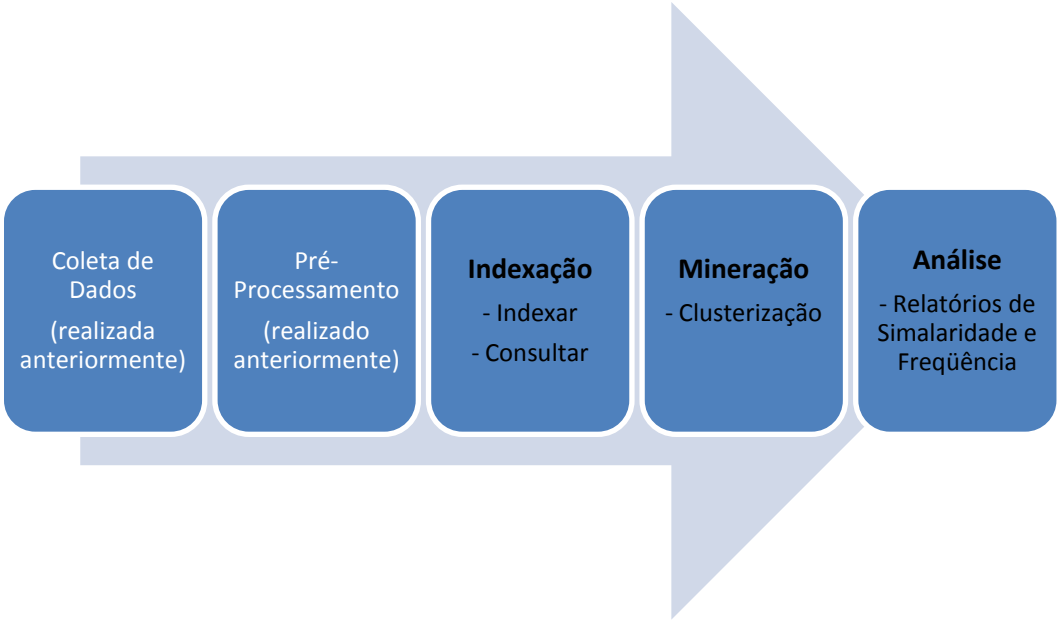

Figura 47 - Atividades executadas na etapa de Mineração de Textos

6.3.3. Multithreading

Em um Sistema Operacional, cada *software* em execução é tratado como um processo. Processo é um local da memória em que se é permitido acessar recursos em concorrência com outros processos sem que um interfira na execução do outro.

Em geral, cada processo possui uma única *thread* que é responsável por executar as operações necessárias para o seu funcionamento. Desta forma, para que uma nova operação possa ser realizada é necessário que a operação iniciada anteriormente pela *thread* tenha sido concluída.

A tarefa de coleta de dados na *Web* é um processo lento, mas, que exige pouco uso de processamento computacional. Recuperar um documento *web* e aguardar a sua chegada é desperdiçar valiosos ciclos de processamento com inatividade. A fim de aperfeiçoar o uso do processamento, o *crawler* utilizado neste estudo foi desenvolvido com o recurso de *multithreading*, isto é, a divisão da execução das tarefas de um processo em diversas *threads*. Com a utilização deste recurso, um único processo do *crawler* funciona como se muitos *web crawlers* estivessem agindo simultaneamente, garantido que todo tempo de processamento seja empregado adequadamente e que toda a largura de banda de rede disponível seja utilizada.

6.4. Estudo de Casos

Para a realização deste estudo de caso, documentos apresentados como exemplo foram textos jornalísticos obtidos aleatoriamente do corpus CETENFolha. CETENFolha, Corpus de Extratos de Textos Eletrônicos NILC/Folha de S. Paulo, é um corpus de cerca de vinte e quatro milhões de palavras em Língua Portuguesa do Brasil, criado pelo projeto de Processamento Computacional do Português com base nos textos do jornal Folha de S. Paulo. Este corpus possui aproximadamente 341 mil porções de textos, classificados por semestre e caderno do jornal do qual provêm. Informações adicionais sobre este corpus podem ser visualizadas na Tabela **7**.

Quantidade de palavras	24 milhões
Idioma	Português (Brasil)
Origem	Textos publicados no Jornal Folha de S. Paulo
Período de Publicação	1994
Quantidade de edições	365
Classificado?	Pelo tipo de caderno a que pertence a notícia
Etiquetado?	Não

Tabela 7 - Informações adicionais sobre o CETENFolha

Na Figura 48 é possível visualizar um exemplo de texto jornalístico, em seu formato original, presente neste corpus. As *tags* “<ext>” “</ext>” delimitam o início e fim de cada notícia ou documento; as *tags* “<t>” e “</t>” delimitam o título da notícia; as *tags* “<a>” e “” delimitam o autor da mesma; cada parágrafo é identificado pelas *tags* “ ” e “ ”; e cada sentença da notícia fica compreendida entre as *tags* “<s>” e “</s>”.

```
<ext id=1 cad="Opinião" sec="opi" sem="94a">
 <s>
  <t> PT no governo </t>
 </s>
 <s>
  <a> Gilberto Dimenstein </a>
 </s>
 <p>
  <s> BRASÍLIA Pesquisa Datafolha publicada hoje revela um dado supreendente:
recusando uma postura radical, a esmagadora maioria (77%) dos eleitores quer o PT
participando do Governo Fernando Henrique Cardoso . </s>
  <s> Tem sentido -- aliás, muitíssimo sentido . </s>
 </p>
 <p>
  <s> Muito mais do que nos tempos na ditadura, a solidez do PT está, agora,
ameaçada . </s>
  <s> Nem Lula nem o partido ainda encontraram um discurso para se diferenciar . </s>
  <s> Eles se dizem oposição, mas ainda não informaram o que vão combater . </s>
  <s> Muitas das prioridades do novo governo coincidem com as prioridades do PT . </s>
 </p>
</ext>
```

Figura 48 - Exemplo de documento do corpus CETENFolha

A maior parte das aplicações desenvolvidas para Mineração de Textos é realizada sobre corpus de textos em Língua Inglesa e este fato constituiu o principal motivo para a escolha deste corpus. Além disso, por ser de conteúdo antigo, a dificuldade encontrada para localizar informação relevante sobre estes

documentos será maior e constituirá uma boa prova de teste para a metodologia proposta.

Foram selecionados diversos documentos, ou seja, notícias deste corpus que possuem mais de seiscentas palavras para o estudo de casos. Os resultados apresentados neste trabalho expressam a média do desempenho de todos os experimentos realizados individualmente.

Assim que apresentado o documento exemplo ao sistema, são realizadas as operações de Pré-Processamento, descritas no item "6.2.1.1" e ilustradas na Figura 44 (*Case Folding*, Tokenização, Remoção de *Stopwords*, Correção Ortográfica, Uso do Dicionário *Thesaurus* e *Stemming).*

Após a etapa de Pré-Processamento, é realizada a análise da distribuição das freqüências de ocorrência dos termos no documento. Este procedimento, além de selecionar os termos mais relevantes do documento e que serão utilizados em uma possível consulta nas máquinas de busca disponíveis, também permite identificar termos irrelevantes e que não constam na *stoplist* padrão. O sistema oferece duas formas de visualização desta informação. A primeira, ilustrada na Figura 49, é a distribuição geral de ocorrência dos termos no documento. A segunda, representada na Tabela 9, é uma lista com o número de ocorrências do termo e o próprio termo, ordenada decrescentemente pelo número de ocorrências.

Ocorrências	Termo
233	Internet
214	web
...	...
100	conexão
...	...
1	cabo

Tabela 9 – Análise pontual da distribuição de freqüências de *tokens* de um documento

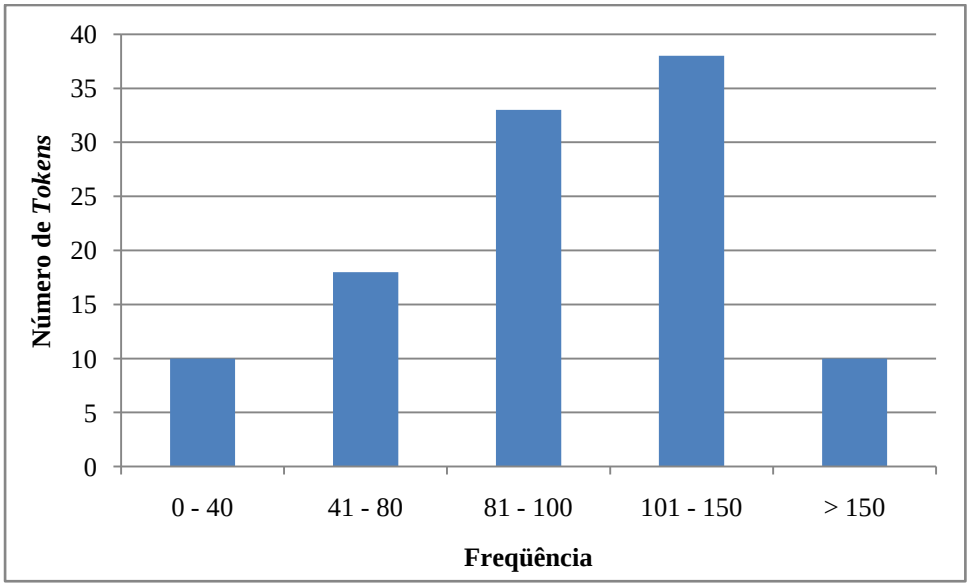

Figura 49 – Análise geral da distribuição de freqüência de *tokens* de um documento

O conjunto de palavras-chaves selecionado para a pesquisa em mecanismos de busca geralmente é composto pelos *tokens* que possuem a maior freqüência no documento. No exemplo acima, este conjunto seria composto pelos termos que possuem freqüência maior do que cento e cinqüenta. Convém ressaltar que nem sempre *tokens* com freqüências maiores possuem grande caráter discriminante. É possível que freqüências altas para *tokens* sejam em razão da não eliminação de *stopwords* relevantes ao contexto da aplicação.

Apesar de uma pequena quantidade de termos (dez mil termos ou 5 por cento de um dicionário comum) em sua base de dados, o dicionário *Thesaurus* foi capaz de auxiliar na consulta semântica de aproximadamente trinta por cento dos *tokens* presentes nos documentos exemplo (Figura 50).

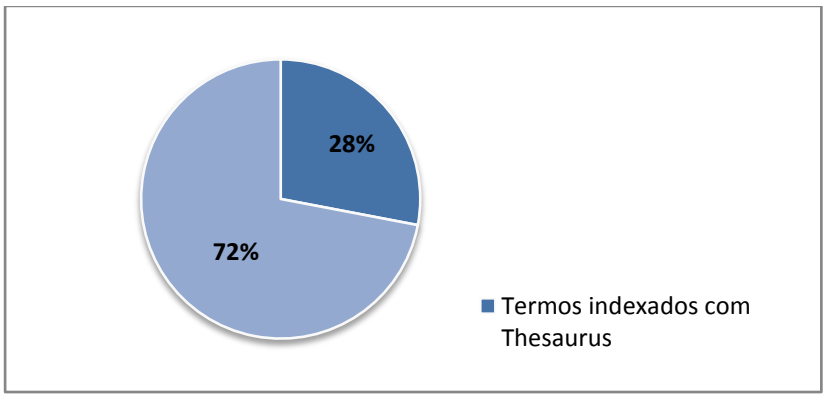

Figura 50 - Consultas realizadas com auxílio do dicionário Thesaurus

Outro detalhe relevante ao uso do dicionário *Thesaurus* é sobre o uso de s*temming*. O principal objetivo de utilização desta técnica é a redução do número de *tokens* através da identificação de palavras que possuam mesmo radical. Entretanto, a utilização desta técnica provou-se pouco eficiente neste estudo de caso, reduzindo, em média, aproximadamente cinco por cento da quantidade de *tokens* de um documento, conforme ilustrado na Figura 51. Com o uso do dicionário *Thesaurus* antes da aplicação do algoritmo de *stemming*, a maior parte dos termos que poderiam sofrer *stemming* foi previamente identificada e mapeada em um único termo.

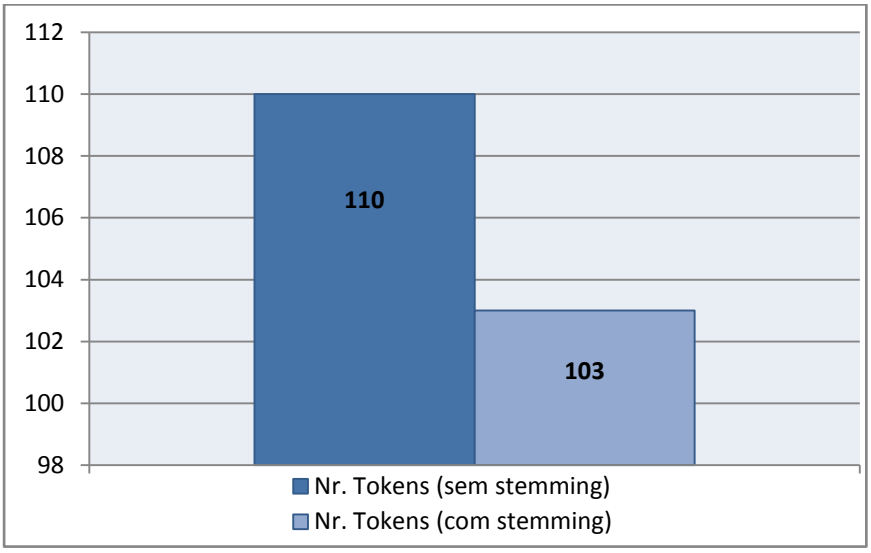

Figura 51 – Redução do número de tokens com *stemming*

A aplicação do algoritmo de Distância de Edição foi responsável por identificar erros de ortografia em dezessete por cento dos documentos, conforme ilustrado na Figura 52. No total, sete por cento dos *tokens* verificados apresentavam erros de grafia. Porém, como o processo de verificação compara os termos encontrados nos documentos com os termos do dicionário de dados, não se pode afirmar que, somente, dezessete por cento dos documentos apresentavam erros de ortografia, pois, a quantidade de termos disponíveis no dicionário é insuficiente para assegurar tal afirmativa.

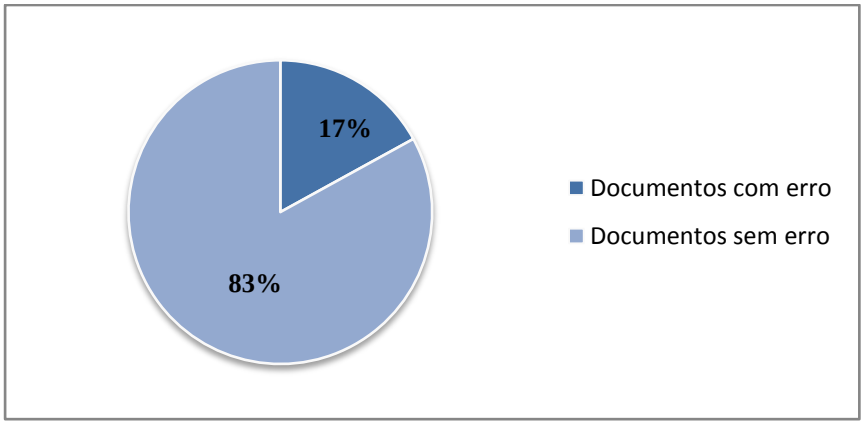

Figura 52 - Documentos com erros de ortografia

No total, foram realizadas consultas nos três maiores *search engines* à saber: "Google", "Yahoo!" e "MSN Live Search". O processo de consulta automática baseia-se na identificação e submissão dos *tokens* mais freqüentes nos documentos exemplo como palavras-chave para a realização da consulta. Foram testadas quantidades de termos entre um e dez. A quantidade de palavras-chave encontrada pelo mecanismo de ajuste automático do critério de consulta que retornou resultados mais satisfatórios foi de seis termos, conforme ilustrado na Figura 53. Esta conclusão é baseada na similaridade média dos quinze primeiros resultados retornados pela máquina de busca. Percebeu-se que, geralmente, consultas realizadas com uma quantidade de termos superior a seis são muito específicas em relação aos critérios definidos, o que leva a, ou obter respostas com taxa de similaridade média menor do que com seis termos, ou a não obtenção de resultados.

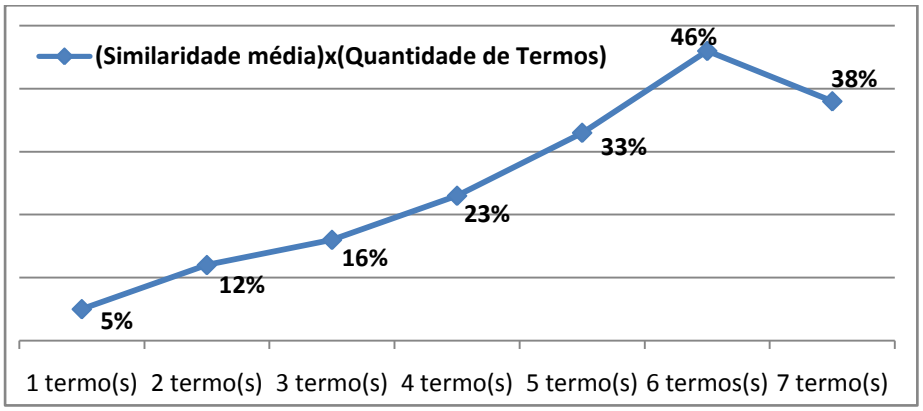

Figura 53 - Relação entre similaridade e quantidade de termos utilizada na consulta realizada nos *search engines*

A estratégia inteligente de *crawling* proposta mostrou-se confiável quando comparada a outras políticas de seleção de conteúdo para *crawling*. Convém ressaltar que utilizar as métricas de Precisão e Abrangência requer que os documentos recuperados sejam classificados em relevantes e não relevantes. Porém, segundo (HERLOCKER, KONSTAN, TERVEEN, & RIEDL, 2004), essa classificação é subjetiva, variando de usuário para usuário. Em geral, essas métricas são utilizadas em conjuntos de dados previamente classificados.

Em razão disto, para avaliar o desempenho da estratégia de *crawling* desenvolvida neste estudo optou-se pela utilização de uma métrica objetiva, tal como o cálculo da similaridade entre os documentos recuperados e o documento exemplo.

A Figura 54 mostra o comparativo de desempenho da estratégia de *crawling* baseado em MT e da estratégia de seleção de conteúdo *breadth-first* quanto à quantidade de dados relevantes recuperados em relação ao total de dados recuperados (em megabytes). Foi considerado relevante todo documento recuperado que apresentou similaridade superior a trinta por cento. A Figura 55 mostra o comparativo de desempenho da estratégia de *crawling* baseado em MT e da estratégia de seleção de conteúdo *breadth-first* quanto à similaridade média dos documentos recuperados ao longo do processo de rastreamento, desta vez, quantificado em número de *links* visitados. Todas as duas heurísticas, *crawler* proposto e *crawler breadth-first*, tiveram os mesmos pontos de partida (sementes).

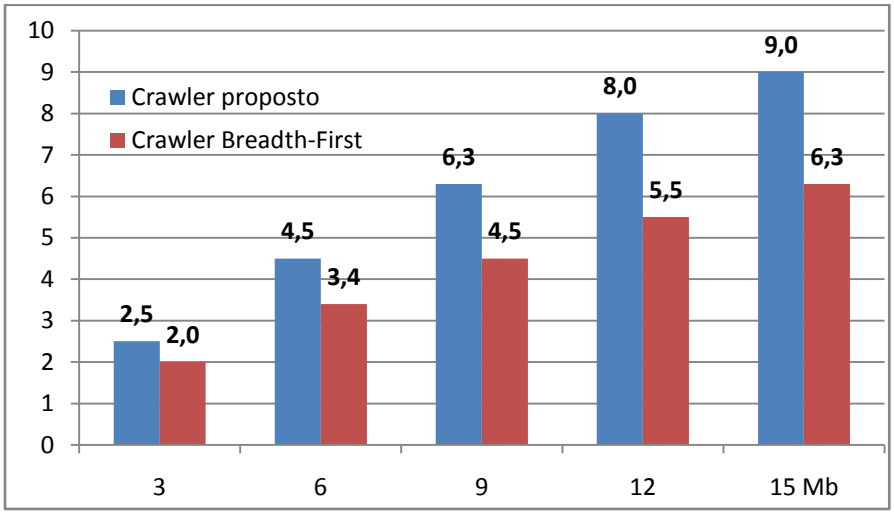

Figura 54 - Comparativo das técnicas de *crawling*

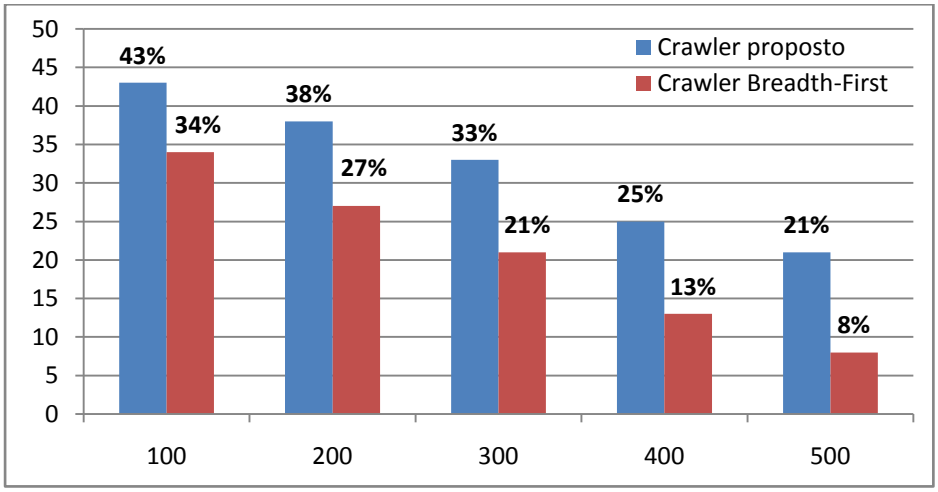

Figura 55 – Similaridade média dos documentos ao longo do processo de *crawling*

Ao término do processo de coleta de dados é realizada uma operação de Clusterização de Documentos. Clusterização é o processo de agrupamento de um conjunto físico ou abstrato de objetos em grupos similares. Uma tarefa de Clusterização bem executava deve gerar *clusters* que reúnam uma coleção de objetos de dados são similares entre si e diferentes de outros objetos dos outros grupos.

A tarefa de Clusterização empregada neste estudo possui como objetivo identificar uma quantidade previamente definida de documentos mais representativos e distintos de toda a coleção de documentos recuperados da *Web*. Os documentos mais representativos serão aqueles que estiverem mais próximos dos centróides dos *clusters*. A distância de medida utilizada foi Método do Cosseno (item "4.3.2.2"), pois, este representa com maior fidedignidade a similaridade entre documentos textuais do que a distância Euclidiana.

Neste estudo, para todos os documentos exemplo, foi definido como três o número de *clusters* a ser criado. Porém, como vários documentos exemplo foram submetidos ao sistema de coleta inteligente de dados, e, portanto, diversos documentos distintos foram recuperados da *Web*, torna-se difícil definir uma média entre distintos processos de Clusterização.

Um dos métodos utilizados para avaliar um processo de Clusterização consiste na visualização espacial do posicionamento dos dados em relação aos *clusters*. Na Figura 56 é exibido o posicionamento espacial resultante de um dos

processos de Clusterização empregados. Como pode ser notado, os documentos foram agrupados em *clusters* coesos e distantes entre si.

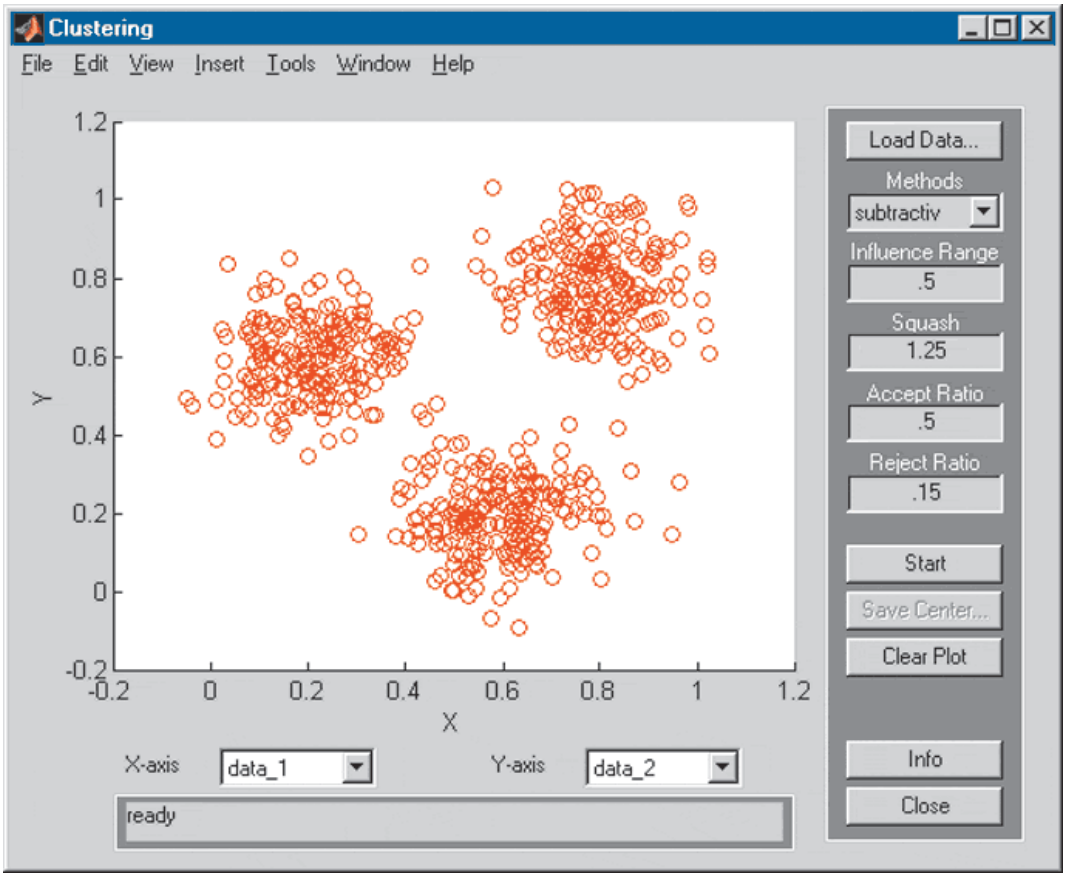

Figura 56 - Análise visual do processo de Clusterização