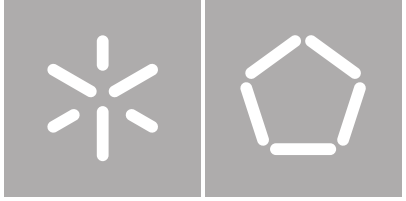
Universidade do Minho
Escola de Engenharia

Pedro Vasco Neto Pereira

Descarga Temporal de Páginas Web

Junho de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Pedro Vasco Neto Pereira

Descarga Temporal de Páginas Web

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Professor Joaquim Melo Henriques Macedo

Junho de 2013

Descarga Temporal de Páginas Web

Pedro Vasco Neto Pereira

Dissertação submetida à Universidade do Minho na área de
Engenharia Informática, para o grau de Mestre, sob a supervisão
científica do Professor Joaquim Macedo.

Universidade do Minho

Escola de Engenharia

Departamento de Informática

Junho, 2013

Agradecimentos

Gostaria de agradecer ao meu orientador, Professor Joaquim Macedo, pela sua orientação, disponibilidade e paciência no decurso da realização deste trabalho. A sua ajuda foi indispensável e contribuiu em muito para o enriquecimento da minha formação académica.

À Márcia por todo o apoio e compreensão.

Ao meu pai, à minha mãe e irmão, por todos os esforços que fizeram e que me permitiram chegar até aqui.

A todos os meus amigos.

Abstract

There is a plethora of information inside the Web. Even the most famous commercial search engines cannot download and index all available information. For this reason, from the last years until now, there are several research works on the design and implementation of focused crawlers in a particular topic, and also on geographic scope crawlers.

Those who follow carefully the research on the area of Web crawling are witnessing that the temporal dimension has not the importance it deserves in the literature.In the opposite direction, there is an increasing interest on time dimension in other areas of information retrieval namely retrieval models, result sets presentation, clustering, classification, and others.

Therefore, the challenge we have set ourselves in this work, was to develop a crawler whose purpose is to deal with time constraints. The importance of this dimension is certainly quite amplified when combined with the topic or geography, but now we wanted to study it in isolation.

The used approach is quite direct. It is based on an algorithm for temporal segmentation of Web pages and follows links only in segments within the temporal scope of the restriction.

This system is designed for Web pages written in Portuguese though its design philosophy can be applied to other languages.

In addition and for increase results effectiveness, the used algorithm prioritized the downloading of pages with more links within the temporal scope. The precision of results is around 75%.

Resumo

Existe uma infinidade de informações dentro da Web. Até mesmo os motores de busca mais famosos não podem descarregar e indexar toda a informação disponível. Por esta razão, desde há já alguns anos que há vários trabalhos de investigação sobre o desenho e implementação de robôs focados num tópico em particular mas também em robôs de âmbito geográfico.

Aqueles que seguem com atenção a investigação na área de descargas Web podem constatar que a dimensão temporal não tem a importância que merece na literatura. Na direcção oposta, há um interesse crescente sobre a dimensão temporal em outras áreas da recolha de informação, nomeadamente modelos de recolha, apresentação de conjuntos de resultados, agrupamento, classificação entre outros.

O desafio para que este trabalho aponta é desenvolver um robô cujo propósito seja lidar com as restrições temporais. A importância desta dimensão é certamente amplificada quando combinada com o tópico ou a geografia, mas agora apenas a iremos estudar isoladamente.

A abordagem aplicada é muito directa. É baseada num algoritmo de segmentação temporal de textos e segue apenas as ligações em segmentos dentro do âmbito temporal imposto pela restrição.

Este sistema está concebido para páginas Web em português, embora a sua filosofia possa ser aplicada a outras línguas.

Além disso, e para melhorar os resultados, o algoritmo utilizado prioriza o descarregamento de páginas com mais ligações dentro do âmbito temporal. A precisão dos resultados ronda os 75%

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

CMS	Content Management System
COP	Co-Ocurrence Processor
FIFA	Fédération Internationale de Football Association
FIFO	First In, First Out
IDF	Inverse Document Frequency
LAN	Local Area Network
MD5	Message-Digest Algorithm 5
MRD	Módulo de Resolução de Datas
SGM	Segunda Guerra Mundial
URL	Uniform Resource Locator
WAN	Wide Area Network
XML	Extensible Markup Language
9/11	Ataques de 11 de Setembro de 2001, nos EUA

Capítulo 1

Introdução

A descarga Web é algo que existe há muitos anos e é amplamente utilizado nos dias de hoje, principalmente por motores de busca como a Google, Yahoo! ou Bing. Estes robôs tentam percorrer uma parte significativa da Web à procura de informação para construir os seus índices. Esses índices são depois apresentados aos utilizadores sob a forma de URLs sempre que eles colocarem uma interrogação no motor de busca.

Embora os motores de busca tentem percorrer a Web completa, isso não é viável nem possível. A Web cresce mais a cada dia embora a sua indexação não siga o mesmo ritmo.

Por isso mesmo, são necessários mecanismos de descarga mais eficientes. Já que não é possível indexar toda a informação da Web, a ideia a seguir é tentar indexar a parte mais importante relativamente às necessidades dos utilizadores. É esta abordagem que os motores de busca actuais seguem: tentar dar ao utilizador a melhor resposta possível sem ter toda a informação disponível indexada.

O trabalho de percorrer a Web é um trabalho muito difícil, daí que a utilização de robôs com restrições seja uma das melhores formas de ultrapassar o problema de escalabilidade inerente à Web.

A descarga Web com restrições é bastante eficaz e aumenta bastante a sua eficiência quando o processo de descarga é descentralizado [6].

Os motores de busca actuais não conseguem relacionar temporalmente os documentos pois não têm mecanismos de descarga com restrições temporais ou porque não analisam as páginas temporalmente após a descarga. Não existe forma de relacionar informação com base no tempo.

Por exemplo, colocando o exemplo descrito em [7] em prática no motor de busca Google, ao fazer a pesquisa Einstein 1900 a 1910, iremos querer resultados que tenham os termos da pesquisa mas também resultados sobre Einstein ter terminado a sua tese em 1905.

Apenas se o sistema souber que 1905 está no intervalo das datas colocadas na interrogação é que ele pode devolver 1905 como resposta correcta.

Para que os motores de busca tenham a capacidade de relacionar os documentos no tempo, eles precisam de robôs que façam a descarga das páginas utilizando restrições temporais ou então que as analisem temporalmente após a descarga.

Um robô tem outros desafios a ultrapassar a fim de lidar com a Web. Que ligações deve seguir, quando deve revisitar as páginas ou a qual a arquitectura do robô são todos aspectos importantes que devem ser resolvidos para que o robô possa fazer bem o seu trabalho.

Neste trabalho apenas irá ser focado o processo de descarga de páginas Web com restrições temporais.

O processo de descarga com restrições temporais serve para que as páginas descarregadas pelo robô digam respeito a um determinado intervalo de tempo ou época. Assim, a informação contida nas páginas descarregadas, pode ser relacionada temporalmente.

1.1 Objectivos

O grande objectivo deste projecto é construir um mecanismo que seja capaz de restringir temporalmente as páginas a ser descarregadas por um robô. Como restrição temporal, entende-se que o conteúdo da página deve estar condicionado a um determinado período de tempo, por exemplo 1939 até 1945.

A ideia é pegar num robô de uso geral e modificá-lo para que ele consiga limitar temporalmente as páginas a descarregar.

Ao pegar num robô de uso geral, alguns componentes terão de ser modificados e acrescentados para que possam suportar a dimensão temporal.

Para limitar temporalmente a descarga, é necessária uma ferramenta capaz de identificar e extrair marcas temporais dos documentos para serem posteriormente normalizadas e interpretadas pelo robô.

A ferramenta usada para extrair a informação temporal basear-se-á numa adaptação de uma plataforma usada para a segmentação temporal de documentos de texto [3, 4, 8]. A informação temporal que esta ferramenta irá encontrar, será depois analisada para perceber se esse documento está ou não dentro das restrições temporais especificadas.

Que seja do conhecimento do autor, não existe nenhum robô cujo foco seja estabelecer restrições temporais aos conteúdos dos documentos descarregados. Por este facto, foi submetido para publicação em conferência internacional um primeiro artigo [9] em que é apresentada uma parte do trabalho desta dissertação.

Com base na mesma ferramenta, foi analisada temporalmente uma colecção da Web Portuguesa e produzido um conjunto de estatísticas consideradas de interesse. Esta análise está descrita no capítulo 6 e será também objecto de publicação [10].

1.2 Estrutura do Documento

No Capítulo 1, é feita uma pequena introdução a este trabalho. Neste capítulo são também apresentados os objectivos que este trabalho quer atingir. É também descrita a estrutura deste documento.

No Capítulo 2, é feita uma análise do estado da arte relativa aos robôs. Eles são os responsáveis por percorrer a Web para descarga de informação que será posteriormente utilizada. Aqui apresentam-se as formas de percorrer a Web, os diferentes métodos de escolha dos URLs a seguir, a actualização das páginas e

maneiras de paralelizar o processo de descarga. É também apresentada a arquitectura para um robô incremental.

No Capítulo 3, é interrogada a problemática da descarga Web com restrições. A utilização das restrições é uma maneira de lidar com o problema de escalabilidade inerente à Web. A utilização de restrições ao nível do tópico ou ao nível da geografia permitem também tornar o processo de descarga mais eficiente.

No Capítulo 4, discutem-se aspectos relacionados com a informação temporal existente na Web. Saber como se organiza essa informação é importante para perceber qual a melhor forma de lidar com o problema das descargas temporais.

O problema da escalabilidade leva a que apenas uma pequena parte da Web seja descarregada e indexada. A informação recolhida deve, por isso, ser aquela que for mais relevante aos utilizadores que pode ser inferida com base nas interrogações feitas por estes. É também feita neste capítulo a comparação entre a arquitectura de um robô geral e um robô com restrições. Para complementar, são comparados os vários tipos de restrições.

No Capítulo 5, apresenta-se a ferramenta que segmenta temporalmente os documentos. Esta ferramenta analisa o texto para descobrir expressões temporais. Esta análise é feita através de um conjunto de módulos que interpretam valores textuais e os convertem em datas. Para além de descobrir as expressões temporais, esta ferramenta divide o documento em várias partes, coerentes do ponto de vista temporal, chamadas de segmentos. Cada segmento corresponde a um momento determinado no tempo.

No Capítulo 6, é feita a análise temporal de uma colecção de páginas da Web Portuguesa. Pretende-se aqui entender de que forma as expressões temporais aparecem nos documentos Web em Português.

No Capítulo 7, é apresentada a arquitectura do robô de descargas temporais. A arquitectura proposta permite ao robô ter efectivamente capacidades de análise temporal das páginas descarregadas, fazendo com que o seu percurso seja guiado pela dimensão tempo.

No Capítulo 8, são discutidos os resultados experimentais resultantes da implementação da arquitectura proposta. Utilizam-se duas colecções obtidas da

Wikipédia em Português, usando como ponto de partida a Segunda Guerra Mundia (1939 - 1945) e os ataques de 11 de Setembro de 2001. São também apresentadas as métricas de avaliação escolhidas.

Quer as conclusões quer o trabalho futuro estão incluídos no Capítulo 9, que termina esta dissertação.

Capítulo 2

Descarga Web

O desenho de um robô apresenta muitos desafios, em particular o ter de lidar com uma quantidade imensa de informação. Como não tem recursos ilimitados nem tempo ilimitado, tem de escolher quais os URLs que deve descarregar e em que ordem o deve fazer.

2.1 Arquitectura Geral de um Robô

O funcionamento de um robô é relativamente simples quando comparado com a complexidade da tarefa que tem em mãos.

Na Figura 2.1 está ilustrado o funcionamento normal de um robô sequencial [1]. O robô mantém uma lista de URLs não visitados a que se chama Fronteira. A lista é iniciada com uma semente de URLs que pode ser fornecida por outro programa. Cada ciclo do robô envolve pegar no URL seguinte da lista, descarregar a página da Web, fazer a análise da página descarregada para extrair novos URLs e informação específica estabelecida anteriormente.

Finalmente os novos URLs não visitados são adicionados à lista de URLs da fronteira. Para uma melhor utilização dos recursos do processo de descarga ou para melhorar a informação a recolher, os novos URLs podem passar por um avaliador que atribui um valor àquele URL.

Esse valor representa o benefício estimado de passar pela página correspondente àquele URL.

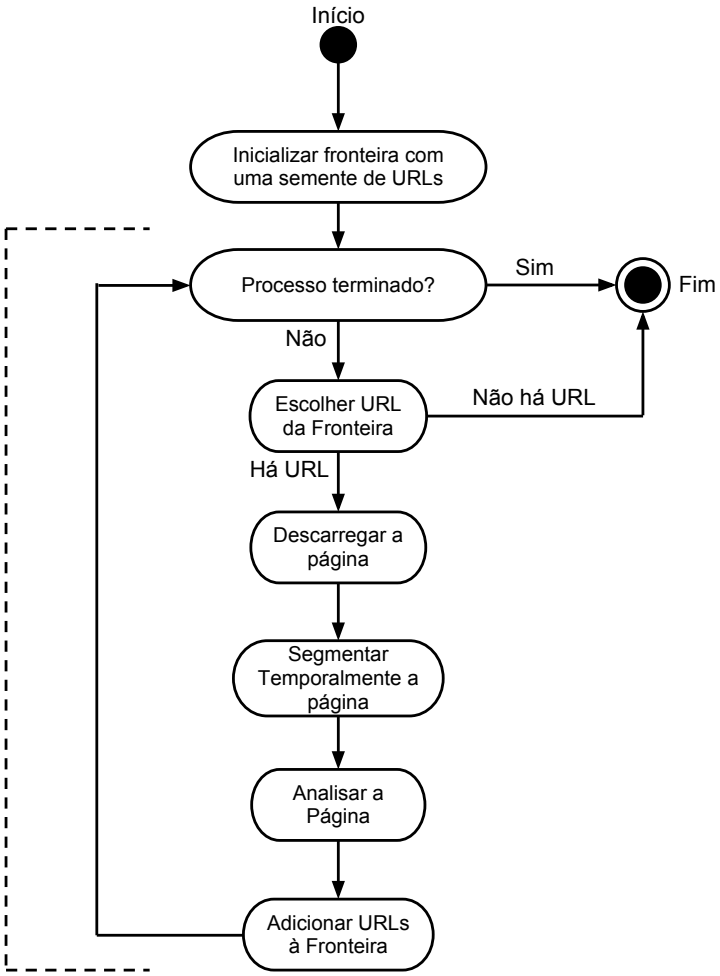


Figura 2.1: Fluxo de um robô sequencial geral (adaptado de [1])

O processo de descarga dá-se por terminado quando um pré-determinado número de páginas forem analisadas ou quando a fronteira estiver vazia. Caso o último ocorra, o processo de descarga termina de imediato pois já não existem mais URLs para visitar.

O processo de descarga pode ser visto como um problema de pesquisa de

grafos [1]. A Web é vista como um grafo gigante com páginas nos seus nós e hiperligações nas suas folhas. Um robô começa com uma quantidade pequena de nós (sementes) e depois segue pelas ligações para chegar a outros nós.

O processo de descarregar uma página e extrair as suas ligações é análogo a expandir um nó numa pesquisa num grafo.

Um robô com restrições, tenta seguir ligações às quais se espera levar a porções do grafo que respeitam a restrição imposta.

2.1.1 Fronteira de URLs

A fronteira é a lista de afazeres de um robô que contém os URLs de páginas por visitar. Em linguagem de grafos, a fronteira é uma lista aberta de nós não expandidos (não visitados).

A fronteira pode ser implementada numa fila de espera por ordem de chegada (FIFO - First In, First Out). Os URLs a procurar vêm da cabeça da fila de espera enquanto os novos URLs são adicionados à cauda da fila.

Alternativamente a fronteira pode ser implementada numa estrutura diferente da de uma fila de espera, como por exemplo, uma fila de prioridades.

Este tipo de estrutura pertence a um robô preferencial. A fila prioritária pode ser um vector dinâmico que está constantemente a ser ordenado pela pontuação estimada de cada nó por visitar.

Desta forma atribui-se pontuações aos URLs, podendo assim percorrer a Web, não de uma forma cega como numa fila FIFO, mas de uma forma mais controlada, tornando assim o processo de descarga mais eficiente.

2.1.2 Histórico e Repositório de Páginas

O histórico do processo de descarga é uma lista de URLs com uma marca temporal que foram descarregados pelo robô. Na prática, mostra o percurso do robô pela Web começando pelos URLs que serviram de semente. Um URL só entra para

o histórico após a página Web correspondente ter sido descarregada e analisada. Este histórico pode ser utilizado mais tarde para análise e avaliação pós-descarga.

Por exemplo, associando um valor com cada página do caminho percorrido pelo robô, podemos com isso encontrar eventos significativos, como a descoberta de um recurso de informação excelente.

Para evitar passar novamente por URLs já visitados, este histórico também é mantido numa estrutura em memória para uma rápida verificação dos URLs, evitando assim percorrer nós já visitados e adicionar "novos"URLs à fronteira. A estrutura mantida em memória pode conter, por exemplo, o MD5[1] dos URLs já visitados.

2.1.3 Descarga das Páginas

Para descarregar uma página Web, é necessário ter um cliente HTTP para enviar pedidos GET às páginas e ler as suas respostas.

O cliente precisa de ter tempos-limite implementados para que não seja gasto tempo desnecessário em servidores muito lentos ou a ler páginas muito grandes. Na verdade, podemos restringir o cliente para descarregar apenas os primeiros 10-20KB de cada página. O cliente precisa de analisar os códigos de estado e as redirecções dos cabeçalhos das respostas.

Também é boa prática analisar e guardar o último cabeçalho modificado para determinar a idade do documento. Verificação de erros e tratamento de excepções durante o processo de descarga é importante pois o mesmo pedaço de código vai interagir com milhões de servidores remotos.

Algo que também ajudará a melhorar o processo de descarga é recolher estatísticas dos tempos-limite e dos códigos de estado para identificar problemas ou para, automaticamente, alterar os valores desses tempos.

[1] Message-Digest Algorithm 5

2.1.4 Políticas de Delicadeza

Esta discussão não estaria completa sem falar sobre o *Robot Exclusion Protocol* [2]. Este protocolo fornece mecanismos para os administradores Web comunicarem as suas políticas de acesso, mais especificamente para identificar ficheiros que não podem ser acedidos por um robô. Isto é feito com um ficheiro chamado robots.txt colocado na raiz do servidor Web (como por exemplo: http://www.google.com/robots.txt). Este ficheiro fornece políticas de acesso diferentes para diferentes *User-agents* (robôs).

Um *User-agent* com o valor de '*' indica uma política por defeito para qualquer robô que não encontre uma correspondência num outro *User-agent* definido no ficheiro. Um conjunto de entradas *Disallow* pode ser fornecida para cada *User-agent*. Qualquer URL que comece com o valor de um campo *Disallow* não pode ser recolhido por um robô que corresponde ao *User-agent*.

Quando um robô quer ir buscar uma página de um servidor Web primeiro tem de ir buscar o ficheiro *robots.txt* apropriado e ter a certeza que o URL que quer não está "bloqueado". É mais eficiente armazenar em cache as políticas de acesso de um número de servidores recentemente visitados pelo robô. Isto evita aceder ao ficheiro *robots.txt* de cada vez que é preciso fazer descarga de um URL. No entanto é preciso garantir que as entradas na cache estão actualizadas o suficiente.

O intervalo de tempo entre descargas também deve ser salvaguardado. A utilização de um intervalo muito baixo leva a que o servidor Web seja inundado com uma quantidade de pedidos anormal, causando lentidão e demora na resposta do servidor a pedidos de outros utilizadores.

Por isso, deve ser utilizado um intervalo que não prejudique a normal funcionamento do servidor Web.

Este intervalo de tempo é facilmente controlado quando apenas existe uma tarefa[3]. Agora quando existem mais poderá não ser assim tão fácil. Com várias tarefas, corre-se o risco de existirem múltiplos pedidos simultâneos ao mesmo

[2]Mais informações sobre este protocolo podem ser encontradas em http://www.robotstxt.org/wc/norobots.html

[3]Thread na terminologia em inglês

servidor.

Uma forma de solucionar este problema é ter cada tarefa responsável por apenas um domínio. Isto permite controlar de forma mais fácil o intervalo entre descargas.

2.1.5 Análise das Páginas

A análise é aquilo que se faz após a página ter sido descarregada. Para extrair a informação dela é preciso analisá-la pois essa informação poderá servir para guiar o caminho futuro do robô. Fazer esta análise pode ser apenas uma questão de extrair ligações ou envolver uma análise mais complexa como, por exemplo, retirar expressões temporais do documento por forma a estabelecer ligações temporais ao mesmo.

Fazer a análise ao documento também pode envolver passos para converter os URLs extraídos para a forma canónica, remover as palavras negativas do conteúdo da página e fazer a lematização às restantes. Isto é feito para que todas as páginas recolhidas sejam analisadas de igual forma.

- **Extracção e Canonização dos URLs**

  Analisadores para HTML estão disponíveis gratuitamente para diferentes linguagens. Estes fornecem a funcionalidade para facilmente identificar as marcas HTML e associar pares atributo-valor para um dado documento HTML. De modo a extrair os URLs das hiperligações, nós podemos utilizar esses analisadores para encontrar as marcas âncora e recolher o valor do atributo href.

  Diferentes URLs que correspondem à mesma página podem ser encontrados e mapeados para uma única forma canónica. Isto é importante para evitar que a mesma página seja descarregada múltiplas vezes. Eis alguns passos típicos nos procedimentos de canonização de URLs:

  - converter o protocolo e o hostname para minúsculas.

Exemplo: *HTTP://www.EXAMPLE.com* é convertido para *http://www.-example.com*.

- remover a parte da âncora ou referência do URL.
  Assim, *http://www.example.com/faq.htmlwhat* fica reduzido a *http://www.example.com/faq.html*.
- fazer a codificação do URL para os caracteres mais comuns como o til (~). Isto previne o robô de tratar *http://www.example.com/pant/* diferente de *http://www.example.com/%7Epant/*.
- para alguns URLs, adicionar no fim a barra '/'.
  http://www.example.com e *http://www.example.com/* têm de ser mapeados para a mesma forma canónica. A decisão de adicionar a barra '/', em muitos casos, vai requerer a utilização de heurísticas.
- utilizar heurísticas para reconhecer as páginas Web por defeito. Nomes de ficheiros como *index.html* ou *index.htm* podem ser removidos do URL pois é assumido que eles são os ficheiros por defeito. Se isso é verdade, eles serão devolvidos utilizando somente o URL base.
- remover '..' e a sua directoria pai do URL. Assim sendo, o URL /Epant/BizIntel/Seeds/../ODPSeeds.dat fica reduzido a /Epant/BizIntel-/ODPSeeds.dat.
- deixar o número dos portos no URL a não ser que seja o porto 80. Como alternativa, deixar os números dos portos no URL e adicionar o porto 80 quando nenhum porto é especificado.

É importante ser consistente quando se aplicam as regras de canonização. É possível que duas regras aparentemente opostas funcionem igualmente bem (como as regras para o número dos portos) desde que elas sejam aplicadas de forma consistente a todos os URLs.

2.2 Escolha dos URLs

Como é que um robô escolhe quais os URLs que deve ou não descarregar da sua lista de URLs? Esta é a pergunta que se tenta responder neste capítulo. Se a intenção do robô é descarregar toda a Web, então qualquer um serve. No entanto, os robôs não conseguem fazer isso devido a duas razões principais: a quantidade limitadas de armazenamento e a revisita das páginas.

O robô, ou o seu cliente, tem quantidades limitadas de armazenamento não sendo possível indexar e analisar todas as páginas. Crê-se que a Web tenha vários milhares de Terabytes de informação, assim não se espera que o cliente queira ou consiga lidar com toda essa informação [11].

O processo de descarga demora e, numa dada altura, ele vai ter de revisitar as páginas previamente descarregadas para registar as mudanças que ocorreram nelas. Isto faz com que muitas páginas nunca sejam visitadas.

Mas em todo o caso, o importante é que o robô visite as páginas "importantes"antes de tudo, para que a fracção da Web que é visitada (e é mantida actualizada) seja a mais significativa.

Nem todas as páginas são importantes, por isso é necessário estabelecer prioridades de acordo com o objectivo previamente estabelecido. Por exemplo, se o cliente do robô está a construir uma base de dados especializada num determinado tópico, então as páginas que se referem a esse tópico são mais importantes e devem ser visitadas o mais cedo possível.

Em semelhança, um motor de busca utiliza o número de URLs que apontam para uma página para classificar os resultados das interrogações dos utilizadores. Se o robô não consegue visitar todas as páginas, então é melhor visitar aquelas com uma contagem de ligações maior, pois essas darão ao utilizador resultados melhor classificados.

Dada uma página Web, podemos definir a importância de uma página p utilizando uma das seguintes medidas.

1. *Semelhança com a interrogação principal*: aqui, uma interrogação principal

é que conduz o processo de descarga e a importância da página é definida através da semelhança textual entre a página e a interrogação.

Para calcular as semelhanças, cada documento é visto como um vector n-dimensional (*w1, ..., wn*) onde *wi* representa a palavra *i* no vocabulário. Se *wi* não existir no documento, então *wi* é zero. Se aparecer, *wi* é definido para representar a significância da palavra.

Uma forma comum de calcular a importância de *wi* é multiplicar o número de vezes que a palavra número *i* aparece no documento pela frequência inversa da palavra *i* no documento (*idf: inverse document frequency*). Este factor é aquele que é dividido pelo número de vezes que a palavra aparece na "colecção"inteira, que neste caso seria a Web. Uma palavra que apareça raramente em documentos tem um *idf* alto, enquanto uma palavra que apareça muitas vezes em documentos tem um *idf* baixo.

A semelhança entre *p* e a interrogação pode ser definida como o produto interno não normalizado entre os vectores da página e da interrogação. Outra opção é utilizar o cosseno, que é o produto interno entre vectores normalizados. [11]

2. *Contagem de ligações remotas*: o valor da importância de *p* é o número de ligações para *p* que aparecem em toda a Web. Intuitivamente, uma página que é referenciada por muitas outras páginas, é mais importante do que uma raramente o é. Na Web este tipo de classificação é útil para classificar resultados de interrogações, dando ao utilizador final páginas de mais provável interesse geral.

   De notar que, para fazer esta avaliação, é necessário contar as ligações de toda a Web. Um robô pode estimar este valor com o número de ligações para a página que viu até ao momento. [11]

3. *PageRank*: esta métrica de contagem de ligações define a importância das páginas calculando a soma ponderada das páginas que têm ligações para *p*. A melhor forma para perceber este algoritmo é pensando num utilizador a "surfar"a Web que começa o seu caminho numa página qualquer e que escolhe uma ligação dessa página para seguir de forma aleatória. Quando

o utilizador chega a uma página sem ligações, ele salta para outra página aleatoriamente. Enquanto ele faz este percurso pela Web, saltando de ligação em ligação, ele irá visitar umas páginas mais vezes do que outras. Intuitivamente, estas serão aquelas que têm muitas ligações vindas de outras páginas.

A ideia por trás do PageRank é que as páginas que foram mais vezes visitadas nesse percurso aleatório são mais importantes do que as outras. [11]

4. *Contagem de ligações próprias*: é considerado o número de ligações iniciadas por p. Com esta métrica, uma página que tenha muitas ligações para o exterior é importante pois poderá ser um directório Web. [11]

5. *Métrica de Localização*: a importância de *p* é calculada em função da sua localização e não do seu conteúdo. Se um URL *u* leva a *p*, então a métrica de localização de *p* é uma função de *u*. Por exemplo, URLs terminados em .com podem ser considerados mais importantes do que URLs com outra terminação ou, então, URLs que tenham a palavra "home"podem ter mais interesse do que outros. Outra métrica de localização que é por vezes utilizada considera os URLs com menos barras mais úteis do que aqueles com muitas barras. [11]

Estas métricas podem ser combinadas de várias formas. Por exemplo, ao combinar a métrica de similaridade com a métrica de ligações remotas, significa que as páginas que têm conteúdo relevante e são mais vezes referenciadas sejam melhor classificadas. [11]

2.3 Actualização das Páginas

Após um robô ter seleccionado e descarregado as páginas "importantes", ele tem de, periodicamente, actualizar essas páginas para que elas se mantenham frescas.

Por exemplo, páginas de notícias como *CNN* ou *NY Times* mudam as suas páginas sempre que há novos desenvolvimentos. Outro exemplo são as lojas de

compra online que actualizam o preço e a disponibilidade dos seus produtos dependendo do inventário e das condições do mercado.

Nestes casos, o robô não sabe exactamente quando e com que frequência as páginas mudam.

2.3.1 Tipo de Alterações

Mas antes de se entrar no tópico sobre como estimar frequência de alteração de um elemento é preciso clarificar o se quer dizer com "alteração de um elemento", o que se quer dizer com "elemento"e o que é que significa haver "mudança".

Então, um elemento é uma página Web e a mudança é qualquer modificação a essa página.

Um elemento pode ser definido como uma página Web, uma parte dessa página, um conjunto de items dessa página, etc.

A mudança pode ser definida como, por exemplo, uma modificação a mais de 30% da página ou a alteração de uma única letra.

Estas mudanças podem ocorrer ao nível de conteúdo, de estrutura, de apresentação e de comportamento.

As mudanças no conteúdo são as mudanças na informação textual. As mudanças de estrutura estão relacionadas com o modelo hierárquico da página Web. As mudanças na apresentação são as mudanças que ocorrem na forma como se apresenta a informação visualmente. As mudanças no comportamento são as mudanças que ocorreram nos componentes activos do html [12].

2.3.2 Frequência da Actualização

Manter a informação indexada constantemente actualizada não é uma tarefa fácil visto que a Web é dinâmica por natureza [13].

Isto significa, por exemplo, que o que está hoje numa determinada página deixa de estar amanhã. Como tal, a página necessita de ser novamente descar-

regada e indexada para que a informação guardada se mantenha fresca. Mas é difícil saber que informação se alterou e capturar as mudanças significativas.

Assim, a estratégia do robô tem de se adaptar à frequência de alterações das páginas Web [12].

A frequência de actualização de uma página vai variar de acordo com o tipo de domínio. Um domínio *.com* demora 11 dias a sofrer alterações enquanto que o domínio *.gov* demora 4 meses a sofrer a mesma quantidade de alterações [13, 14].

Adequar a estratégia de descarga para cada tipo de domínio é importante para optimizar o processo. Assim, em vez de se andar a perder tempo em páginas que não têm alterações, as atenções estão focadas naquelas que de facto foram alteradas e necessitam de ser novamente indexadas.

Esta tarefa ganha ainda mais importância visto que a Web demora apenas 50 dias para que 50% dela se altere [13, 14].

Existem 3 abordagens para determinar a frequência com que uma página é actualizada: a abordagem estática, dinâmica e estatística.

Na abordagem estática são retiradas marcas temporais do conteúdo da página ou do cabeçalho *http* e toda a página é actualizada, não existindo nenhum critério de frequência de actualização.

Na abordagem dinâmica existe um processo de comparação que faz com que as alterações venham ao de cima por si. Para isto é necessário haver versões para comparar, definir um modelo para a página Web e definir métricas de semelhança entre os elementos do modelo.

Por fim existe a abordagem estatística em que é feita uma estimativa sobre quando ocorrerá uma nova mudança. Após terem sido observadas várias mudanças para uma página, são extraídos delas modelos preditivos para a próxima data de alteração [12].

Cho e Garcia Molina em [2] definem duas métricas do estado de actualização de uma colecção: a frescura e a idade.

A frescura mede a percentagem de páginas actualizadas e a idade mede o tempo que as páginas se encontram desactualizadas.

A maior parte dos robôs não tem nenhuma política de revisita de páginas, mas, considerando que um conjunto de páginas são descarregadas uma vez, é sempre possível voltar a descarrega-las e assim actualizá-las.

Uma alternativa é a descarga incremental em que a descoberta de novas páginas e a actualização das mesmas é feita alternadamente. Este método tem a vantagem de não ter débitos de descarga tão elevados quanto o primeiro [2].

No entanto, com outro trabalho de Cho e Molina, provou-se que visitar todas as páginas com a mesma frequência leva a que a colecção tenha níveis de frescura médios mais elevados em toda a colecção [13].

2.4 Paralelização do Processo de Descarga

A paralelização do processo de descarga pode acontecer de duas maneiras: 1) os vários processos de descarga ocorrem na mesma rede física e comunicam através de uma ligação de alta velocidade (como uma rede local) ou 2) os processos de descarga ocorrem em zonas geográficas distintas e estão ligados através da Internet ou de uma rede de longa distância [15].

A distribuição geográfica do processo leva a que haja uma dispersão do acesso à Internet, tirando partido de vários fornecedores de serviço em simultâneo. Isto faz com que haja uma dispersão e redução da carga pela rede. Tudo isto produz débitos agregados maiores diminuindo o tráfego global gerado. A carga computacional e de armazenamento também são dispersas pelas máquinas [15].

Este processo é vantajoso mas tem de existir uma coordenação entre os vários processos de descarga. Como o processo é distribuído, há o risco de os vários processos percorrerem as mesmas páginas. Isto leva a que todo o processo perca eficiência, pois a mesma página poderá ser descarregada várias vezes. Por isto, é necessário haver algum tipo de coordenação.

Esta coordenação pode ser feita segundo: 1) uma atribuição dinâmica, controlada por um coordenador central que diz qual o caminho que o processo de descarga deve seguir; 2) uma atribuição estática que dispensa a utilização de um

coordenador uma vez que cada processo conhece em antemão qual o caminho a seguir.

Devido à escalabilidade dos sistemas de descarga com restrições e à possibilidade da descentralização do processo de descarga, algo que vem sem surpresa é a partição da Web por zonas geográficas para a distribuição do processo de descarga [6].

A descarga por âmbito geográfico implica que um processo de descarga seja responsável por uma zona física. Por exemplo, se um processo de descarga está nos Estados Unidos não faz muito sentido que seja ele o responsável por descarregar uma página da Europa. Faz mais sentido que esse processo seja distribuído por zonas, tendo cada processo uma zona a seu cargo.

Essas zonas podem ser as zonas .pt ou então zonas que se encontram nas redondezas do processo de descarga e ele fica responsável pelas páginas nas suas proximidades.

Esta paralelização faz com que o processo de descarga seja feito, no seu todo, de forma mais eficiente e obtenha resultados combinados muito melhores do que se estivesse a trabalhar sozinho.

2.5 Arquitectura de um Robô Incremental

O robô percorre a Web continuamente, revisitando as páginas periodicamente. Durante o processo contínuo de descarga, poderá também retirar algumas páginas da colecção local para dar espaço a novas páginas. Este processo é feito para que a colecção local esteja sempre actualizada e para melhorar a qualidade da colecção local. Melhorar a qualidade da colecção local significa substituir as páginas "menos importantes"com páginas "mais importantes".

Este processo é necessário por duas razões. Primeiro porque as páginas são constantemente criadas e removidas. Algumas das novas páginas podem ser "mais importantes"que as existentes na colecção e assim o robô deve substituí-las pelas "menos importantes". Segundo, a importância das páginas varia com o tempo.

Quando algumas das páginas existentes tornam-se menos importantes que as páginas previamente ignoradas, o robô deve substituir as páginas existentes por aquelas previamente ignoradas.

2.5.1 Modelo Operacional de um Robô Incremental

Na Figura 2.2 está ilustrado o modelo de funcionamento de um robô incremental proposto em [2]. Aqui, a fronteira guarda todos os URL já descobertos e a colecção local guarda todos os URLs na colecção. Para simplificar a discussão, vamos assumir que a colecção local mantém um número fixo de páginas e que a colecção local encontra-se na sua capacidade máxima desde o início.

Nos passos 1 e 2, o robô escolhe o URL e obtém a página. Se a página já existir (condição no passo 3) o robô actualiza a imagem da página que tem na colecção local (passo 4). Se não, o robô descarta uma página existente da colecção (passos 5 e 6), guarda a nova página (passo 7) e actualiza a colecção local (passo 8). Finalmente, o robô extrai os URLs existentes na nova página para os adicionar à fronteira (passos 9 e 10).

O robô toma decisões nos passos 1 e 5. No passo 1, o robô decide qual a página a descarregar e no passo 5 qual a página a descartar. No entanto, as decisões nestes dois passos estão interligadas, isto é, quando o robô decide descarregar uma nova página, ele tem de descartar uma página da colecção local para haver espaço para a nova página. A este processo chama-se decisão de refinamento.

Esta decisão de refinamento não deve ser baseada na "importância"das páginas. Para medir a importância, o robô pode utilizar várias métricas de importância listadas no subcapítulo 2.2. Claramente a importância da página descartada deve ser menor do que a da nova página. Na verdade, a página descartada deverá ter o nível de importância mais baixo de toda a colecção, para manter o nível de qualidade da colecção ao seu máximo.

Juntamente com a decisão de refinamento, o robô decide que página deve actualizar no passo 1. Isto é, em vez de visitar uma nova página, o robô decide visitar uma página existente para actualizar a imagem dela. Para manter a colecção

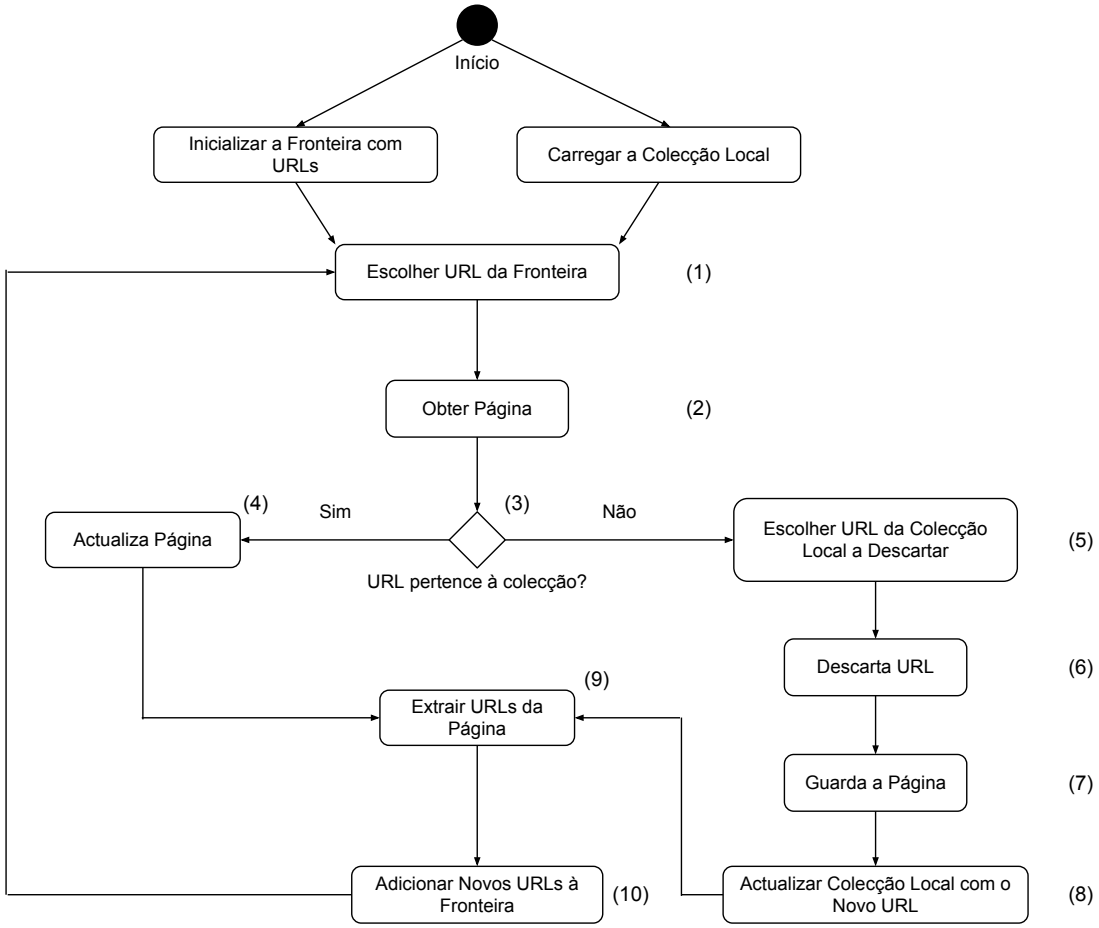


Figura 2.2: Modelo Conceptual da Operação de um robô incremental (adaptado de [2])

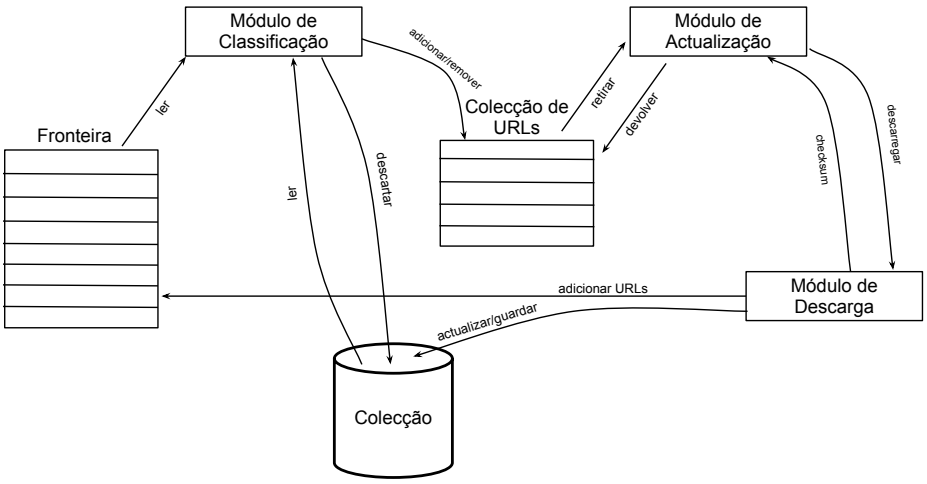


Figura 2.3: Arquitectura de um robô incremental (adaptado de [2])

"fresca", o robô tem de decidir qual a página que mais aumentará o seu nível de "frescura". Esta chama-se uma decisão de actualização.

2.5.2 Descrição da Arquitectura

Segundo Cho em [2], a arquitectura de um robô incremental consiste em três módulos principais (módulo de classificação, módulo de actualização e módulo de descarga) e três estruturas de dados (fronteira, colecção de URLs e colecção) tal como está ilustrado na Figura 2.3. As linhas e setas representam o fluxo de dados entre os módulos e as etiquetas nas linhas mostram os comandos correspondentes.

Duas estruturas de dados, fronteira e colecção de URLs, mantêm a informação sobre as páginas. A fronteira guarda todos os URLs descobertos pelo robô e a colecção de URLs guarda todos os URLs que estão/estarão na colecção. A colecção de URLs é organizada como uma fila prioritária em que os URLs para serem lidos primeiro são colocados no início da lista.

Os URLs na colecção de URLs são escolhidos pelo módulo de classificação. Este módulo lê constantemente a fronteira e a colecção de URLs para tomar a

decisão de refinamento. Neste módulo é que são introduzidas as métricas descritas em 2.2.

Por exemplo, se a métrica de importância escolhida é o *PageRank* então o módulo de classificação reavalia constantemente o *PageRank* de todos os URLs baseando-se na estrutura de ligações na colecção. De notar que, mesmo que uma página p não exista na colecção, o módulo de classificação consegue estimar o *PageRank* de p baseando-se na quantidade de páginas na colecção que apontam para p.

Enquanto o módulo de classificação refina a colecção, o módulo de actualização mantém a colecção "fresca"(decisão de actualização). Este extrai constantemente a primeira entrada na colecção de URLs, faz um pedido ao módulo de descarga para descarregar a página e volta a colocar o URL já descarregado na colecção de URLs. A posição do URL descarregado na colecção de URLs é determinada pela frequência estimada de actualização da página e pela sua importância, ou seja, quanto mais perto o URL está no início da fila, mais frequentemente ele será revisitado.

Para estimar a frequência com que uma página muda, o módulo de actualização guarda o *checksum* da página da última vez que a descarregou e depois compara esse *checksum* com aquele da descarga actual. A partir desta comparação, o robô percebe se a página foi alterada ou não. Neste módulo, Cho propõe a introdução de dois tipos de "estimadores"E_p e E_b para a frequência de alteração de uma página.

E_p é baseado no modelo de *Poisson* e o E_b baseia-se num método *Bayesianno*. Estes podem ser vistos em mais pormenor em [13].

O módulo de descarga descarrega uma página e guarda/actualiza a página na colecção baseando-se no pedido feito pelo módulo de actualização. O módulo de descarga também extrai todos os URLs existentes na página e envia-os para a fronteira. Esses URLs enviados são incluídos na fronteira caso sejam novos.

Separar a decisão de actualização (módulo de actualização) da decisão de refinamento (módulo de classificação) é crucial por razões de desempenho. Por exemplo, para visitar 100 milhões de páginas todos os meses, o robô tem de visitar

40 páginas por segundo. No entanto existe demora a escolher e a remover páginas da colecção de URLs pois calcular a importância das páginas é muito pesado.

Por exemplo, para um robô calcular o *PageRank*, necessita de percorrer toda a colecção de URLs múltiplas vezes, mesmo que a estrutura de ligações tenha alterado pouco. Claramente o robô não consegue recalcular a importância das páginas para cada página descarregada quando precisa de correr a 40 páginas por segundo.

Ao separar a decisão de actualização da decisão de refinamento, o módulo de actualização pode-se preocupar em actualizar as páginas a alta velocidade, enquanto o módulo de classificação refina a colecção com o devido cuidado.

2.6 Desafios para um Robô

Existem alguns problemas que fazem com que a tarefa de percorrer a Web seja muito difícil. O maior deles todos é o tamanho da Web. A cada dia que passa a Web aumenta em tamanho. Ao aumentar, aumenta também a pressão para que os robôs indexem essa nova informação.

A Google, em 2008, disse que atingiu uma nova marca na quantidade de URLs únicos na Web: 1 bilião (1,000,000,000,000) [16]. Actualmente estima-se que, apenas a Google, tenha à volta de 50 mil milhões de páginas indexadas [4].

Em 2000, o tamanho da Web estimava-se em 10 mil milhões de páginas estáticas e, dessas, apenas 2 mil milhões estavam indexadas [14]. Comparando com os números de agora, pode-se verificar que a Web aumentou mas a indexação não seguiu o mesmo ritmo.

Ao problema do tamanho da Web junta-se o problema da "frescura"dos índices. Como já foi descrito no capítulo 2, as páginas terão de ser revisitadas periodicamente para que elas sejam actualizadas. Elas são actualizadas para que a informação indexada seja a mais actual possível.

Tomemos o exemplo de um motor de busca. Este faz o melhor que pode

[4]Dados disponíveis em http://www.worldwidewebsize.com/

para responder às interrogações feitas pelos utilizadores. Mais, se pensarmos que os utilizadores apenas irão olhar para os primeiros resultados, reparamos que a precisão do motor de busca tem de ser muito elevada para conseguir chegar aos padrões elevados dos utilizadores. Aquilo que irá fazer a diferenciação entre um bom motor de busca e um mau são os resultados que ele apresenta.

Isso leva-nos ao outro problema: o tempo. Efectuar a descarga de páginas Web é algo que demora tempo. Ter que descarregar páginas, analisá-las e indexá-las demora bastante tempo pois tem de repetir todo este processo para milhões delas. Se um robô conseguir operar a uma cadência de 40 páginas por segundo, ele, num mês, apenas vai ter descarregado 100 milhões de páginas.

Este é um número pequeno quando comparado com o tamanho da Web e ainda mais pequeno fica se se considerar que parte dessa descarga é de revisita de páginas para actualização dos índices.

Este é, portanto, um trabalho que não tem fim. Percorrer a Web completa à procura de informação não é viável e novas formas de descarga teriam que ser encontradas.

Capítulo 3

Descarga Web com Restrições

Devido aos problemas expostos anteriormente, não é viável que um robô percorra toda a Web à procura de informação. Não é necessário percorrer toda a Web para conseguir bons resultados. É preciso sim, melhorar a forma como é feita a procura dos resultados para que sejam encontrados os melhores, sem ter que percorrer toda a Web.

Assim, percorrer a parte da Web que é mais relevante deve ser a solução a usar, visto que não é fácil percorrê-la toda.

A descarga com restrições é uma das formas para ultrapassar os problemas da escalabilidade da Web durante o processo de descarga. Este tipo de descarga responde às necessidades particulares de informação expressas através de interrogações ou perfis de interesse.

As restrições estabelecidas são utilizadas para restringir a descarga das páginas. As restrições podem ser encontradas nos conteúdos das páginas mas também através das ligações.

Através das ligações é calculada uma probabilidade de uma ligação ter relação com a restrição em causa fazendo com que a descarga seja direccionada através de ligações a páginas Web com informação relevante de acordo com a restrição.

Essa restrição pode ser qualquer coisa como uma palavra, uma expressão, uma localização geográfica, uma data, um intervalo temporal ou outro tipo de restrição

capaz de ser utilizada para agrupar informação.

A Tabela 3.1 mostra a comparação entre algumas características de robôs gerais e robôs com restrições.

Pela comparação, podemos comprovar que a abordagem com restrições é a melhor para percorrer a Web pois é aquela que retorna resultados mais relevantes. A capacidade de ir estreitando o raio de procura com caminhos específicos faz com que o robô com restrições produza resultados de melhor qualidade.

Um URL mal colocado, facilmente desvia um robô geral do caminho correcto pois segue uma abordagem cega e guiada somente pelo algoritmo para percorrer grafos. Com o robô com restrições, isso não acontece pois ele utiliza um classificador para saber se a página cumpre as restrições definidas [5].

O classificador pode ser considerado como o componente mais importante de um robô com restrições [19]. A sua tarefa é fazer juízos sobre a relevância das páginas descarregadas.

Foram descritos no subcapítulo 2.2 alguns classificadores que podem ser utilizados para medir a importância de uma página para um dado contexto.

A utilização dos classificadores vai depender da abordagem escolhida. Existem três possibilidades: 1) a classificação é feita com base na análise do conteúdo da página; 2) a classificação é feita através da análise das ligações ou 3) através da combinação das duas anteriores.

1. *Classificação baseada no conteúdo:* os robôs com este tipo de classificação baseiam-se somente no conteúdo das páginas descarregadas para estimar a sua relevância com o tópico em causa. Estes classificadores utilizam técnicas de classificação de documentos para assim poderem testar a relevância da página descarregada com o tópico que está a ser analisado.

2. *Classificação baseada na análise de ligações:* os robôs que utilizam esta abordagem baseiam-se nas ligações para identificar potenciais ligações que levem a páginas relevantes. O *PageRank* é um classificador deste tipo.

3. *Classificação baseada em conteúdo e ligações:* os robôs que utilizem esta

	Robô Geral	Robô com Restrições
Definição	Percorre a Web descarregando as páginas Web visitadas de forma exaustiva.	Percorre a Internet descarregando apenas as páginas que cumpra uma determinada restrição.
Percurso	Procura aleatória que pode perder o sentido enquanto percorre a Web.	Páginas que cumprem *a priori* ou *a posteriori* as restrições estabelecidas.
Páginas Web	Não necessariamente relacionadas entre si.	Têm de cumprir com o critério estabelecido.
Robustez	Propenso a distorções de URL.	Robusto contra quaisquer distorções pois segue caminhos de URLs relevantes.
Descoberta	Raio alargado de descoberta de informação mas pouca informação relevante.	Raio estreito de descoberta de informação com muita informação relevante.
Flexibilidade	Bastante personalizável.	Menos personalizável devido às suas dependências.
Classificação	Não utiliza qualquer classificador. Apenas depende de algoritmos para percorrer grafos.	Depende de classificadores para prever a relevância das ligações à restrição em causa.
Visão Geral	Gastos menores de recursos mas desempenho inferior.	Maior consumo de recursos mas grande performance na obtenção de uma colecção de páginas Web com alta qualidade.

Tabela 3.1: Comparação entre um robô geral e um robô com restrições (adaptado de [5])

abordagem utilizam classificadores baseados no conteúdo das páginas e também classificadores com base nas ligações. Como já havia sido dito em 2.2, as métricas podem ser combinadas para assim tentar obter melhores resultados.

Os robôs com restrições percorrem a Web de uma forma bastante eficaz, recolhendo informação com bastante qualidade [2, 5, 11, 14, 17]. Os motores de busca utilizam robôs com restrições para percorrem a Web à procura de informação sobre um determinado tópico.

É importante lembrar que a qualidade da informação recolhida depende em muito da qualidade do classificador usado, ou seja, quanto melhor o classificador, melhor é a informação recolhida.

3.1 Tipos de Restrições

A utilização de restrições, como já foi dito, utiliza-se para lidar com o problema de escalabilidade da Web. Um robô com restrições descarrega apenas as páginas que cumprem as restrições em causa. Essas restrições podem ser ao nível de conteúdo, do âmbito geográfico ou do âmbito temporal.

- *Restrições ao nível de conteúdo:* significa que o percurso do robô pela Web é dirigido pela informação contida nas páginas. A informação contida nas páginas Web é que vai ser útil para os utilizadores pois, cada vez que colocam uma interrogação num motor de busca, o utilizador espera que lhe sejam retornados documentos que digam respeito à interrogação.

  Com este método, espera-se que as páginas descarregadas estejam relacionadas com o tópico estabelecido.

- *Restrições ao nível do âmbito geográfico:* este tipo de restrição faz com que apenas sejam descarregadas páginas que estejam de acordo com um âmbito geográfico. Páginas Web de instituições bancárias ou lojas online são de interesse "global", isto é, independentemente do sítio onde estejamos, são

relevantes para os utilizadores da Web. No entanto, existem outras que o são apenas numa comunidade geográfica.

É importante distinguir a geografia do conteúdo da geografia da localização dos servidores que mantêm a informação. Esta última é importante na descentralização da descarga e também na primeira para determinar o âmbito geográfico do cliente.

Isto permite-nos fazer uma pesquisa num motor de busca por "restaurantes italianos"e nos seja devolvida informação sobre restaurantes italianos na minha zona geográfica [18]. Isto é a procura por âmbito geográfico. Para os motores de busca conseguirem fazer isto, é necessário que existam robôs que tenham restrições de âmbito geográfico para descarregarem apenas as páginas dentro desse âmbito.

- *Restrições ao nível do âmbito temporal:* estas restrições podem ser ao nível do âmbito temporal do conteúdo ou do momento da publicação da página.

  A restrição ao nível do âmbito temporal do conteúdo está relacionado com o processo de descarregar as páginas que estejam inseridas num dado intervalo de tempo ou época. Como já foi dito no capítulo 4, existem várias formas de obter informação temporal de uma página Web, mas aqui trata-se, não de extrair a informação temporal, mas sim de a analisar. Ao retirar as expressões temporais dos documentos Web, o robô vai conseguir perceber se uma página está ou não no âmbito temporal definido. Se estiver, essa página é descarregada, caso contrário ela é ignorada.

  Todas as páginas, para serem aceites, têm de se inserir num dado intervalo de tempo. Se assim for, as páginas podem ser mais tarde relacionadas e interrogações como *Einstein 1900 a 1910*, quando colocadas num motor de busca, poderão retornar informação que esteja inserida naquele intervalo de tempo e não apenas relativo a *1900* e *1910* como acontece actualmente.

  O âmbito temporal ao nível do momento de publicação da página, está relacionado com o momento temporal em que a página foi publicada e não com o seu conteúdo.

	Conteúdo Geral	Âmbito geográfico	Âmbito temporal
Definição das Restrições	Temas ou tópicos abordados.	Local ou espaço geográfico referenciado.	Intervalo de tempo ou época a que diz respeito.
Informação extraída	Termo ou palavras relativas ao tópico.	Localizações geográficas.	Datas e outras expressões temporais.
Avaliação dos documentos	Documentos similares ou classificados no tópico em causa.	Documento de acordo com o âmbito geográfico estabelecido.	Documentos com informação no intervalo de tempo ou época definida.

Tabela 3.2: Comparação entre robôs com diferentes tipos de restrições

Na Tabela 3.2 é apresentada a comparação entre robôs com âmbitos diferentes. Cada âmbito tem a sua forma de obter e analisar a informação pois cada tipo de informação diverge na forma como é analisada.

Para obter melhores resultados, estes três eixos podem ser combinados. Ao serem combinados a informação resultante é mais refinada e isso produz melhores resultados. A combinação entre os eixos fará com que a informação tenha mais relevância para o utilizador.

Embora a combinação entre os eixos seja benéfica, com este projecto apenas nos iremos concentrar na descarga temporal. Esta é uma área que não tem muito trabalho feito e é por isso o foco deste trabalho. Ainda não existe, actualmente, um mecanismo capaz de fazer descargas temporais. No entanto, existe alguma informação sobre o tema como veio sido mostrado ao longo deste documento.

3.2 Restrições temporais

Já se viu que a utilização de restrições é bastante útil para conduzir um percurso pela Web à procura de informação. As restrições permitem obter informação relevante de uma forma eficiente. A restrição temporal, o grande foco deste trabalho,

é uma dessas restrições.

Grande parte da informação na Web pode ser categorizada temporalmente, ou seja, pode ser colocada num eixo temporal e analisada. A informação temporal pode dizer respeito ao conteúdo ou aos metadados. A informação temporal por conteúdo significa que o conteúdo que está no documento está limitado temporalmente.

Por exemplo, se o conteúdo fala sobre a Segunda Guerra Mundial, então fará parte do intervalo entre os anos 1939 e 1945. Por metadados, significa que a referência temporal usada é a data de criação do documento Web. No entanto, como já foi discutido anteriormente, existem formas boas e menos boas de recolher informação temporal.

O objectivo deste trabalho tem a ver com a descarga temporal e não com a forma de obtenção das expressões. Tomemos o exemplo na Figura 3.1.

Na Figura 3.1 é possível ver páginas Web diferentes que se referem à mesma informação. Estão ilustrados vários exemplos de acontecimentos importantes cuja informação está dispersa.

6 de Agosto de 1945 simboliza o fim da Segunda Guerra Mundial, *11 de Setembro de 2001* simboliza o ataque às Torres Gémeas e *25 de Abril de 1974* simboliza a mudança de regime político em Portugal. Todos estes acontecimentos dizem respeito a um período no tempo e, por isso, podem ser relacionados. Mas a informação sobre os mesmos temas está espalhada pela Web o que torna difícil a sua obtenção.

As restrições temporais num robô servem para isto mesmo: percorrer a Web à procura de informação que esteja dentro do intervalo de tempo ou época definida. Cruzando o eixo temporal com o tópico, grande parte dos documentos sobre esse tópico podem ser obtidos de uma forma mais eficiente.

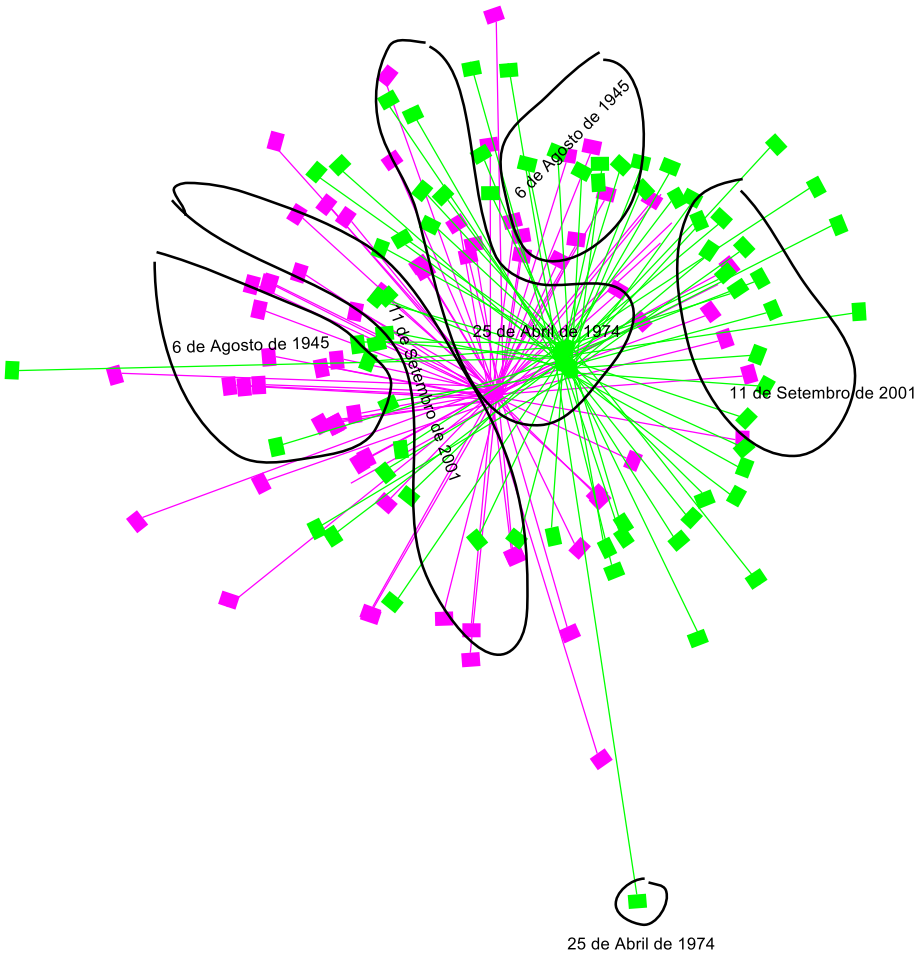


Figura 3.1: Distribuição da informação temporal pela Web.

Capítulo 4

Informação Temporal na Web

4.1 Intenção Temporal

Os motores de busca actualmente fazem o melhor que podem para conseguir fornecer a melhor informação possível aos seus utilizadores. Sempre que colocamos uma interrogação num deles, eles esforçam-se para nos conseguir dar a melhor resposta o mais depressa possível. Por vezes não acertam nem relacionam correctamente aquilo que queremos com aquilo que obtemos.

Relacionar temporalmente as interrogações pode ser uma mais-valia pois, na sua maioria, os motores de busca não conseguem estabelecer relações temporais implícitas nas interrogações, o que faz com que os resultados produzidos não sejam os melhores [19].

Recolher informação temporal é um tópico que tem vindo a atrair grande interesse nos anos recentes. O seu propósito é melhorar a recolha de documentos explorando a sua informação temporal, tornando assim possível colocar um tópico numa linha de tempo.

No entanto, apesar da maturidade relativa da área e do envolvimento constante da comunidade, poucos trabalhos utilizaram efectivamente a informação temporal para exploração e propósitos de procura.

Além disso, a maior parte das abordagens existentes baseiam-se na suposição

dos utilizadores forneceren algum tipo de informação de contexto temporal, o que não é sempre o caso.

Um exemplo ilustrativo disso é o *Campeonato do Mundo da FIFA na Alemanha* e *Guerra do Iraque de George Bush*. Estas são interrogações temporais implícitas que, apesar de não conterem expressões temporais, pode-se deduzir implicitamente a intenção temporal de 1974 e 2006 no primeiro caso e 1991 e 2003 no segundo.

Estabelecer relações temporais nas interrogações feitas é, por isso, da mais alta importância para melhorar a procura de resultados através da análise temporal das interrogações, análise dos eixos temporais ou através de aglomeração temporal. Pouco trabalho foi feito com este propósito.

A maior parte dos trabalhos feitos seguem uma abordagem baseada nos metadados que se foca na data de publicação dos documentos ou então uma abordagem com base nas interrogações feitas pelos utilizadores [19].

Existem alguns problemas com estas abordagens. A primeira abordagem não é significativa pois a data de criação do documento pode diferir em muito do seu conteúdo. Já a segunda pode mostrar ser bastante útil para compreensão de interrogações dos utilizadores mas que, fora dos grandes laboratórios industriais, pode ser muito difícil de alcançar.

Além disso, uma abordagem para buscar as intenções temporais de interrogações temporais implícitas com base em registos, depende da intenção do próprio utilizador e do facto de algumas versões da interrogação já terem sido feitas. Juntando a isto o facto de apenas 1.21% das interrogações incluírem datas [20], isto pode ser visto como uma desvantagem substancial.

Por isso é importante perceber quais as intenções do utilizador quando ele coloca uma interrogação no motor de busca. Utilizar as intenções temporais como meio de formular uma interrogação é uma tarefa particularmente difícil e pode tornar-se ainda mais se o utilizador não for claro com o seu propósito.

Por exemplo, um utilizador que insere a interrogação *"Lady Gaga"* pode desejar encontrar a página oficial da cantora ou outro tipo de informação (como curiosidades) sobre ela. No entanto, poderá também querer explorar a informação

biográfica, informação temporal relacionada com a sua discografia ou as datas esperadas da sua digressão [20].

Tentar perceber a natureza temporal de uma interrogação, principalmente das implícitas, é dos desafios mais interessantes para a recolha de informação temporal. Mas para isso é necessário estudar essas interrogações e tentar perceber quantas delas têm intenção temporal [20].

4.2 Factos e Expressões Temporais

A verdadeira informação temporal não está nas interrogações feitas, mas sim nos documentos. A informação contida nos documentos é a que realmente interessa aos utilizadores sempre que fazem uma interrogação num motor de busca.

Como tal, estudar as expressões temporais mais relevantes contidas na Web é algo que pode ajudar a criar mecanismos e estratégias para adequar a pesquisa temporal de acordo com a realidade da Web [7].

A relevância das expressões Web é algo que não tem tido uma grande atenção. Apenas existe a preocupação sobre como encontrar e marcar as expressões e não em tentar compreendê-las. Perceber se uma expressão temporal é relevante para uma dada interrogação é algo importante.

As expressões temporais são, normalmente, consideradas isoladamente. Não há métodos para calcular a relevância de uma dada expressão no geral ou de acordo com uma interrogação.

Criar um sistema para calcular a relevância de expressões é algo que ajudará na forma como classificamos os documentos de acordo com uma interrogação ou como os dispomos numa linha de tempo.

Em vez de se olhar para todas as expressões temporais com a mesma relevância, é necessário dar mais ou menos importância a essa expressão de acordo com o documento, o contexto ou a interrogação fornecida.

Uma abordagem para fazer isto foi apresentada em [7]. Aqui foi criada uma forma de atribuir uma classificação às expressões (tal como um algoritmo de clas-

sificação de páginas de um motor de busca atribui aos documentos) e, de acordo com a interrogação, essa expressão terá mais ou menos pontuação. Isto permite que para além do tópico em causa, a dimensão tempo seja incluída nessa pesquisa e possa assim devolver resultados melhorados.

Por exemplo, com a pesquisa *Einstein 1900 a 1910*, iremos querer documentos que falem sobre Einstein ter terminado a sua tese em 1905 e, por outro lado, informação contendo as duas partes da interrogação, isto é, a parte textual e temporal. Apenas se o sistema souber que *1905* está no intervalo da interrogação feita é que ele pode devolver *1905* como uma resposta correcta à interrogação.

Identificar as expressões temporais mais relevantes é uma tarefa bastante complicada devido ao enorme número de expressões temporais mencionadas em vários tipos de documentos. Os artigos seleccionados da *Wikipedia* têm uma média de 80 expressões por artigo [21]. Existindo uma fonte tão grande como esta com informação temporal, poderá ser bom começar a aproveitá-la para extrair informação temporal de eventos ocorridos para ajudar na criação de uma base de dados de conhecimento.

O grande sucesso da *Wikipedia* e o progresso das técnicas de extracção automatizadas levou à construção de algumas grandes bases de dados de conhecimento. Infelizmente, a maioria delas foca-se em factos estáticos e ignoram a dimensão temporal dos factos, mesmo que a maioria dos factos evolua durante o tempo ou apenas sejam válidos durante um período de tempo [21].

A dimensão temporal é particularmente importante em relações um-para-um como ***éCasadoCom*** ou ***éChefeDe***. Uma base de dados de conhecimento que contém múltiplas esposas para uma dada pessoa apenas é consistente se anexar intervalos de tempo aos factos. Além disso, a dimensão temporal ajuda a distinguir factos actuais de factos passados.

Adicionar a dimensão temporal às bases de dados de conhecimento fornece novas e melhores aplicações sobre como visualizar linhas de tempo para pessoas ou eventos importantes, inferir a ordem cronológica e, com isso, talvez a causalidade dos factos e eventos, analisando a co-relação entre eles.

Bases de dados de conhecimento com requisitos temporais, podem ser uma

perfeita mais-valia para historiadores, analistas de media ou investigadores sociais.

A informação temporal pode dizer-se que é bem definida, isto é, existem sempre dois pontos (um de início e um de fim). Mais, a informação temporal pode ser normalizada para um formato padrão tornando assim mais fácil a análise e troca de informação. A informação temporal pode também ser organizada hierarquicamente e ter diversas granularidades (ano, mês, ...) tornando possível o agrupamento da informação temporal [22].

A informação temporal nos documentos apenas pertence a uma de quatro tipos: datas (*21 de Janeiro de 2013*), horas (*14:56*), duração (*duas semanas*) ou frequência (*duas vezes por semana*) e pertencem a um de três tipos de expressões: explícitas, implícitas ou relativas.

As expressões explícitas referem-se a um ponto específico no tempo, ou seja, contêm toda a informação explicitamente. Estas expressões podem ser normalizadas sem necessitarem de conhecimento adicional [22].

As expressões implícitas são, por exemplo, nomes de feriados, nomes de celebrações importante ou acontecimentos relevantes. *Dia de Natal de 2012* ou *Dia do Trabalhador de 2012* são exemplos de expressões temporais implícitas. Estas datas podem ser normalizadas para *25/12/2012* e *01/05/2012* respectivamente, mas necessitam de mais informação para poderem ser normalizadas. Como não são explícitas, é necessário conhecimento sobre os eventos aos quais se estão a referir.

As expressões relativas são aquelas que se obtêm por análise das relações temporais inseridas no texto. Por exemplo, *"há duas semanas atrás"* é uma expressão relativa que tem o dia de hoje como o ponto final do intervalo. Para saber a que dia se refere efectivamente, é preciso retirar ao dia de hoje 14 dias para encontrar no tempo o ponto inicial.

Eis um exemplo de uma expressão temporal relativa retirada de um segmento de uma notícia extraída da página do Correio da Manhã[1]:

[1]Notícia acessível em: http://www.cmjornal.xl.pt/detalhe/noticias/nacional/economia/governo-prepara-subsidio-de-desemprego-para-funcao-publica

Data de criação do documento: 2013-01-21, 13:55

"A coordenadora da Frente Comum, Ana Avoila, afirmou nesta segunda-feira que o Governo propôs (...)."

Segunda-feira é uma expressão relativa que se refere, neste caso, ao mesmo dia em que a notícia saiu, 21 de Janeiro de 2013.

4.3 Proliferação da Informação

A dimensão temporal torna-se novamente importante quando queremos detectar o ponto de partida da informação. Actualmente, qualquer pessoa pode publicar informação na Web. Seja através de blogs ou através das redes sociais, ela facilmente se propaga.

Num futuro próximo, saber quando é que a informação primeiro apareceu na Web quase que equivale a saber quando é que a informação foi tornada pública para a pessoa comum. Assim, a descoberta da primeira aparição da informação na Web é um problema importante para a sociedade [23].

Perceber a disseminação de uma dada informação é importante para perceber, primeiro de onde partiu, depois qual foi o seu grau de propagação e de que forma essa propagação ocorreu.

Por exemplo, recentemente muitos rumores tiveram o seu início na Web, depois espalharam-se e, no fim, tornaram-se famosos.

Tentar perceber a origem e a propagação da informação é bastante importante [23]. A Figura 4.1 mostra um exemplo da dispersão de uma dada informação pela Web. Os pontos verdes e roxos dizem respeito a dois pedaços de informação distintos.

Vemos que existem muitos pedaços da mesma informação espalhados pela Web onde conseguimos também ver a origem de cada um deles. A origem é o ponto central de onde todas as linhas têm o seu começo. Esse ponto foi o iniciador da informação e saber esse ponto trará benefícios para para determinar locais na Web com fontes ricas em informação [23].

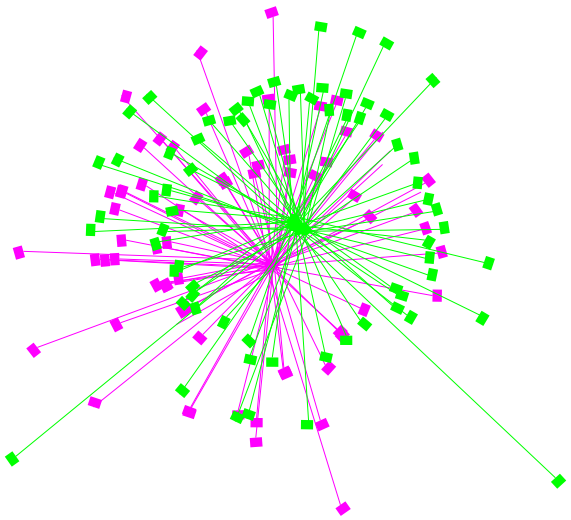


Figura 4.1: Exemplo da dispersão da informação pela Web.

A tecnologia Web actual não permite estabelecer relações entre estes dois tipos de informação. Por causa disso, foi proposta uma abordagem para tentar estabelecer relações entre a primeira página onde apareceu primeiro a informação e como essa mesma informação se espalhou pela Web.

Para detectar em que lugar apareceu a informação, primeiro é preciso retirar as expressões temporais da página e perceber se são válidas ou não.

Ora, a data de criação do documento, não é uma forma fiável, e, como é dito em [24], a tecnologia Web actual não permite a obtenção fiável de datas de criação para páginas arbitrárias. O foco da investigação são páginas que publicam nova informação como artigos de páginas de notícias ou páginas de blogs.

Hoje em dia, tais páginas são criadas através de um CMS *(Content Management System)* tal como Wordpress [25] ou Joomla [26] e, normalmente, estes embebem nos artigos as datas de criação.

Estas marcas não são muito fiáveis pois, na maior parte das vezes, para além de não indicarem o fuso horário, têm os tempos mal colocados [12].

Em [23], outra técnica é proposta para tentar detectar qual a primeira página a

obter a informação. Este método consiste em recolher todos os documentos Web que contêm um dado tópico, depois organizá-los no tempo e, por fim, escolher o primeiro como sendo o iniciador do tópico.

Mas todos os métodos estão susceptíveis a erros assim sendo, é importante encontrar uma forma para combater esses erros, isto é, o ruído. A ideia aqui pode passar por examinar não só as propriedades de cada página mas também as relações temporais existentes entre elas [27, 28].

Após terem sido seleccionadas as páginas candidatas e as suas marcas temporais, é criada uma linha temporal provisória.

Como as linhas temporais de informação real seguem padrões normais, esta linha temporal provisória tem, potencialmente, muita informação para servir de eliminação de ruído.

Capítulo 5

Segmentação Temporal

O método descrito em [3, 4, 8] baseia-se num conjunto de padrões temporais, baseados em expressões regulares, usados para identificar e classificar as expressões temporais encontradas em textos Portugueses. Os padrões são criados utilizando palavras co-ocorrentes, determinadas a partir de um conjunto de palavras-chave que são referências temporais em Português.

Este método baseia-se numa abordagem em duas fases, senda cada fase levada a cabo por um módulo diferente. A primeira fase é executada pelo processador de co-ocorrências (daqui para a frente COP: *Co-Ocurrence Processor*) e a segunda fase é executada pelo módulo de Anotação.

Os módulos funcionam da seguinte maneira. Primeiro o módulo COP cria os padrões temporais baseando-se em corpora de treino e num conjunto de palavras que está dividido em duas partes: marcadores léxicos e marcadores gramaticais. Depois, estes padrões são utilizados pelo módulo de Anotação para executar a anotação das expressões temporais em Português. A Figura 5.1 mostra um diagrama da arquitectura do modelo e a interligação dos módulos.

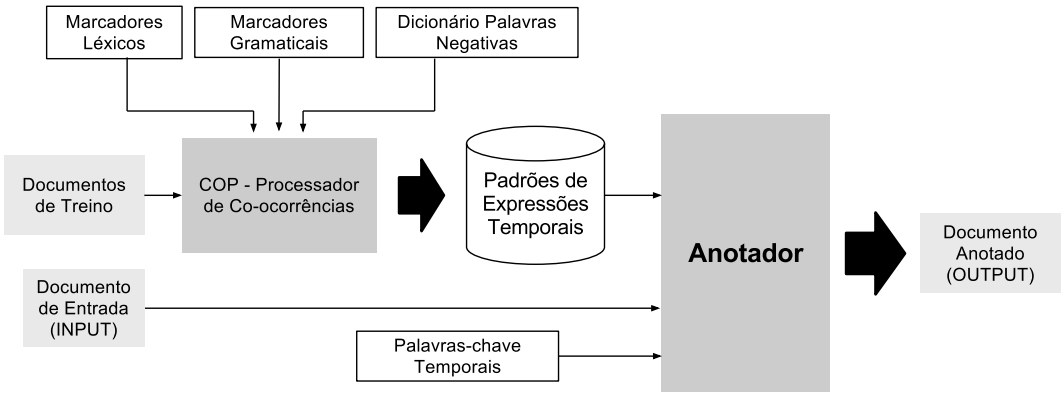


Figura 5.1: Arquitectura do modelo e interligação dos módulos (adaptado de [3])

5.1 Esquema de Anotação

As expressões temporais são anotadas de acordo com directrizes temporais definidas em [29]. A Anotação abrange uma única identificação, uma categoria que é o TEMPO, um tipo (calendar_ref, duração ou frequência) e um subtipo apenas para o tipo CALENDAR_REF (data, duração ou frequência). Este é um exemplo da como a ferramenta faz a anotação de expressões temporais:

(1) Eu estava em Berlim <EM ID="1" CATEG="TIME" TYPE="CALENDAR_REF" SUBTYPE="DATE"> em 2008 </EM>.

(2) Eu vou visitar os meus pais <EM ID="2" CATEG="TIME" TYPE="FREQUENCY" > todos os dias </EM>.

5.2 Processador de Co-Ocorrências

O objectivo do módulo COP é criar um conjunto de padrões para ser usado pelo módulo de Anotação. O COP pode ser executado em cima de vários documentos para assim conseguir um número considerável de padrões que enriquecem a fase de anotação. De notar que o módulo COP apenas é necessário para recolher o conjunto de padrões e, uma vez obtidos, o módulo COP não é mais utilizado. No

entanto, os padrões podem ser ajustados mais tarde para melhorar o processo de identificação das expressões temporais.

O módulo COP analisa os documentos de treino, determina a combinação de palavras e a sua frequência e utiliza uma abordagem estatística para decidir que padrões devem ser criados de acordo com as co-ocorrências encontradas. O módulo COP tem vários passos de execução.

O primeiro passo cria uma lista composta pelas expressões temporais encontradas e a sua frequência. Estas expressões foram encontradas utilizando os marcadores léxicos. Estes marcadores devem ser compostos por todas as palavras Portuguesas das quais podem ser compostas expressões temporais (e.g. meses, estações, dias de semana, unidades temporais de medida como dia, semana, mês, ano, ...).

Este conjunto de palavras é utilizado para detectar as suas co-ocorrências que estão presentes num máximo de *n* palavras antes e/ou *n* palavras depois. Um exemplo duma expressão temporal utilizando a palavra ano e *n=2* é *"No **ano** passado"* e *"No próximo **ano** de 2013"*.

No segundo passo, a lista de expressões temporais é podada, especificamente as expressões que não fazem qualquer sentido semântico, num contexto de linguagem, são removidos da lista utilizando os marcadores gramaticais. No entanto, as expressões que apenas contêm marcadores léxicos e gramaticais são mantidos na lista, desde que nenhuma palavra negativa exista na sua proximidade.

Por exemplo, na frase *"A rua 1º de Maio"*, a expressão *"1º de Maio"* não é uma expressão temporal porque é o nome de uma rua. Como a palavra *"rua"* é uma palavra negativa, ela é excluída.

O próximo passo agrega expressões temporais encontradas no passo anterior de acordo com as regras seguintes.

Primeiro, as expressões temporais são agregadas se elas contiverem uma data ou uma referência temporal. Por exemplo, *"Em Abril"* e *"Em Maio"* são agregadas numa única expressão com a tag especial *"Em tag_MONTH"*.

Em segundo, as expressões temporais são agregadas se elas contiverem mais

do que uma co-ocorrência com a mesma palavra temporal na mesma posição. Por exemplo, as expressões *"No **ano** passado"* e *"No **ano** seguinte"* são agregadas para *"No **ano** passado | seguinte"*. A frequência das expressões agregadas é a soma da frequência de cada expressão. A lista resultante é ordenada descendentemente pela frequência.

Finalmente, os padrões são definidos por expressões regulares. Para cada padrão está associada a classificação de acordo com as directrizes descritas em [29].

5.3 Anotador de Expressões Temporais

O objectivo do anotador é identificar e classificar expressões temporais Portuguesas com a anotação relativa escrita no texto original, através de padrões definidos pelo módulo COP. Após o texto ser dividido em frases, cada uma delas é processada em cinco passos. O primeiro passo é introduzido para melhorar o desempenho pois exclui todas as frases que não têm qualquer expressão temporal.

Por exemplo, a frase *"Lisboa é a capital de Portugal"* não é processada mas a frase *"**Hoje** está bom dia"* é processada.

A geração de expressões temporais candidatas é feita no segundo passo. Primeiro identifica as expressões de tempo e de data que podem ser datas completas ou incompletas. Depois, essas expressões são marcadas com uma "*tag* especial"como tag_DATE, tag_MONTH, tag_YEAR, tag_WEEK.

No terceiro passo, o método verifica se as frases têm correspondência com algum padrão de expressões temporais. Neste caso, cada frase é anotada com uma classificação semântica correspondente ao padrão igualado (quarto passo).

Por fim, as "*tags* especiais"são substituídas pelo texto original.

5.4 Módulo de Segmentação Temporal

O objectivo da segmentação temporal de documentos é a representação temporal de um documento, incorporando a dimensão temporal no modelo de recolha de informação de forma a melhorar a qualidade dos resultados. Para fazer isso, primeiro é necessário identificar as expressões temporais e, sempre que possível, os seus valores temporais normalizados.

Uma expressão de *data* denota um momento que se pode encontrar num calendário, como *15/06/2010*. Uma expressão horária tem uma granularidade mais baixa que o dia, referindo-se a entradas de um relógio, como por exemplo, *14:30*. Um *intervalo* é uma expressão complexa composta por duas expressões temporais simples agrupadas por um conector, como por exemplo *de Abril a Maio de 2009*. Uma expressão de *duração* representa a quantidade de tempo como por exemplo, *durante dois meses*. Expressões de *frequência* denotam a repetição no tempo, por exemplo, *diariamente*.

A resolução envolve a interpretação e normalização das expressões temporais. A interpretação consiste na inferência de uma nova data, utilizando informação de um documento, como a marca temporal do documento ou uma data posicionada antes ou depois do texto. Desta forma, a expressão reconhecida, que não é uma data completa ou explícita, é mapeada para uma data mais completa.

É igualmente importante a classificação da expressão e a forma como as referências são expressas no texto. A definição de expressões como sendo explícitas, implícitas, relativas ou vagas, é suportado pela proposta definida em [30].

Uma referência explícita é uma data ou expressão temporal referindo-se directamente a uma entrada num calendário ou num relógio.

Uma referência implícita é uma expressão que não é uma data explícita mas pode ser directamente ancorada a um calendário/relógio, como feriados ou eventos. Por exemplo, *Natal de 2010* pode ser mapeado para *25/12/2010*. A eficácia da resolução destas expressões está dependente da utilização de ontologia temporal adequada. *Referências indiciais* são expressões que precisam de um ponto no tempo para poderem ser totalmente resolvidas e ancoradas a um ca-

lendário/relógio. Estas referências podem ser *deictic timexes* ou *anaphoric timexes*, segundo Ahn em [31]. *Deictic timexes*, como *hoje, no próximo mês*, são resolvidos utilizando a marca temporal do documento. *Anaphoric timexes*, como *na próxima semana, um dia antes*, são resolvidos utilizando um ponto temporal relevante invocado no texto anteriormente.

Referências vagas são expressões que são difíceis de precisar no tempo porque começam e/ou acabam em pontos que não são claros, como por exemplo *daqui a algumas semanas* ou *recentemente*.

Finalmente, a normalização é a transformação dessas datas para um formato normalizado, ancorado a um calendário/relógio através de linhas de tempo definidas por pontos, denotados por *chronons*. A linha de tempo pode ter diferentes níveis de granularidade como ano, mês, semanas, dia, hora.

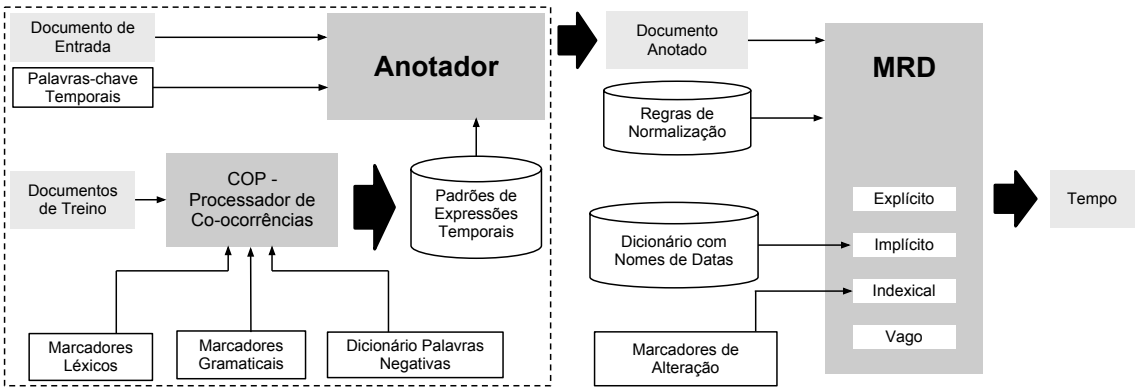


Figura 5.2: Arquitectura do sistema de segmentação temporal (adaptado de [4])

O sistema para a identificação e extracção temporal, ilustrado na 5.2, é composto por três módulos principais: Processador de Co-ocorrências (COP), o Anotador e o Módulo de Resolução de Datas (MRD).

O módulo COP cria padrões temporais classificados semanticamente. Estes padrões são utilizados pelo Anotador para identificar e classificar expressões temporais Portuguesas, anotando-as no texto original. Por fim, o MRD pega nessas anotações e mapeia as expressões temporais para que possam ser traduzidas em

chronons.

A tarefa do MRD é transformar as anotações feitas no texto original e mapeá-las para uma representação discreta no tempo, denotada por *chronons* em [32]. Esta representação é aplicada para ancorar os documentos num calendário ou num relógio, assumindo as quatro linhas de tempo $T = \{T_h, T_d, T_m, T_y\}$ para horas, dias, meses e anos, respectivamente. A linha de tempo base, T_d, é uma linha temporal com a granularidade ao dia, definindo um intervalo de dias *chronons* sequenciais. Desta forma, intervalos são normalizados com dois *chronons*, um para cada limite do intervalo.

O formato padrão *YYYY-MM-DD hh:mm:ss*, especificado na norma *ISO 8601:2004*, é utilizado para representar os *chronons*. Por exemplo, *"1 de Maio, 2010"* é normalizado como *2010-05-01* e pode ser ancorado a T_d. Se a expressão está incompleta então o algoritmo faz uso do marcador *X*. Por exemplo, *"Junho de 2000"* é normalizado como *2000-06-XX* e, uma vez que não pode ser ancorado à linha de tempo base, uma granularidade menor é utilizada como T_m ou T_y.

Este módulo baseia-se num conjunto de regras, utilizadas para interpretar expressões temporais previamente anotadas pelo módulo de Anotação. Começa pela normalização da marca temporal do documento.

A marca temporal do documento que é uma data relacionada com os metadados de um documento, como a data de criação ou de publicação de um documento, é muito importante para que possam ser resolvidas referências temporais, como os *deictic timexes*.

A capacidade para resolver referências implícitas e ancorá-las numa linha de tempo baseia-se na ontologia subjacente ao tempo da abordagem escolhida. Neste sistema, foi definido um dicionário com nomes de datas onde esta informação está guardada. Por exemplo, *"Dia de Natal 2010"* pode ser representado como uma referência explícita a *25 de Dezembro de 2010*, que é normalizado para *2010-12-25*.

A resolução de referências indiciais segue um processo mais complexo. Isto ocorre porque, primeiro, eles podem precisar da marca temporal do documento (*deitic timexes*) ou outro tipo de referência invocada no texto (*anaphoric timexes*)

e segundo, porque estas referências podem mencionar um passado, presente ou evento futuro.

Os modificadores, como *seguinte, anterior, depois, antes, de seguida* ou *mais tarde*, permitem a obtenção da regra correcta para ser aplicada a cada um destes casos.

Por exemplo, a expressão *"No próximo ano"* é resolvida como *"marca temporal do documento + 1 medida_de_tempo(y)"*. Para esta fase do trabalho, os *anaphoric timexes* são resolvidos correctamente apenas se eles contêm um marcador de alteração.

Capítulo 6

Análise Temporal de uma colecção Web

A utilização de uma colecção Web permite que possa ser feita uma análise mais cuidada sobre um conjunto determinado de documentos. Estes documentos são estáticos significando que não vão sofrer alterações durante o estudo dos mesmos.

Neste caso específico, a colecção Web estudada foi a WPT03 [33].

A WPT03 é uma colecção com mais de 3 milhões de documentos recolhidos pelos robôs do motor de busca Tumba! e produzida pelo pólo XLDB da Linguateca. Engloba conteúdos em português de páginas alojadas em domínios .pt ou escritas em português e alojadas sobre um domínio .com, .org, .net ou .tv desde que tenham sido referenciadas por, pelo menos, uma página alojada sob um domínio.pt.

A colecção, contém os documentos tal como foram recolhidos, isto é, sem a aplicação de qualquer tipo de pós-processamento, filtragem ou uniformização da codificação. Esta colecção tem um número total de 3.775.611 documentos dos quais 1,5 milhões são únicos. O número de documentos escritos em português é de 2.583.176.

6.1 Introdução

O grande objectivo deste capítulo é analisar temporalmente os documentos Web para estudar como é que a informação temporal se encontra nas páginas Web. Algumas conclusões e decisões futuras poderão ser retiradas com base no estudo estatístico desta colecção.

A análise da informação temporal na Web, neste caso da Web Portuguesa, irá ser feita estudando certos aspectos que podem ser recolhidos, analisando estatisticamente os documentos contidos na colecção.

Antes da colecção puder ser analisada temporalmente, os documentos dela teriam de sofrer um pré-processamento para estarem aptos a serem processados pela ferramenta de segmentação temporal.

A colecção WPT03 é composta por várias páginas Web. Na Figura 6.1 está um exemplo que ilustra o formato em que essas páginas se apresentam.

Cada página começa por indicar qual o URL correspondente (linha 2). A partir daí, aparecem os campos que dizem respeito aos metadados de cada página (linhas 4 a 17) e depois o conteúdo das páginas.

O número de propriedades nos metadados não é fixo, por isso, os metadados acabariam quando houvesse uma linha em branco (linha 18).

O conteúdo de cada página acaba quando aparecer uma linha igual à linha 1.

Na Figura 6.2 está ilustrada a mesma página da Figura 6.1 mas após o processamento. Podemos ver a estrutura XML definida que tem como raíz *colCHAVE* (linha 1). Dentro desta raíz serão declarados todos os documentos.

Para converter as páginas para o formato certo, o campo da data de modificação (linha 6 da Figura 6.1) teria de ser utilizado. Essa utilização está clara na linha 3 da Figura 6.2. A data está sem espaços e colocada no formato *AAAAMMDD* (A - Ano, M - Mês, D - Dia).

```
1  ----------------------------------------
2  URL: http://100.pt
3  ----------------------------------------
4  (Content-Length, 582)
5  (Content-Type, text/html)
6  (Last-Modified, 30/08/2002)
7  (ServerSW, Apache)
8  (dataRec, 21/03/2003)
9  (estado, 200)
10 (filtrado, /vcrMom/data/WEBSTATS/filtrados/33/44/0)
11 (host, 100.pt)
12 (ip, 195.22.10.103)
13 (language, portuguese)
14 (prof, 0)
15 (realSize, 582)
16 (textSize, 108)
17 (title, Brevemente... Um projecto 100 Limite, Lda.)
18
19 Brevemente... Um projecto 100 Limite, Lda.
20 Brevemente... Um projecto 100 Limite
21 100limite@100limite.com
```

Figura 6.1: Exemplo de um documento antes do pré-processamento

O conteúdo da página é colocado dentro do nó *<TEXT>* (linha 6) e todos os parágrafos do documento original teriam de estar entre as marcas *<P>* e *</P>* (linhas 7 a 10).

Quando o conteúdo acabar, o nó *TEXT* (linha 11) e o nó *DOC* (linha 12) são fechados, dando por concluída a conversão da página.

Este processo é repetido e as restantes páginas, continuam a ser convertidas e a serem acrescentadas (linha 13) ao nó *colCHAVE*.

```
1 <colCHAVE>
2  <DOC>
3    <DATE>20020830</DATE>
4    <DOCID>1</DOCID>
5    <DOCNO>1</DOCNO>
6    <TEXT>
7       <P> </P>
8       <P> Brevemente... Um projecto 100 Limite, Lda.</P>
9       <P> Brevemente... Um projecto 100 Limite</P>
10      <P> 100limite@100limite.com</P>
11   </TEXT>
12 </DOC>
13 <DOC>
14   ...
15  </DOC>
16 </colCHAVE>
```

Figura 6.2: Exemplo de um documento após o pré-processamento

Depois de todos os ficheiros estarem no formato apropriado, a ferramenta descrita no capítulo 5 foi aplicada a todos eles para serem processados.

Após o processamento dos documentos, era preciso analisá-los para retirar deles toda a informação necessária. Para guardar a informação sobre cada documento, a informação seria colocada numa base de dados para posterior análise.

Assim, foi criada uma aplicação para analisar os mais de 9 milhões de documentos processados pela ferramenta de segmentação. Esta aplicação também seria responsável por guardar toda a informação recolhida na base de dados.

Após a informação ser inserida na base de dados, é possível trabalhar as interrogações de modo a obter aquilo que queremos. A utilização da base de dados faz com que os documentos apenas tenham de ser processados e analisados uma só vez.

Para fazer as interrogações à base de dados, foi criada outra aplicação para esse efeito. Esta está construída para interpretar os vários dados presentes para conseguir obter a informação mais pertinente sobre a temática da informação temporal nos documentos Web.

6.2 Documentos

Começemos pela análise aos documentos. Olhando para a Tabela 6.1, temos que de um total de 2.583.176 documentos, 73,66% deles (1.902.700) contêm informação temporal. Esta informação temporal podem ser datas ou marcas de um relógio.

Logo à partida conseguimos perceber que vão existir documentos que não são analisados através da dimensão temporal. Na realidade, significa que um documento, mesmo relevante para um determinado âmbito temporal, não será devolvido como resultado. Isto faz com que os resultados finais possam não ser os melhores.

Tabela 6.1: Quantidade de Documentos com Informação Temporal

Documentos	Total	Percentagem (%)
Todos	2.583.176	100,00
Com Informação Temporal	1.902.700	73,66

É importante realçar que a qualidade dos resultados obtidos depende da ferramenta de análise temporal utilizada.

A data de criação do documento é a data em que a página Web passou a existir, ou seja, a data do ficheiro ao qual acedemos. A data de criação poderá ser importante para encontrar informação temporal relativa. Por exemplo, se durante o texto aparecer uma expressão temporal implícita como "há duas semanas", a data de criação do ficheiro permite encontrar a data efectiva à qual a expressão se refere.

Olhemos para a Tabela 6.2.

Tabela 6.2: Documentos com e sem a Data de Criação

	Total de Documentos	Percentagem (%)
Com data de criação	960.495	37,18
Sem data de criação	1.622.681	62,82

Nesta colecção, quase 63% dos documentos não têm a data de criação. Embora este elemento possa ser importante para estabelecer relações temporais no texto, não é a única forma de o fazer. A ferramenta de segmentação consegue também estabelecer relações temporais relativas através de expressões temporais explíctas anteriormente encontradas no texto.

6.3 Expressões Temporais

As expressões temporais são todas as expressões das quais é possível retirar um momento no tempo. Assim sendo, começemos por olhar para a Tabela 6.3.

Tabela 6.3: Número médio de expressões por Documento

	Total
Por Documento	7,5
Por Documento com Informação Temporal:	10,2

Pela Tabela 6.3 podemos ver que cada documento tem uma média de quase 8 marcas temporais por documento. Isto é relativamente bom, pois significa que existe muita informação que pode ser extraída e posteriormente utilizada.

Se olharmos apenas para o universo dos documentos com informação temporal, o número de expressões sobe para 10 expressões por documento.

Olhemos para a 6.4 para percebermos qual a quantidade e qual a qualidade das expressões temporais que a ferramenta de segmentação temporal consegue encontrar e resolver.

Tabela 6.4: Número de Expressões

	Total de Expressões	Percentagem
Resolvidas	18.675.134	96,21
Não Resolvidas	736.476	3,79

De um total de quase 19,5 milhões de expressões, mais de 96% foram resolvi-

das. Isto quer dizer que quase a totalidade das expressões podem ser colocadas num ponto no tempo. Estes números aumentam a qualidade da informação temporal presente nos documentos visto que quase todas as expressões são usáveis.

Tabela 6.5: Número Médio de Expressões Resolvidas Por Documento com informação temporal

	Total
Número de Expressões	9,8

Ao pegar nos resultados das expressões resolvidas e no universo dos documentos com informação temporal, podemos comparar os resultados da Tabela 6.3 com os da Tabela 6.5 onde vemos que quase 100% das expressões que são encontradas são resolvidas.

Estes resultados levam a que a probabilidade de extrair informação de qualidade seja mais alta, visto que as expressões temporais que são identificadas conseguem ser quase todas resolvidas.

Usando a Tabela 6.6, facilmente se percebe que a maior parte dos documentos tem entre 0 e 3 expressões temporais (mais de 1,6 milhões de documentos).

É interessante verificar que, a partir daí a quantidade de documentos que têm entre 4 e 8 expressões é apenas ligeiramente maior do que a quantidade de documentos com mais de 9 expressões temporais.

Tabela 6.6: Distribuição dos Documentos por Número de Expressões

Número de Expressões	Total de Documentos	Percentagem (%)
0	680.476	26,34
1	398.568	15,43
2	392.257	15,19
3	208.170	8,06
4	159.103	6,16
5	103.313	4,00
6	83.178	3,22
7	69.955	2,71
8	57.334	2,22
9 ou mais	430.822	16,68

O estudo do local da página em que as expressões aparecem, também pode ser feito. Dividindo o documento em quartis, podemos saber quantas expressões estão em cada quartil.

Tabela 6.7: Distribuição da Posição das Expressões pelos Documentos

	Total de Documentos	Percentagem (%)
1º Quartil	801.796	31,04
2º Quartil	407.578	15,78
3º Quartil	713.443	27,62
4º Quartil	660.359	25,56

Pela Tabela 6.7, vemos que, à excepção do 2º quartil, as expressões temporais encontram-se divididas em número semelhante pelos restantes quartis.

6.4 Segmentos Temporais

Os segmentos temporais dividem o texto em vários pedaços que correspondem a um período temporal ou a um conjunto deles. Isto significa que o texto contido dentro de um dado segmento corresponde ao período definido nesse mesmo segmento.

Algo que ainda não foi analisado e que, devido à importância dos segmentos, o deve ser, é a granularidade das várias datas associadas aos segmentos temporais.

É através dos segmentos que um dado pedaço de texto é identificado como fazendo ou não parte de um âmbito temporal, torna-se importante estudar a granularidade das datas associadas a eles.

Na Tabela 6.8, é apresentada a granularidade das datas dos segmentos temporais.

Tabela 6.8: Granularidade das Datas dos Segmentos Temporais

Tipo de Data	Quantidade	Percentagem (%)
Completa	3.021.972	13,47
Ano	15.579.822	69,44
Ano e Mês	3.345.587	14,91
Mês	6.822.162	30,41
Mês e Dia	5.709.199	25,45
Dia	5.709.274	25,45

Pela Tabela 6.8, vemos que quase 70% das datas dos vários segmentos têm o ano presente, 30% têm o mês e 25% têm o dia. Vemos também que apenas 13,5% das datas estão completas. Datas contendo apenas o ano e mês, os resultados aumentam para os quase 15%.

Pela análise da Tabela 6.8, rapidamente nos apercebemos que a maioria das datas tem presente o ano. Isto significa que, se utilizarmos apenas anos no âmbito temporal, vamos ter melhores resultados.

A utilização de âmbitos temporais delimitados apenas por anos serão aqueles que melhores resultados trarão devido à baixa quantidade de datas com mês e dia.

Tabela 6.9: Estatísticas sobre segmentos

	Total
Média da quantidade de datas	1,53
Média em dias do âmbito temporal	39,69
Posição Média dos Segmentos no Documento	0,55

Na Tabela 6.9, é apresentada a média da quantidade de datas a que cada segmento diz respeito, a média, em dias, que cada segmento abrange e a posição média dos segmentos pelo documento.

A quantidade média de datas em cada segmento é de 1.5. Isto significa que a maioria dos segmentos diz respeito a mais do que um momento temporal. Isto torna possível o estabelecimento de intervalos temporais nos segmentos.

Por causa disso, é possível estudar o âmbito temporal médio por segmento. Na Tabela 6.9, vemos que esse âmbito tem uma média de quase 40 dias. Significa isto que os segmentos falam sobre factos que são próximos uns dos outros.

Estudando a posição em que os segmentos aparecem no documento, vemos que o 3º quartil é o local em que mais vezes aparecem. Não poderia deixar de ser assim, visto que os segmentos são construídos com base nas expressões temporais e é no 3º quartil que as expressões temporais mais vezes aparacem (Tabela 6.7).

O número perto de 0.5 significa que a diferença não será muito grande e que, os segmentos estarão espalhados por todos os quartis de uma forma relativamente uniforme.

Tabela 6.10: Distribuição de Segmentos por Quartil

	Total de Documentos
1º Quartil	2.932.536
2º Quartil	3.594.209
3º Quartil	3.773.435
4º Quartil	4.367.044

Na Tabela 6.10, é apresentada a distribuição dos segmentos pelos quartis do documento. Vemos que a distribuição dos segmentos pelos quartis, é relativamente uniforme, sendo que o 1º quartil é aquele com menos segmentos (quase 3 milhões) e o 4º aquele com mais (quase 4,4 milhões).

6.5 Ligações

As ligações são a informação mais importante para um robô. Um robô percorre a Web saltando de página em página através das ligações presentes nelas. Um robô temporal tem de descarregar páginas que sejam relevantes para um dado período de tempo. Uma ligação será relevante para esse período de tempo caso se encontre dentro de segmentos que correspondam a esse mesmo âmbito.

Passemos a analisar as ligações que se encontram nesta colecção de documentos.

Tabela 6.11: Quantidade de Ligações e o seu Destino

	Total	Percentagem
Ligações que apontam para dentro da colecção	1.108	1,57
Ligações que apontam para fora da colecção	69.649	98,43

Olhando para a Tabela 6.11, vemos de imediato que, para os quase 2,5 milhões de documentos, apenas existem cerca de 70 mil ligações. Isto é um número muito baixo, demonstrando a fraca qualidade desta colecção no que toca a ligações disponíveis.

Das quase 70 mil ligações, apenas 1.108 (1,57%) apontam para documentos que estão dentro desta colecção. Embora estas ligações sejam poucas, permitem que as datas dos documentos de partida e chegada possam ser comparadas bem como o âmbito dos mesmos.

Tabela 6.12: Diferenças, em, dias entre âmbito temporal e datas de partida/chegada dos documentos

	Média	Mínima	Máxima
Datas Documento	238,55	0	1.087
Âmbito temporal	43,30	0	26.487

Olhando para a Tabela 6.12, vemos que a diferença média de dias entre cada documento é de 238. Isto significa que as ligações são criadas muito depois da

criação dos documentos originais. Os valores para esta diferença estão entre 0 (documentos com a mesma data) e 1087 dias.

Ao relacionarmos o âmbito temporal das páginas, vemos que a diferença média entre o âmbito temporal de cada página é de 43 dias. Isto quer dizer que os documentos de partida e chegada discutem assuntos que estão próximos no tempo. No entanto, há outros documentos que têm um âmbito temporal tão diferente que a diferença entre eles pode chegar aos mais de 26 mil dias.

Tabela 6.13: Distribuição dos Documentos por Número de Ligações

	Total de Documentos
Sem ligações	2.569.662
Com 1 ligações	5.376
Com 2 ligações	6.700
Mais de 3 ligações	1.438

Sobre as ligações por documento, a maioria deles não tem qualquer ligação, como já seria esperado, devido ao pequeno número de ligações na colecção.

Mas, pela Tabela 6.13, percebemos que, dos documentos que têm ligações, a maior parte deles têm uma ou duas ligações. Documentos com mais de 3 ligações são pouco mais de 1000.

Estudando a distribuição das ligações por quartil, vemos que é no primeiro quartil que se encontram a maior parte das ligações.

Tabela 6.14: Distribuição das Ligações por Quartil

	Total de Documentos	Percentagem (%)
1º Quartil	2.806	3,97
2º Quartil	60.550	85,57
3º Quartil	5.262	7,44
4º Quartil	2.139	3,02

Comparando estes resultados com os apresentados na Tabela 6.10 sobre a distribuição dos segmentos pelos quartis, vemos que as ligações não se encontram

nos mesmos quartis onde se encontram a maior parte dos segmentos.

6.6 Colecção

Sobre a colecção na sua generalidade, podemos analisar e estudar quais os períodos no tempo que mais vezes foram referenciados nos documentos. Para fazer isto, apenas poderemos analisar os documentos que têm informação temporal. Neste caso, recordando os dados da Tabela 6.1, são pouco mais de 1,9 milhões.

Para identificar esses períodos, foram criados vários intervalos temporais. Assim sendo, de modo a compreender melhor a divisão temporal adoptada e os resultados obtidos, observemos a Tabela 6.15.

Tabela 6.15: Distribuição do tempo do conteúdo dos documentos pela linha do tempo

	Total de Documentos	Percentagem (%)
Antes de 1900	132.384	6,96
Década de 1900	4.746	0,25
Década de 1910	4.567	0,24
Década de 1920	5.549	0,29
Década de 1930	5.817	0,31
Década de 1940	7.308	0,38
Década de 1950	10.492	0,55
Década de 1960	11.709	0,62
Década de 1970	20.150	1,06
Década de 1980	36.085	1,90
Década de 1990	325.563	17,11
Ano 2000	260.350	13,68
Ano 2001	341.649	17,96
Ano 2002	467.760	24,58
Ano 2003	204.074	10,73
Ano 2004	1.031	0,05
Depois de 2004	10.123	0,53

Fazendo uma breve análise à Tabela 6.15, constatamos que o maior número

de referências temporais (quase 70%) dizem respeito ao intervalo de tempo de 2000 a 2003. Este é um resultado expectável uma vez que esta foi uma colecção recolhida durante o ano de 2003.

Embora seja numa percentagem muito reduzida, é possível encontrar documentos que referenciam datas futuras.

Olhando para a Tabela 6.16 vemos a distribuição dos documentos pela linha de tempo mas agora através do estudo da sua data de criação.

Logo à partida reparamos que existem documentos com data de criação posteriores a 2003 e anteriores 1990. Esta colecção foi recolhida em 2003 logo os valores que se encontram depois de 2004 estarão errados. Isto poderá ter acontecido devido a algum problema durante a recolha dos documentos da colecção ou então, talvez a razão mais acertada, porque a data e hora dos servidores em que os documentos estavam alojados estavam erradas.

Tabela 6.16: Distribuição da data de criação dos documentos pela linha de tempo

	Total de Documentos	Percentagem (%)
Década de 1970	3	0,00
Década de 1980	12	0,00
Década de 1990	54.918	5,72
Ano 2000	67.438	7,02
Ano 2001	132.448	13,79
Ano 2002	330.310	34,39
Ano 2003	375.333	39,08
Ano 2004	0	0,00
Depois de 2004	33	0,00

Mas mesmo com esses documentos que estarão com a data de criação erradas, vemos que a maior parte deles está dentro do intervalo de anos correcto e adequado, ou seja, entre 2000 e 2003.

Capítulo 7

Descarga Temporal

Um robô temporal deve percorrer a Web à procura de informação que esteja dentro de um âmbito temporal. Este âmbito temporal é especificado antes do processo de descarga para que as páginas descarregadas possam ser analisadas temporalmente para garantir que elas estão dentro do âmbito temporal imposto.

O robô temporal desenvolvido não é mais que um robô geral com capacidades de análise temporal embebidas em módulos específicos. A análise temporal é feita a cada página resulta de um conjunto de métodos que são aplicados aos documentos Web para obter a informação necessária.

Na Figura 7.1, vemos ilustrada a arquitectura proposta. Esta arquitectura não difere em muito, na generalidade, de uma de um robô geral. A única diferença, como já foi mencionado, está na análise temporal que é preciso fazer a cada página.

O robô apenas pode começar a trabalhar depois das sementes terem sido inseridas. As sementes são URLs pelos quais o robô inicia a sua procura. A operação descrita na Figura 7.1 deduz que o robô já tem as sementes inseridas e que está agora pronto para percorrer a Web à procura de páginas para descarregar.

A Fronteira é onde todos o URLs que devem ser visitados estão. Estes URLs vão sendo adicionados sempre que o robô descarrega páginas que têm URLs que cumprem um conjunto de restrições. É na Fronteira que os URLs, de acordo com

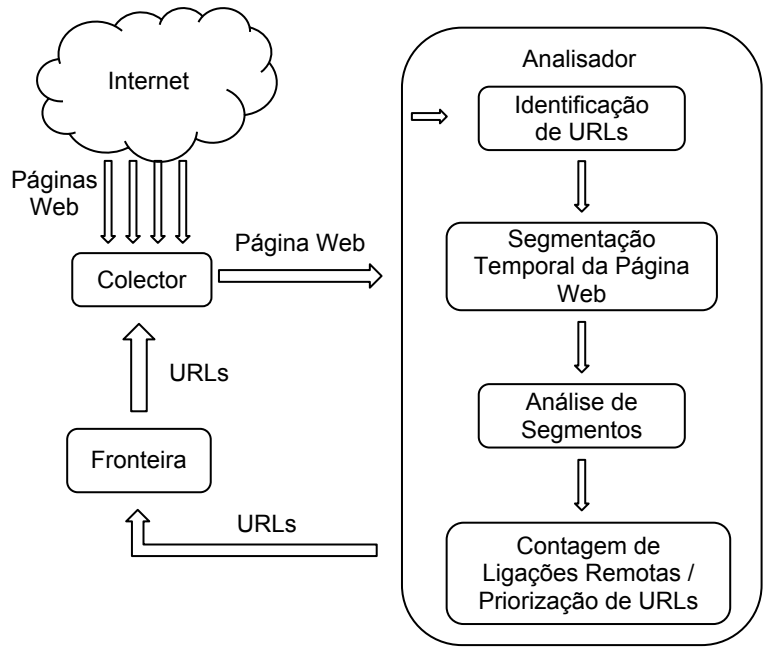


Figura 7.1: Arquitectura do Robô Temporal

a sua prioridade, são descarregados mais cedo se tiverem uma prioridade alta ou mais tarde se tiverem uma prioridade baixa.

Então, o primeiro passo é remover um URL da lista de espera da Fronteira. Esta operação apenas é feita se ainda existirem URLs na lista da Fronteira. Se não existir nenhum URL, o processo de descarga termina. Se ainda existirem URL, então esse URL é passado para o passo seguinte. Como estão implementadas prioridades, o URL com a prioridade mais alta é descarregado primeiro.

O próximo passo é verificar se esse URL deve ou não ser descarregado. Um URL é seguido se pertencer a um domínio pré-definido. Se esta simples condição for satisfeita, a ligação tem luz verde para continuar.

Após receber autorização, as ligações são descarregadas da Web através do Colector. O Colector é o módulo responsável por descarregar as páginas Web correspondentes a cada URL. Este módulo é alimentado com URLs vindos da Fronteira. Depois disto, a página descarregada é passada para o Analisador.

No módulo de Análise, a primeira coisa que é feita é identificar todos os URLS

que existem na página Web e transformá-los para a sua forma canónica para que eles possam ser interpretados por todos os módulos do robô sem terem de fazer mais mudanças. Logo após a identificação dos URLs no documento, o documento é passado ao módulo de análise temporal para ser segmentado.

7.1 Análise Temporal

O modulo de análise temporal é responsável por identificar todos os URLs que estejam dentro de segmentos temporais que cumprem com o âmbito temporal imposto pelo robô. Mas, antes de isso acontecer, os documentos Web devem ser processados pela ferramenta mencionada no capítulo 5. Esta ferramenta transforma os documentos Web num documento XML dividido em segmentos temporais.

Um segmento temporal é um pedaço do documento que foi identificado como pertencendo a um momento específico no tempo ou a um conjunto de momentos. Esta ferramenta permite que pedaços do documento sejam identificadas como pertencendo a um determinado âmbito, enquanto outras pertencem a outro.

No entanto esta é uma ferramenta que tem uma limitação na divisão do documento em segmentos. Neste momento, a ferramenta apenas consegue dividir os períodos temporais por parágrafo e não por frases.

Por exemplo a frase: *Eu no dia 22 de Outubro de 2013 vou embora mas volto no dia 2 de Janeiro de 2014*, será interpretada como sendo um só segmento constituído por dois momentos temporais. O primeiro *22/10/2013* e o segundo *02/01/2014*.

Se conseguise segmentar os documentos por frases, esta frase seria dividida em dois segmentos, cada um contendo apenas uma data.

Por esta razão, os segmentos criados poderão ter mais do que um momento temporal.

Após os segmentos temporais terem sido criados, a análise dos mesmos começa. A análise aos segmentos é feita percorrendo-os a todos, procurando aqueles que estão dentro do âmbito temporal imposto. Um segmento considera-se dentro do

âmbito temporal quando mais de 99% das datas associadas a ele estão dentro do âmbito imposto. Um segmento pode ter mais do que um data associada a ele mas, a sua maioria, apenas vai ter um momento temporal. Isto acontece porque a maior parte das secções de textos em documentos apenas se referem a um momento no tempo.

Os 99% significa que aqueles segmentos que tenham a maior parte das datas dentro do âmbito temporal são escolhidos. Isto faz com que aqueles segmentos que se refiram a mais do que um momento temporal, necessitam que a maior parte das suas datas estejam dentro do âmbito imposto. Esta restrição significa que os segmentos escolhidos são ricos em informação temporal que está dentro do âmbito imposto, daí a escolha dos 99%.

Assim, sempre que isto acontece, um segmento é considerado válido e todos os URLs dentro desse segmento são recolhidos e marcados como válidos. Quando todos os segmentos foram analisados, as ligações que foram marcadas como válidas são mantidas na lista de URLs de saída enquanto as outras são rejeitados.

7.2 Priorização de URLs

Após os URLs terem sido retirados da lista de URLs de saída, a contagem de ligações remotas acontece. Foi criada uma lista de ligações remotas para encontrar caminhos que são mais vezes referenciados, querendo isso dizer que esses caminhos poderão ser mais ricos do que outros tendo em conta a contagem do número dessas ligações. Esta lista consiste numa *String* representativa da ligação e um *Inteiro* representando o número de vezes que a ligação foi referenciada durante o processo de descarga.

A lista de URLs de saída é "unida"com a Lista de Ligações Remotas. Isto resulta numa lista com o total de vezes que um dado URL foi referenciado por outras páginas.

O número de vezes que uma ligação foi referenciada faz com que ela seja mais ou menos importante, de acordo com esse número de vezes. Esta técnica é utilizada para permitir que robô tenha melhores resultados, apanhando o mais

cedo possível as ligações que tenham uma contagem de Ligações Remotas maior dando-lhes uma prioridade mais alta.

Após a lista de ligações remotas ter sido actualizada, a priorização toma lugar na lista que irá ser enviada para a Fronteira. No nosso caso, a prioridade é mais alta quão mais baixo for o seu valor.

A priorização é feita para dar aos URLs com uma contagem de ligações remotas alta o primeiro lugar na fila da Fonteira.

```
(1) Para cada URL na ListaDeUrlsDeSaída {
(2)     saidaURL = ListaDeUrlsDeSaída.procurar(URL);
(3)     contagem = saidaURL.buscarTotal();
(4)     URL.defPrioridade(ParteInteira(100/(contagem+1)));
(5) }
```

Figura 7.2: Pseudo código para a atribuição de prioridades aos URLs

A forma como os URLs são priorizados é melhor entendida olhando para o pseudo código ilustrado na Figura 7.2. Cada URL na lista de URLs de saída (a lista que é entrega na Fronteira) vai ser avaliado e vai-lhe ser atribuído uma prioridade. Essa prioridade é dada de acordo com o número de ligações remotas para esse URL. Quanto maior for a quantidade de ligações remotas, mais pequeno vai ser o valor da prioridade para que esse URL seja visitado à frente de outros.

A linha número 1 é o início do ciclo que vai analisar todos os URLs que estejam na lista de URLs de saída. Para cada um desses URLs, vamos buscar o número de vezes que esse URL foi referenciado. Fazemos isto indo procurar aquele URL à Lista de Ligações Remotas (linha 2). Após termos o objecto correspondente àquele URL, vamos buscar a contagem de referências que esse URL tem (linha3).

Na linha 4, a prioridade para esse URL é atribuída baseando-se na contagem de ligações remotas e, de acordo com isso, o valor atribuído vai ser mais alto ou mais baixo de acordo com o resultado da fórmula. A fórmula devolve um número menor se o URL tem uma contagem de ligações remotas alta e um número maior se o URL tiver uma contagem de ligações remotas baixa. As prioridades vão de 0 a 100, sendo o 0 a prioridade mais alta e 100 a mais baxa.

A implementação do robô temporal utiliza o *Crawler4j* [34] como base.

7.3 Crawler4j

Crawler4j [34] foi o robô genérico utilizado para ser alterado conforme necessário. Este é um robô construído na linguagem Java, simples de usar e de personalizar visto que certas características chave do processo de descarga são herdadas. Então, mudanças que precisem de ser feitas podem-no ser sem haver necessidade de refazer o funcionamento lógico de toda a operação de descarga ou a forma como os vários módulos interagem entre si.

Este robô, é composto por 6 módulos principais: O módulo de descarga, o módulo da fronteira, o módulo do colector, o módulo de análise e o módulo das políticas de delicadeza.

O módulo de descarga controla todos os outros. Este é o responsável por coordenar as várias tarefas [1] para que cada uma siga um URL diferente. Contém as classes que permitem modificar e personalizar o robô. É aqui que se controla se se deve guardar ou limpar a descarga anterior, qual a profundidade da descarga ou qual o número total de páginas a descarregar são todos exemplos de opções que são aqui guardadas e geridas.

A fronteira é o módulo que controla a fila de URLs para o robô descarregar. Cada URL novo é colocado na fila para que ele seja visitado. Este módulo controla a quantidade de URLs, quais aqueles que já foram visitados e quais aqueles que falta visitar. Este módulo trata também os URLs repetidos para que eles não sejam visitados múltiplas vezes.

O colector, tal como o nome indica, é o módulo que esta responsável por ir descarregar os documentos Web relativos aos vários URLs que vêm da fronteira. Depois de os descarregar, ele passa-os para o módulo seguinte para serem analisados.

O analisador é o módulo responsável por analisar as páginas Web. Por de-

[1] Threads na terminologia em inglês.

feito, este módulo apenas analisa as páginas para descobrir ligações. Depois de as descobrir, envia a nova lista criada para a Fronteira.

O módulo das políticas de delicadeza é responsável por fazer cumprir o protocolo *robots.txt*. Este módulo vai buscar os ficheiros *robots.txt* de cada URL e, caso exista, faz com que todo o processo de descarga cumpra com as directivas impostas nesse ficheiro.

Os módulos presentes neste robô têm já um nível de detalhe e uma construção muito boa. Por isso mesmo, este robô é uma solução muito boa para construir um robô temporal.

Capítulo 8

Resultados Experimentais

Para avaliar os resultados produzidos por este robô, iremos utilizar a precisão e a cobertura[1].

A avaliação incidirá apenas sobre páginas do mesmo domínio. Escolhemos a *Wikipédia* escrita em português que está alojada sob o domínio *http://pt.wikipedia.org/*.

Para trazer mais valor ao teste do robô, foram utilizados dois âmbitos temporais baseados em dois marcos importantes da história contemporânea. Um âmbito temporal utilizado diz respeito à Segunda Guerra Mundial (SGM – 1939 até 1945) e o segundo âmbito utilizado diz respeito aos Ataques de 11 de Setembro (9/11 - 2001).

Com base no estudo temporal realizado no Capítulo 6, a utilização de âmbitos temporais apenas com anos é aquela que melhor resultados trará, uma vez que a maior parte das datas encontradas têm o ano presente (Tabela 6.8).

O robô precisaria também de sementes URL. Escolhemos 10 sementes para cada âmbito. As sementes foram recolhidas utilizando o motor de busca Google e podem ser vistas na Tabela 8.1. Escolhemos apenas resultados que estão dentro do domínio definido anteriormente.

Todas as sementes têm o prefixo (omitido da tabela) *http://pt.wikipedia.org/wiki/*.

[1] Recall na terminologia em inglês.

Tabela 8.1: Sementes para os âmbitos temporais da SGM e 9/11.

Âmbito da SGM	**Âmbito do 9/11**
Segunda_Guerra_Mundial	Ataques_de_11_de_setembro_de_2001
Portugal_na_Segunda_Guerra_Mundial	Memorial_%26_Museu_Nacional _do_11_de_Setembro
Causas_da_Segunda_Guerra_Mundial	Assist%C3%AAncia_financeira_ap %C3%B3s_os_ataques_de_11_de_setembro
Brasil_na_Segunda_Guerra_Mundial	Guerra_ao_Terror
Invas%C3%A3o_da_Pol%C3%B4nia	Teorias_conspirat%C3%B3rias_sobre_os_ ataques_de_11_de_setembro_de_2001
Ocupa%C3%A7%C3%A3o_alem %C3%A3_da_Checoslov%C3%A1quia	World_Trade_Center
Partido_Nazista	Comiss%C3%A3o_do_11_de_Setembro
Pacto_Ribbentrop-Molotov	One_World_Trade_Center
Adolf_Hitler	Planejamento_dos_ataques_de _11_de_setembro_de_2001
Hermann_G%C3%B6ring	George_W._Bush

O robô também irá gerar uma lista com URLs representando a ordem pelos quais eles foram descarregados.

Para começar, o robô foi configurado para descarregar 5000 páginas, utilizando as sementes apresentadas na Tabela 8.1, mas sem qualquer restrição. Isto dar-nos-ia parte da nossa colecção.

No fim, teríamos 5000 páginas descarregadas para o âmbito temporal da SGM e 5000 páginas descarregadas para o âmbito do 9/11. Todas as páginas descarregadas passaram pela ferramenta de segmentação temporal.

A seguir, todas as restrições explicadas no capítulo 7 foram ligadas. Assim, o robô iria percorrer a Web guiado através de restrições temporais.

Para a segunda ronda de descargas, as restrições temporais especificadas foram

1939 – 1945 (SGM) e 2001 (9/11). No final, cada processo de descarga iria gerar 5000 ficheiros totalmente segmentados e a lista de URLs com a ordem da descarga, tal como o robô geral.

Para fazer a avaliação é preciso definir quando um documento é ou não considerado válido. Então, um documento é considerado válido quando pelo menos um segmento pode ser colocado dentro da restrição temporal imposta.

Para calcular o total de documentos relevantes na colecção (necessário para calcular a cobertura), considerámos a união do conjunto relevante da descarga temporal com o da descarga geral para a mesma colecção.

8.1 Robô Geral

Passemos a analisar a Figura 8.1 com os resultados da precisão e cobertura do robô geral para ambos os grupos de sementes. Estes resultados servem de base para o nosso robô temporal.

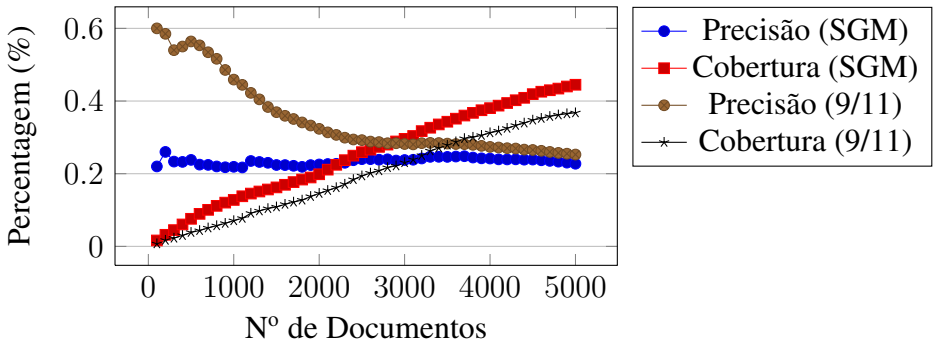


Figura 8.1: Precisão/Cobertura do Robô Geral

A Figura 8.1 mostra que a precisão do robô geral, utilizando as sementes para a SGM, é muito baixa e mantém-se assim durante todo o processo de descarga. As sementes só por si, não foram suficientes para dar ao robô um bom começo pois, como podemos ver, o valor da precisão aos 100 ficheiros é muito baixa.

Agora, olhando para a descarga utilizando as sementes do 9/11, vemos que até aos primeiros 800 ficheiros descarregados a precisão está acima da marca dos

50%. Os resultados razoavelmente bons nas primeiras centenas de ficheiros contrastam com os resultados seguintes pois, logo após esses 800, vemos uma descida bastante acentuada nos resultados da precisão.

No fim, os resultados da precisão para os dois processos de descarga terminam com valores para a precisão a rondar os 25%.

O valor da cobertura é similar em ambas as descargas, terminando à volta dos 40% aos 5000 ficheiros.

8.2 Robô Temporal

Olhemos agora para os resultados da avaliação do nosso robô temporal para os dois conjuntos de documentos (SGM e 9/11) e os seus âmbitos temporais (1939-1945 e 2001).

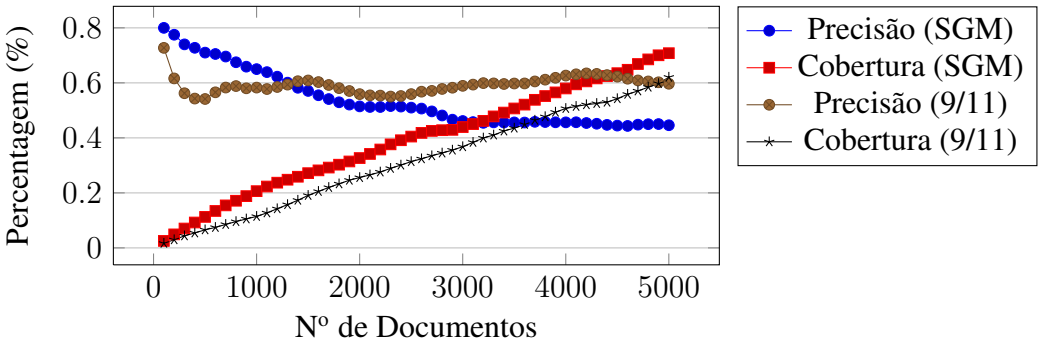


Figura 8.2: Precisão/Cobertura do Robô Temporal

Como pode ser visto na Figura 8.2, a descarga da SGM começa com uma precisão de 80% para os 100 ficheiros. Depois começa a descer de uma forma suave até à marca dos 2000 ficheiros. Nessa mesma marca, o valor da precisão estabiliza na marca dos 50% e mantém-se assim até aos 2600 ficheiros. Após os 2600, o valor da precisão começa a cair e estabiliza por volta dos 45% de precisão.

Se olharmos para a descarga da avaliação do 9/11, não vemos a mesma descida lenta na precisão que anteriormente.

A precisão aos 100 ficheiros está acima dos 70%. A partir daí começa a cair e atinge o valor mais baixo de toda a descarga aos 500 ficheiros com uma precisão de 54%. Depois, sobe para os 60% onde se mantém até ao fim do processo.

A estabilização numa precisão relativamente alta, poderá ser causada pelo facto do intervalo imposto ser de anos recentes (2001) e por isso ser mais fácil encontrar documentos com informação dentro deste âmbito.

Os maus resultados iniciais para a descarga do 9/11 poderão ser causados pelo apertado âmbito temporal imposto. SGM tem 6 anos enquanto 9/11 tem apenas 1.

Analisando a cobertura, vemos que em ambas as recolhas têm curvas semelhantes, tendo a descarga da SGM, uma pequena vantagem sobre a do 9/11.

8.3 Análise Estatística aos Resultados

Com o objectivo de enriquecer e dar mais informação sobre os resultados observados, algumas estatísticas foram recolhidas durante e após o processo de descarga. As estatísticas, apresentadas na Tabela 8.2 caracterizam também a colecção Web utilizada.

Tabela 8.2: Estatísticas das Colecções Descarregadas

	SGM	**9/11**
Tempo de Processamento Total (min)	85	92
Processamento Médio por Página (sec)	1.0	1.1
Tamanho da Colecção (MB)	441.0	430.0
Tamanho Médio das Páginas (KB)	90.0	88.0
Média de Segmentos por Página	82.6	74.9
Segmentos dentro do Âmbito Temporal (%)	32.6	71.8
Média de *chronons* por Página	111.9	96.4

Através da Tabela 8.2, rapidamente vemos que os artigos da *Wikipedia* são muito mais ricos em informação temporal do que a Web normal. Percebemos isso comparando estes resultados com aqueles obtidos no capítulo 6.

Na secção 4.2 do capítulo 4, foi dito que a *Wikipedia* tinha uma média de 80 expressões temporais por página. Os nossos resultados apontam para valores da mesma ordem de grandeza. A quantidade média de expressões para a descarga da SGM tem quase 112 marcas temporais enquanto a descarga do 9/11 tem 96.

Com a análise aos segmentos percebemos que, embora a descarga do 9/11 tenha tido mais segmentos dentro do âmbito temporal, isso não significou melhores resultados de precisão ou cobertura. A quantidade alta de segmentos significa que as páginas dentro do âmbito são mais ricas em informação sobre esse intervalo de tempo.

Capítulo 9

Conclusões

Já vimos os motivos pelos quais a descarga Web é um trabalho bastante complicado. A escalabilidade da Web é o maior problema que os robôs têm de enfrentar. Por isso mesmo, tentar percorrer a Web completa não é solução.

A utilização de restrições é, por isso, uma das melhores formas de lidar com esse problema pois permite que a descarga Web seja guiada e devolva apenas aqueles resultados que interessam aos utilizadores.

Através das interrogações feitas pelos utilizadores, é possível perceber aquilo pelo qual eles mais se interessam. Por isso, são colocadas restrições para procurar informação que os utilizadores querem ver. Essa procura não é feita analisando a Web completa, mas apenas na parte que cumprir a restrição imposta.

Como foi visto com a comparação de robôs com e sem restrições, as restrições guiam a descarga por zonas mais interessantes e mais relevantes. Tudo isto leva a que haja uma maior eficiência na procura de informação e que esta seja de mais alta qualidade.

A utilização das restrições permite que o processo de descarga possa ser descentralizado. A paralelização do processo de descarga veio permitir que múltiplos processos de descarga operem simultaneamente cooperando entre si, para um objectivo comum.

Combinar a paralelização com as restrições, torna as descargas de páginas

Web mais abrangentes, mais rápidas e mais eficientes.

As restrições podem também ser utilizadas de forma combinada. A utilização de várias restrições em simultâneo leva a que a informação recolhida seja mais refinada do que aquela que apenas utiliza uma restrição. Por isso, a informação recolhida é de mais qualidade.

A utilização da dimensão temporal como restrição, é outra forma de limitar a descarga das páginas Web. Utilizando o tempo para restringir a descarga das páginas, é uma forma de trazer mais valor e qualidade às páginas descarregadas.

A criação de um robô com capacidade de descarga temporal era o objectivo deste projecto.

Para isso, foi proposta uma arquitectura para um robô de descarga temporal. Esta arquitectura é composta por um módulo especial que analisa temporalmente as páginas descarregadas e segue as ligações dentro dessas páginas, desde que estejam dentro de segmentos temporais que cumpram a restrição temporal imposta.

A utilização da dimensão tempo como forma de percorrer a Web mostrou-se um conceito promissor. Os bons resultados para um método simples como aquele que foi apresentado, prova que a descarga temporal é viável e que tem muito para evoluir.

No fim de todo, foi um projecto bem sucedido. Foi criada e posta em prática uma arquitecura que se revelou funcionar bem com a dimensão temporal. Num tópico em que não existe nenhuma arquitectura deste tipo, esta foi a primeira proposta de uma.

Daqui para a frente, espera-se que este trabalho permita ajudar na construção de um robô temporal mais robusto e distribuído.

Este conceito, embora promissor, ainda tem algumas arestas a limar.

Antes de mais, é preciso perceber como este robô se comporta. Só mudando as variáveis de funcionamento é que é possível estudar o comportamento deste robô. A quantidade das sementes URL ou o âmbito temporal, são dois exemplos de variáveis que podem ser modificadas para testar o comportamento do robô.

Outro aspecto que deverá ser afinado será o método de avaliação de uma

página. O método que foi aplicado é muito simples e, como qualquer outro, está propenso a erros.

Aliando o tempo ao tópico, as decisões de avaliação estariam relacionadas com o tempo mas também com o tópico discutido em todos os documentos já descarregados e classificados como relevantes.

Desta forma aumenta-se a fiabilidade da avaliação de um dado documento.

São necessárias experiências mais exaustivas variando alguns parâmetros como o âmbito temporal e o conjunto de sementes URL para perceber em mais pormenor como se comporta este robô.

Detectar mais facilmente ligações importantes para o âmbito é algo que trará muitos benefícios ao processo de descarga, tornando-o mais eficiente, produzindo melhores resultados mais depressa. A utilização de modelos probabilísticos, a utilização de outras métricas para medir a importância das páginas são tudo formas que permitem optimizar o processo de descarga melhorando assim os resultados produzidos.

A utilização de um robô temporal juntamente com a segmentação temporal em publicações de redes sociais ou artigos de blogs, poderá produzir resultados interessantes. Toda a informação destes obtida poderá ser dividida temporalmente e analisada de forma nunca antes vista.

Existem também outras direcções de investigação que poderão ser tomadas. A primeira é a introdução da dimensão tempo em robôs focados e de âmbito geográfico. A utilização do âmbito geográfico serve para paralelizar o processo de descarga baseando-se nas restrições temporais.

Bibliografia

[1] Gautam Pant, Padmini Srinivasan, and Filippo Menczer. Crawling the web. *Web Dynamics*, pages 153–178, 2004.

[2] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. *VLDB*, pages 200–209, 2000.

[3] Olga Craveiro, Joaquim Macedo, and Henrique Madeira. Use of co-occurrences for temporal expressions annotation. *SPIRE*, pages 156–164, 2009.

[4] Olga Craveiro, Joaquim Macedo, and Henrique Madeira. Leveraging temporal expressions for segmented-based information retrieval. *ISDA'2010*, page 754–759, 2010.

[5] Ayoub Mohamed H. Elyasir and Kalaiarasi Sonai Muthu Anbananthen. Focused web crawler. *IPCSIT*, 45:149–153, 2012.

[6] José Exposto, Joaquim Macedo, and António Pina. Geographical partition for distributed web crawling. *GIR*, 2005.

[7] Jannik Strötgen, Omar Alonso, and Michael Gertz. Identification of top relevant temporal expressions in documents. *TempWeb*, pages 33–40, 2012.

[8] Olga Craveiro, Joaquim Macedo, and Henrique Madeira. It is the time for portuguese texts! *PROPOR'2012*, page 106–112, 2012.

[9] Pedro Pereira, Olga Craveiro, Joaquim Macedo, and Henrique Madeira. Temporal web crawling. *6th Information Retrieval Facility Conference,*

Chipre, 2013. Submetido na 6ª Conferência Information Retrieval Facility 2013.

[10] Pedro Pereira, Olga Craveiro, Joaquim Macedo, and Henrique Madeira. Temporal analysis of a portuguese web collection. 2013. To be submitted.

[11] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through url ordering. *WWW*, 1998.

[12] Marilena Oita and Pierre Senellart. Deriving dynamics of web pages: A survey. *TempWeb*, pages 25–32, 2011.

[13] Junghoo Cho and Hector Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Trans. Database Syst.*, 28:390–426, 2003.

[14] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology*, 4:378–419, 2004.

[15] Junghoo Cho and Hector Garcia-Molina. Parallel crawlers. *WWW*, pages 124–135, 2002.

[16] Google. We knew the web was big... *Blog Oficial da Google*, http://googleblog.blogspot.pt/2008/07/we-knew-web-was-big.html. Acedido a 3 de Janeiro de 2013, 2008.

[17] Sameendra Samarawickrama and Lakshman Jayaratne. A survey of focused web crawling approaches. *Journal of Information Organization*, 2:1–9, 2012.

[18] Orkut Buyukkokten, Junghoo Cho, Hector Garcia-Molina, Luis Gravano, and Narayanan Shivakumar. Exploiting geographical location information of web pages. 1999.

[19] Ricardo Campos, Gaël Dias, Alípio Mário Jorge, and Célia Nunes. Enriching temporal query understanding through date identification: How to tag implicit temporal queries? *TempWeb*, pages 41–48, 2012.

[20] Ricardo Campos, Gaël Dias, and Alípio Mário Jorge. Using web snippets and query-logs to measure implicit temporal intents in queries. *SIGIR*, 2011.

[21] Erdal Kuzey and Gerhard Weikum. Extraction of temporal facts and events from wikipedia. *TempWeb*, 1:25–32, 2012.

[22] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. Temporal information retrieval: Challenges and opportunities. *TempWeb*, 2011.

[23] Xin Jin, Scott Spangler, Rui Ma, and Jiawei Han. Topic initiator detection on the world wide web. *WWW*, pages 481–490, 2010.

[24] Keishi Tajima Masahiro Inoue. Noise robust detection of the emergence and spread of topics on the web. *TempWeb*, pages 9–16, 2012.

[25] Wordpress Website: http://wordpress.org/.

[26] Joomla Website: http://www.joomla.org/.

[27] Yaniv Bernstein and Justin Zobel. A scalable system for identifying co-derivative documents. *SPIRE*, page 55–67, 2004.

[28] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. Using neighbors to date web documents. *WIDM*, page 129–136, 2007.

[29] Hagège, J. C., Baptista, and N. Mamede. Apêndice b: Proposta de anotação e normalização de expressões temporais da categoria tempo para o harem ii. *Mota, C., Santos, D. (eds.) Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca*, 2008.

[30] Frank Schilder and Christopher Habel. From temporal expressions to temporal information: Semantic tagging of news messages. *Proc. of ACL'01 workshop on temporal and spatial information processing*, pages 65–72, 2001.

[31] David Ahn and Maarten de Rijke Sisay Fissaha Adafre. Extracting temporal information from open domain text: A comparative exploration. *DIR 2005: 5th Dutch-Belgian Information Retrieval Workshop*, pages 3–10, 2005.

[32] Omar Alonso, Ricardo Baeza-Yates, and Michael Gertz. Clustering and exploring search results using timeline constructions. *CIKM '09*, pages 97–106, 2009.

[33] Linguateca. Wpt 03 - artigo sobre a colecção de dados. http://www.linguateca.pt/wpt03/. Acedido a 3 de Fevereiro de 2013.

[34] Crawler4j website: https://code.google.com/p/crawler4j/. Visitado pela última vez em 27 de Junho de 2013.