

# Ranking no Motor de Busca TUMBA

Miguel Costa, Mário J. Silva

Faculdade de Ciências da Universidade de Lisboa, 1700 Lisboa

{mcosta@xldb.fc.ul.pt, mjs@di.fc.ul.pt}

**Palavras chave:** Ranking, PageRank, Motor de Busca, TUMBA.

## Resumo

Apresentamos os algoritmos e mecanismos utilizados num motor de busca por nós desenvolvido, o TUMBA, para ranking (classificação ordenada) de documentos. Estes dividem-se em três tipos, consoante o tipo de análise que realizam: conteúdos, estrutura de links da WWW e interação com o utilizador. Evidenciamos os problemas que levam os algoritmos de ranking a não obter os melhores resultados, e apresentamos algumas propostas de resoluções desses problemas. Este trabalho inclui ainda a descrição das técnicas por nós encontradas para conjugação dos resultados dos vários tipos de algoritmos utilizados.

## 1 Introdução

Os motores de busca na WWW são cada vez mais uma ferramenta imprescindível na pesquisa de informação. Graças a estas ferramentas temos acesso a informação em toda a Web de forma rápida e eficiente. Essa eficiência deve-se em grande parte à qualidade dos algoritmos de ranking hoje usados nos motores globais de pesquisa. Contudo, a WWW continua a expandir-se rapidamente, havendo proliferação de informação sem qualidade e estrutura, misturada nos documentos e interligada sem qualquer relação semântica. Alguns documentos são muito focados num tema, enquanto outros são muito vagos. Existe, por outro lado, uma cada vez maior heterogeneidade dos formatos destes. Para piorar, geralmente os utilizadores dos motores de busca tendem a inserir poucos termos, de um a três em geral, sem dar muita atenção à sua formulação [1]. Muitas vezes os próprios utilizadores apenas têm uma ideia vaga da sua necessidade de informação, inserindo termos ambíguos, originando a produção de resultados totalmente diferentes dos esperados. Por fim, os resultados são sempre subjectivos: o que pode ser relevante para um utilizador pode não ser para outro. Mas, todos esperam que os documentos mais relevantes apareçam em primeiro lugar, ou quando muito, nas primeiras dez ou vinte posições apresentadas. Em geral, os utilizadores dos motores de busca não costumam ver mais que a primeira página de resultados.

Estando nós a desenvolver um motor de busca centrado na Web Portuguesa, o TUMBA [2], ofereceu-se-nos a oportunidade de estudar os vários algoritmos de ranking publicados e propor um mecanismo que permitisse a conjugação de alguns destes, de forma a poder oferecer resultados óptimos nas pesquisas realizadas. Os algoritmos estudados dividem-se em três tipos: os baseados na

análise do conteúdo dos documentos, os baseados na análise da estrutura de links da WWW e os que utilizam a informação do registo das interacções do utilizador com o motor de busca.

A nossa avaliação dos algoritmos implementados no TUMBA permitiu-nos concluir que o ranking apresenta melhores resultados quando conjuga diferentes tipos de algoritmos. Observámos que muitas fragilidades de um tipo são eliminadas pelos pontos fortes de outros tipos, complementando-se.

O presente documento apresenta-se estruturado da seguinte forma: na secção 2, detalhamos os algoritmos e mecanismos utilizados e a obtenção do ranking global; na secção 3 fazemos uma apresentação resumida do algoritmo PageRank; na secção 4, detalhamos a nossa implementação. Os resultados obtidos são descritos na secção 5 e as nossas conclusões e perspectivas sobre trabalho futuro na secção 6.

## 2 Algoritmos de ranking

Os algoritmos propostos na literatura para ranking em motores de busca podem ser classificados em três tipos, consoante a informação que analisam:

**conteúdo:** analisam toda a informação que se pode extrair do próprio documento, como por exemplo títulos e texto. Procura-se em geral a conjugação entre os termos das pesquisas e os termos dos conteúdos.

**estrutura de links:** analisam a interligação entre as diversas páginas na WWW, que pode ser vista como um grafo, sendo as páginas os nós e os links as suas arestas. Esta estrutura pode servir para inferir estimativas da importância relativa das páginas.

**interacção:** analisam a informação obtida a partir do registo das interacções dos utilizadores com o motor de busca.

### 2.1 Algoritmos de análise de conteúdo

Os motores de busca da Web inicialmente desenvolvidos usavam exclusivamente algoritmos de recuperação de informação clássicos, baseados apenas no texto contido nos documentos. Estes devolviam os documentos ordenados pelos valores duma função de ranking que contabilizava o número de ocorrências dos termos da pesquisa no documento. Destacam-se entre estes os que se baseiam no modelo vectorial proposto por Salton [3].

Os algoritmos deste tipo servem também para restringir o universo de pesquisa, filtrando os documentos que não possuem os termos da pesquisa. Se um documento contiver os termos da pesquisa, existe uma boa probabilidade de o documento interessar ao utilizador. Inversamente, se não os contiver o seu interesse será reduzido. Por isso, este é o primeiro passo na obtenção do ranking em tempo de execução. Um motor de busca para além de eficiente deve ser rápido, e o cálculo do ranking para os documentos é um processo que demora algum tempo. Logo, quanto menor for o número de documentos sobre os quais se computa o ranking, mais rápida será a resposta.

No entanto, os documentos da Web para além do seu texto contêm informação adicional relevante para o ranking. Por exemplo, as palavras em negrito e do título podem constituir uma melhor

caracterização do conteúdo de um documento do que as outras palavras. Consideremos um documento com a expressão *Faculdade de Ciências* no título e várias vezes em negrito ao longo do texto. Quando é feita uma pesquisa *Faculdade de Ciências*, este documento aparenta ter uma importância maior para o utilizador que um documento que nele contenha apenas uma vez esta expressão em texto normal. Logo, se utilizássemos apenas esta informação, esta importância seria reflectida num ranking mais elevado.

A selecção de documentos baseada no seu conteúdo apresenta no entanto alguns problemas clássicos [4]:

**Sinónimos:** os documentos podem conter apenas termos sinónimos aos termos da pesquisa, sendo ignorados quando o utilizador não os indica explicitamente.

**Ambiguidade:** dado um termo de pesquisa, este pode ter vários significados.

**Estilos do autor:** documentos sobre o mesmo tópico podem estar escritos com vocabulário e figuras de estilo diferentes.

O problema dos sinónimos é fácil de contornar, recorrendo a um thesaurus, ie, guardando num dicionário os sinónimos de cada termo. Na pesquisa procuram-se então para além dos termos inseridos, os seus sinónimos. Os outros problemas já requerem uma análise mais complexa ao nível semântico, o que dificulta em muita a tarefa dos motores de busca.

O ranking dos resultados das pesquisas com base nos conteúdos sofre, quando utilizado na Internet, da limitação de ser fortemente vulnerável à colocação deliberada de termos em certas páginas, com o único fim de aumentar o seu ranking nos motores de busca. Existem inclusive empresas que fazem desta actividade negócio, estudando os algoritmos e heurísticas dos motores de busca de forma a determinar palavras a colocar nos documentos que os façam aparecer nos primeiros lugares das respostas para determinadas pesquisas. Muitas vezes acontece mesmo que os termos assim colocados não possuem nenhum relacionamento com o contexto do documento apresentado.

Por último, este tipo de algoritmos não avalia documentos que não contenham texto algum, como por exemplo algumas páginas de entrada de sites na WWW constituídas apenas por imagens. Uma solução reside na utilização dos textos das âncoras de outros documentos que referenciem o respectivo documento. Geralmente, estas contêm melhores descrições do conteúdo de um documento que o próprio documento, por serem sumários escritos por terceiras partes. Em [5] para além de se usarem os termos das âncoras, utiliza-se também termos perto destas, pesados mediante a sua distância medida em número de termos, destes à âncora do documento.

## 2.2 Algoritmos de análise de estrutura

A estrutura de links da WWW fornece informação acerca da importância dos documentos. Podemos ver a WWW como um enorme grafo em que as páginas são os nós e os links entre elas as arestas. Deste grafo podemos extrair informação com base na análise da sua conectividade. A assunção que se faz, análoga à das análises bibliométricas, é que a existência de um link de um documento para outro, quando estes têm autores diferentes, indica que o primeiro dá importância à informação do documento referenciado, e que existe uma boa probabilidade de a informação estar relacionada.

Pode-se retirar mais informação baseado noutras assunções: por exemplo, se o documento A é referenciado por mais documentos que o documento B, então A tem um peso maior que B; se A for referenciado por um documento de boa qualidade, tem maior peso do que se for referenciado por um documento de má qualidade. Kleinberg [6] propôs uma ideia simples para computação do valor desta informação: para cada documento existem dois valores, o valor de *autoridade* (um documento com elevada autoridade contém muita informação relevante sobre um tópico) e o de *hub* (reflecte o número de autoridades referenciadas por um documento). Cada documento tem um peso de hub tanto maior quanto mais documentos referenciar e tem um peso de autoridade tanto maior quanto o número de links de documentos a referenciá-lo. Com base nesta ideia, Kleinberg criou um algoritmo de análise de conectividade entre os documentos que obteve bons resultados e inspirou os motores de busca da última geração. Dele têm surgido muitas variações que têm melhorado os seus resultados e solucionado alguns dos problemas por ele trazidos [1],[7]. Um algoritmo também baseado na análise da estrutura da WWW que demonstra obter bons resultados é o PageRank, utilizado no motor de busca Google [8].

Apesar de apresentar bons resultados, este tipo de algoritmos tem alguns problemas no processamento de:

**Links automáticos:** existem links que não conferem qualquer tipo de autoridade à informação do documento referenciado, como links de navegação dentro do site, links de *banners* publicitários e todo o tipo de referencias em que os links são criados por ferramentas. Estes, por não serem editados por humanos, não conferem a mesma autoridade aos documentos referenciados. Todos estes links de navegação ou gerados automaticamente devem ser portanto ignorados, mas a tarefa é muito mais difícil do que pode parecer. Para identificar os links de navegação dentro do site, surge o problema de um site poder estar alojado em mais de um servidor, tornando difícil detectar as suas fronteiras. Os links de publicidade e respeitantes a acordos entre sites são ainda mais difíceis de detectar, já que não há nenhuma estrutura pré-definida que permita descobri-los.

**Mistura de informação nos documentos:** existem muitos documentos com vários tópicos e informação misturada, embora apenas uma parte dela seja relevante para a pesquisa. Este tipo de algoritmos trata o documento atómicamente e prejudica os resultados da pesquisa. Em [9] resolve-se o problema dividindo os documentos em árvores DOM (Document Object Models) [10] e utilizando em vez do documento total para cálculo do algoritmo apenas a parte relevante à pesquisa.

**Links sem sentido semântico:** existem muitos documentos e sites ligados entre si sem nenhuma relação entre a informação. Estas ligações são criadas com o único intuito de tentar enganar os motores de busca de modo a que os seu sites apareçam nos primeiros lugares nas respostas a pesquisas específicas. Pensou-se que este tipo de algoritmos baseados na estrutura de links iria acabar com o spamming dos motores de busca baseados em conteúdo, mas tal não veio a acontecer. O problema subsiste, embora já não seja hoje tão linear defraudar um motor de busca que utilize algoritmos baseados na estrutura de links da WWW.

**Generalização:** algoritmos como o HITS [6], desenvolvido por Kleinberg, tendem a generalizar os resultados das pesquisas. Por exemplo, insere-se o termo *informática* e ele retorna os docu-



para ajuste da função de conjugação dos diferentes tipos. Apresentamos duas fórmulas propostas para essa conjugação, no caso de estarmos em presença de três tipos de algoritmos:

$$ranking(x) = a * TIPO\_A(x) + b * TIPO\_B(x) + c * TIPO\_C(x) \quad (1)$$

$$ranking(x) = 1 - (1 - TIPO\_A) * (1 - TIPO\_B) * (1 - TIPO\_C) \quad (2)$$

A primeira é uma função aritmética simples em que os pesos  $a$ ,  $b$  e  $c$  são dados conforme a influência dos resultados de cada tipo de algoritmos no ranking global. Cada algoritmo pode ter também um peso associado. Em [12] é apresentada uma fórmula semelhante. Nesta em vez de nela se fazer a conjugação de diferentes tipos de algoritmos de ranking, faz-se a conjugação de diferentes algoritmos todos do mesmo tipo.

A segunda fórmula, proposta em [13], é uma função geométrica que apresenta características próprias diferentes da primeira. Nesta, se um tipo de algoritmos apresentar o ranking máximo, ou seja 1, o valor total da função será também o máximo, mesmo que todos os outros tipos de algoritmos apresentem o valor mínimo, ou seja 0. Por outro lado, se um tipo de algoritmo apresentar o valor mínimo, este valor é anulado na função, contando apenas os valores dos outros tipos. Na sua globalidade esta fórmula tende a elevar o valor total.

### 3 Algoritmo PageRank

O algoritmo PageRank [11], que analisa da estrutura de links da Web, foi desenvolvido para o motor de busca Google. Este foi também utilizado e implementado por nós no componente de ranking do TUMBA, pelo que faremos aqui uma síntese do seu funcionamento.

Conceptualmente o PageRank de uma página representa a probabilidade de uma pessoa que navega na Internet vir a visitar essa página. O PageRank de uma página é tanto maior, quanto maior for o número de páginas a referenciem-na, dependendo também este valor do PageRank dessas páginas. Seja  $G(V, A)$  o grafo correspondente à estrutura de links da WWW,  $P_i$  a página  $i$  que corresponde a um vértice do conjunto de vértices  $V$  do grafo  $G$ , e  $A$  todos os links  $(P_i, P_j)$  que correspondem ao conjunto de arestas do grafo  $G$ .  $i$  e  $j$  variam entre 1 e  $N$  (número total de páginas de  $V$ ).

Matematicamente, o PageRank de uma página  $P_j$ ,  $PR(P_j)$ , é o somatório de todas as divisões entre o PageRank das páginas  $P_i$  que referenciam  $P_j$  e o número de links  $C(P_i)$  que saem dessas páginas:

$$PR(P_j) = (1 - d) + d * \sum_{P_i=1}^N \left( \frac{PR(P_i)}{C(P_i)} \right) \quad \forall (P_i, P_j) \in A, \quad 1 \leq i, j \leq N$$

Na equação acima,  $d$  é a probabilidade de o utilizador inserir outro URL aleatoriamente, após ter chegado a uma página em vez de seguir um dos links. No motor de busca Google [8] este valor é tipicamente 0,85.

A título de exemplo, na Figura 1 podemos ver que o PageRank de B é 10, ou seja a soma do PageRank de A a dividir pelo seu número de links de saída ( $9/1=9$ ), mais o PageRank de C a dividir pelo seu número de links de saída ( $2/2=1$ ).

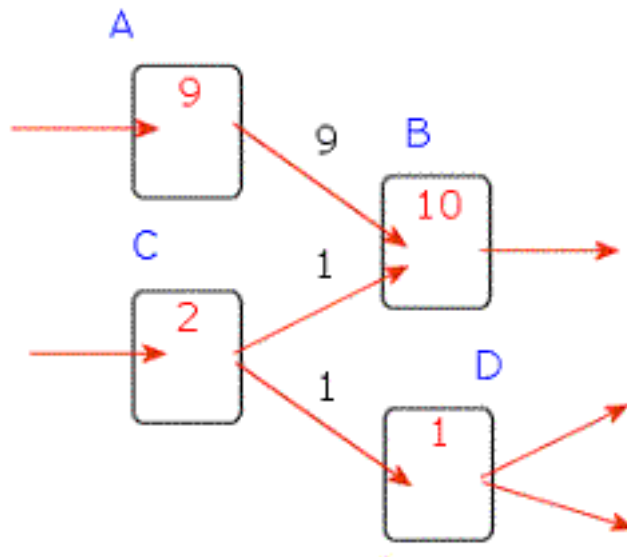


Figura 1: Uma estrutura de páginas Web. Os números dentro das páginas indicam o seu PageRank, enquanto o valor por cima de cada link indica o PageRank que cada página confere à página referenciada

O algoritmo atribui a cada documento um PageRank inicial, que pode ser igual em todos, ou dar maior peso a documentos mais importantes, devendo o somatório desses valores iniciais ser sempre igual a um (somatório das probabilidades de escolher uma página). De seguida, o algoritmo vai iterando de acordo com a fórmula, obtendo o PageRank de cada página até o PageRank de todas as páginas convergir para um erro mínimo.

Para executar o algoritmo necessitamos de ter como estrutura de dados, um vector de origem *Vorigem* de dimensão  $N$  (número de documentos) que vai conter todos os PageRanks iniciais. Há também uma matriz  $M$ ,  $N * N$ , booleana, para registar se a página  $P_i$  aponta para a página  $P_j$  ou não. Temos também um vector de destino  $Vdestino$ , para guardar o resultado da multiplicação de *Vorigem* por  $M$  (ver Figura 2).

$$\begin{matrix}
 & \text{A} & \text{B} & \text{C} & \text{D} & & \\
 \text{[ } & 1/4 & 1/4 & 1/4 & 1/4 & \text{] X} & \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \end{matrix} \begin{matrix} \text{A} & \text{B} & \text{C} & \text{D} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{matrix}$$

Figura 2: O vector de origem *Vorigem* e a matriz  $M$ , usadas na computação do algoritmo PageRank sobre a Web da Figura 1, na primeira iteração.

Abaixo apresentamos o pseudo-código de um algoritmo iterativo para cálculo do PageRank de todas as páginas. O algoritmo itera até o erro ser menor que um dado  $\epsilon$ .

```
for (i:=0;i<N;i++) Vdestino[i] := 1/N;
while (erro>ε) {
  Vorigem:=Vdestino;
  Vdestino:=Vorigem * M;
  Vdestino:=Vdestino * d + (1-d);
  erro:= || Vdestino - Vorigem ||;
}
```

Trata-se portanto de um algoritmo que de uma forma simples permite calcular uma medida de autoridade de cada página com base no número de links que a referenciam. Prova-se matematicamente que este algoritmo é convergente, ie, o valor computado do PageRank tende a estabilizar no PageRank efectivo das páginas.

## 4 Implementação

No TUMBA utilizamos para função global de ranking, uma fórmula que conjuga os diferentes algoritmos utilizados. Partimos de uma fórmula semelhante à fórmula (1) da subsecção 2.4, para a sua optimização, eliminando os pesos de cada tipo de algoritmo, e considerando apenas pesos nos algoritmos para minimizar processamento em multiplicações e divisões. A fórmula do TUMBA, para um documento  $d$  e um conjunto de termos  $T$ , utilizando  $k$  algoritmos é:

$$Ranking(d, T) = \sum_{i=1}^k C_i * A_i(d, T)$$

onde  $A_i$  representa o algoritmo  $i$  e  $C_i$  o seu factor de ponderação na função de ranking, entre 0 e 1. Estes factores de ponderação não devem ter valores muito desnivelados entre si, já que um factor de ponderação em muito superior aos outros faz com que o algoritmo correspondente conduza os resultados da função global de ranking, anulando quase por completo os resultados dos outros algoritmos. No TUMBA foi construído uma ferramenta que guarda toda a informação da interacção do utilizador com o motor de busca, como as pesquisas efectuadas e os documentos escolhidos. A partir desta informação temos primeiro uma noção das pesquisas mais efectuadas e de como os algoritmos e os seus factores influenciam nas respostas. Em segundo, depois de modificarmos a função global de ranking e seus factores, podemos observar o seu efeito nos resultados apresentados ao utilizador.

Os métodos e algoritmos utilizados no TUMBA, assim como os maiores problemas de implementação encontrados são em seguida descritos.

### 4.1 Algoritmos de análise de conteúdo

No motor de busca TUMBA, são inicialmente colecionados num repositório todos os documentos do universo português, pelo seu sistema de recolha [14].

É então efectuado um pré-processamento de todos os documentos. Começamos por criar um índice invertido de todos os documentos usando apenas o seu texto (todo o código e tags HTML são previamente filtrados). Este índice é constituído por uma tabela com todos os termos, de todos os documentos, excepto as palavras mais usuais como artigos e determinantes já que estes aparecem regularmente em todos os documentos. Cada termo contém uma lista com todos os documentos onde se encontra. Deste modo, rapidamente obtemos resposta de quais os documentos que contêm os termos especificados. No TUMBA, este índice invertido é construído num sistema de base de dados ORACLE8, com a extensão Intermedia [15].

A função de TFxIDF utilizada (algoritmo de Salton) é também fornecida pelo ORACLE Intermedia. Esta devolve, para um dado documento, o número de ocorrências no documento do(s) termo(s) indicado(s) na pesquisa,  $f$ , sobre o número de documentos em que aparece esse termo,  $n$ . A fórmula utilizada pelo Intermedia é uma derivação desta feita também por Salton que maximiza os resultados:

$$TF\_IDF(f, N, n) = 3 * f * (1 + \log(\frac{N}{n}))$$

A variável  $N$  representa o número total de documentos indexados. O Intermedia converte o resultado dado pela função acima num número entre 0 (nenhum termo) e 100 (bastantes termos, superior ou igual ao limite para obter pontuação máxima).

A função acima não tira proveito de toda a informação existente nas páginas Web, tratando todos os termos de forma idêntica, independentemente de fazerem parte de títulos ou cabeçalhos de documentos. Para superar esta limitação, extraímos dos documentos informação para ponderar o peso de cada termo nos documentos com base nesta informação. Estes pesos são calculados por um novo algoritmo, sendo estes também utilizados para apuramento do ranking global no TUMBA.

Inicialmente, em off-line, é computado um *peso de apresentação* para cada termo de cada documento que aparece em destaque, em função do tamanho do tipo de letra e formato com que é representado. Para exemplificação da forma como são computados os pesos de apresentação, tomemos um documento HTML com apenas termos em *negrito*, *Heading 1*, *Heading 2*, *Heading 3* e *Heading 4*. Como *Heading 4* apresenta o menor tamanho de entre todos, atribuímos aos termos dentro deste Heading um peso de uma unidade; a *Heading 3*, que apresenta um tamanho um pouco maior, duas unidades; a *Heading 2* três unidades e a *Heading 1* que apresenta o maior tamanho, quatro unidades. Os termos em *negrito* valem também duas unidades já que apresenta tamanho semelhante a *Heading 3*. Cada termo num documento terá um peso de apresentação total correspondente ao valor do somatório de todas estas contribuições. Por exemplo, se num documento o termo *ranking* aparecer uma vez em *negrito* e outra vez a *Heading 1*, o seu valor será  $2 + 4 = 6$  unidades.

Contudo, o peso assim atribuído aos termos é independente do tamanho dos documentos, daí que esse valor deva ser ajustado pelo número total de termos do documento:

$$\text{peso\_apresentação\_ajustado}(t) = \frac{\sum \text{peso\_apresentação}(t)}{\text{total\_termos\_documento}} \quad (3)$$

Devido à enorme quantidade de termos e respectiva informação passível de ser armazenada com estes pesos, e como estes vão ser usados em tempo de execução, há que procurar a melhor relação precisão/tempo de resposta. Para tal torna-se necessário limitar o número de termos a apresentar.

No TUMBA fizemos esta limitação com base numa fórmula simples. Achámos razoável inserir um termo apenas se este aparecer em negrito pelo menos uma vez em cada 50 termos. Como os valores dos pesos dos termos são atribuídos em função da relação do tamanho do tipo de letra e formato em que são visualizados, é o mesmo que dizer que inserimos um termo se aparecer pelo menos num *Heading 1* em cada 100 termos, ou pelo menos num *Heading 4* em cada 25 termos. Só registamos assim na base de dados um termo se este tiver um peso de apresentação ajustado superior ou igual a 0.04 ( $2/50 = 4/100 = 1/25$ ).

Obtemos assim, por cada documento, os termos que mais descrevem cada documento, juntamente com o valor obtido pela fórmula (3) que é utilizado na função de ranking. Quando se faz uma pesquisa, verifica-se para cada documento se possui os termos da pesquisa, e para cada um desses termos adiciona-se na função de ranking uma contribuição relativa a esse termo com base no seu peso total de apresentação ajustado, previamente armazenado.

Para análise dos termos que sobressaem nos documentos, extraímos também, off-line, o título de cada documento. Dispomos para tal de uma outra função de ranking que, com base nos termos dos títulos, devolve um valor que corresponde ao número de termos que foram inseridos na pesquisa e que estão no título, dividido pelo número total de palavras do título.

## 4.2 Algoritmos de análise da estrutura

O algoritmo de PageRank explicado na secção 3, foi dos algoritmos que utilizam a estrutura de links da Web o que implementámos. Para utilização efectiva deste algoritmo, há um conjunto de detalhes de implementação importantes a resolver:

- Muitas páginas apontam para páginas iguais mas com URLs diferentes. Por exemplo, a página A da Figura 1 aponta para a página B e a C para a D. Imaginemos que B e D são a mesma página mas com URLs diferentes. Neste caso o PageRank é repartido pelas duas páginas em vez de ser posto numa só, ficando aquela página com um PageRank menor do que ela tem na realidade. Há também o caso de a mesma página ter *aliases* de URLs no servidor Web. Estes problemas só podem ser resolvidos comparando os conteúdos das páginas e associando a todas um único URL.
- Consideremos uma página P de um site com um PageRank elevado. Todas as páginas apontadas por P no mesmo site, poderão também ter um PageRank elevado se forem referidas por P, o que é provável. Deste modo todas essas páginas de um site com a mesma informação relacionada tenderão a aparecer nas primeiras posições do ranking, o que não é desejável já que um utilizador procura não só precisão mas também diversidade nas fontes de informação. Uma solução consiste em utilizar só os links de páginas que apontem para fora do seu site. Contudo, a definição do que constitui um site é sempre subjectiva, pelo que se torna difícil a sua delimitação. No TUMBA optámos inicialmente por assumir que um site corresponde a um servidor Web.
- No problema do reforço mútuo entre sites, como descrito na subsecção 2.2, se muitas das páginas de um site tiverem um PageRank elevado e referenciarem todas a mesma página de outro site, esta página irá ficar com um PageRank elevado dependente apenas do critério do autor desse site. Este problema costuma acontecer por exemplo quando todas as páginas de um site têm

um link no seu fim referenciando outra página (por exemplo da empresa que concebeu o site). Idealmente todas as páginas de um site ao apontarem para uma página de outro site deveriam ter a mesma influência nesta, como se apenas uma ou poucas a referenciassem. Para conseguir isto foi dado a cada link um peso. Se estiverem  $n$  páginas de um site a apontar para apenas uma página de outro site, o peso de cada link é  $1/n$ , sendo depois o PageRank desse documento multiplicado pelo peso do link antes de ser somado ao documento referido. Ou seja:

$$PageRank(P_j) = \sum (PageRank(P_i) * Peso(Link(P_i, P_j))) \quad \forall (P_i, P_j) \in A$$

Em [6] Kleinberg propõe que, em vez de a influência de vários documentos de um site ao referenciar outro ser igual à influência de apenas um documento, ou seja  $1/n$ , ser antes contabilizada a influência de  $m$  documentos, com  $m$  entre 4 e 8. Assim limita-se as referências a  $m$  documentos. Mas, deste modo os pesos não estão distribuídos equitativamente pelos vários links, já que podem estar a ignorar os links dos documentos que tiverem maior ou menor peso. O melhor é permitir todos os links mas apenas contar uma percentagem deles, sendo esta  $m/n$ . No TUMBA seguimos esta opção, e tomámos o parâmetro  $m = 4$ , ou seja permite-se que até 4 documentos de um site referenciem o mesmo documento de outro site para que o peso do link seja normal, ou seja 1. Caso contrário, contabiliza-se o peso dos links em  $4/n$ , sendo  $n$  o número de links a referenciar o documento.

Outro problema, embora não intrínseco ao algoritmo como os anteriores, e que afecta os resultados deste assim como o ranking global, prende-se com o facto de muitas páginas de entrada de sites, apesar de terem um PageRank elevado não conterem qualquer texto, mas apenas imagens ou animações do programa Macromedia Flash. Nestes casos, se seleccionarmos apenas as páginas com os termos de pesquisa para posterior processamento do seu ranking, as páginas sem os termos nunca serão seleccionadas. A solução por nós encontrada baseia-se na utilização do texto das âncoras dos links que referenciam a página em questão, já que as âncoras normalmente têm descrições mais exactas do conteúdo da página que a própria página [8].

Para que o PageRank possa ser computado eficientemente, há que ter algum cuidado na codificação do algoritmo. Uma implementação simplista terá de fazer imensas multiplicações desnecessárias, já que a matriz  $M$  é booleana. Sendo assim, implementámos o algoritmo PageRank da seguinte forma:

1. Construir uma estrutura que vai conter para cada página, a lista de páginas que referencia e o total das páginas que referencia ( $tot\_ref(P)$ ).
2. Inicializar o vector de PageRank  $PR$  das páginas com  $1/N$  (sendo  $N$  o número total de páginas)
3. Em cada iteração fazer:
  - (a) para cada página  $P$ , atribuir a cada página  $F$  referida por  $P$ ,  
 $PR(F) = PR(F) + (PR(P)/tot\_ref(P))$
  - (b) multiplicar todos os elementos do vector  $PR$  por  $d$  e adicionar-lhe  $(1 - d)$

No cálculo do PageRank executamos um número de iterações igual a  $\log l$ , sendo  $l$  o número de links do universo de documentos. Este valor foi obtido a partir da análise do estudo efectuado em [11]. Embora este estudo tenha sido efectuado para grandes colecções de links, o valor adapta-se bem à nossa colecção, convergindo para um valor exacto ao fim do número de iterações previsto pela expressão acima. Deste modo apenas com somas, calculamos o PageRank de todas as páginas de maneira eficiente e rápida, tal como demonstrado em [16].

De salientar que no passo 2 do algoritmo, podemos inicializar o vector de PageRank  $PR$  de forma diferente. Em vez de atribuir  $1/N$  a todas as páginas, podemos dar um peso maior a páginas mais visitadas e de maior importância. No entanto, o somatório dos valores do vector  $PR$  tem de continuar a ser 1, visto que representa a soma das probabilidades de escolher uma página [17].

## 5 Resultados

Os resultados iniciais obtidos são promissores tanto em tempo de resposta às pesquisas, como também em precisão. Ainda não dispomos de dados estatísticos relativos às pesquisas com a Web correspondente ao universo Português presentemente indexado pelo TUMBA. Por isso, os dados apresentados nesta versão do artigo referem-se a um sub-universo deste de menores dimensões, correspondente às páginas do domínio *.FC.UL.PT* (*Faculdade de Ciências da Universidade de Lisboa*). Este contabiliza um total de 48748 documentos e tem 13137 links de páginas para páginas exteriores ao seu site. Os dados relativos aos algoritmos de ranking aqui apresentados, baseiam-se numa análise de 375 pesquisas realizadas pelos utilizadores do site da Faculdade de Ciências em <http://www.fc.ul.pt>. Interessa por isso salientar que as pesquisas efectuadas, assim como os seus resultados estão restringidos a este domínio, já que os utilizadores são na sua maioria alunos e docentes da Faculdade de Ciências.

O tempo de resposta da função de ranking do TUMBA a um pedido varia entre 0,02 e 5 segundos, num servidor Dell Power Edge 1300, com 2 processadores Pentium III a 500 Mhz. O valor médio do tempo de resposta observado é inferior a 3 segundos.

O algoritmo de PageRank efectuou  $\log n$  (5) iterações em 3.8 segundos para computação do PageRank de cada página, sendo  $n$  já referido anteriormente de 13137 links, o que dá uma média de 0.76 segundos por iteração. Apesar de os valores não serem comparáveis, a equipa do Google refere que calcula 26 milhões de páginas em poucas horas num computador médio [8], isto em 1998.

Em relação à avaliação da precisão do TUMBA, observámos por análise dos dados registados sobre as pesquisas efectuadas, que o documento escolhido pelo utilizador após colocada uma pesquisa está em média na quarta posição. É um resultado positivo embora condicionado por dizer respeito a um domínio relativamente restrito.

A Tabela 1 refere as 10 pesquisas mais efectuadas pelos utilizadores no motor de busca TUMBA. Tal como os autores de outros motores de busca, observámos que as pesquisas efectuadas no TUMBA são pouco específicas e muito abrangentes. Quase todas têm apenas um termo na pesquisa.

A Figura 3 mostra o número de vezes que documentos apresentados na posição de ranking  $X$  foram escolhidos. O número de vezes que as páginas foram escolhidas na primeira posição do ranking, ou seja 50, é um bom indicador da precisão do TUMBA. Pode-se ver que este valor é superior ao das outras posições do ranking. A soma de documentos seleccionados apresentados nas 10 primeiras

Termos	Ocorrências
HORÁRIOS	25
MESTRADOS	10
BIOLOGIA	9
DA	9
BIBLIOTECA	8
CENTRO	8
RESULTADOS	7
FOTOGRAFIA	6
ESPECIAL	6
INFORMÁTICA	6

Tabela 1: pesquisas mais frequentes no TUMBA

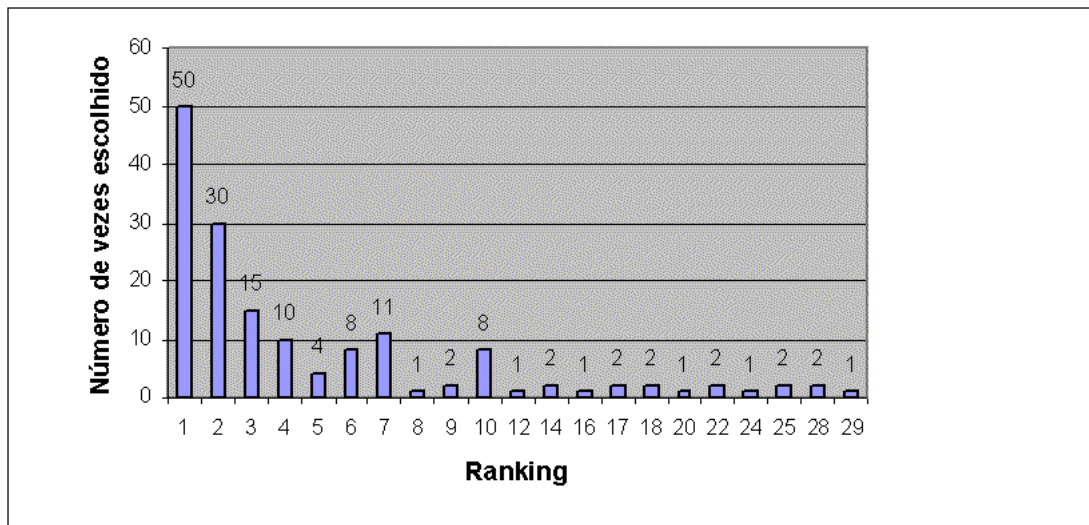


Figura 3: Número de seleções de documentos em função do ranking com que foram apresentados

posições é de 139, ou seja, em 37% das pesquisas, o TUMBA encontrou documentos relevantes na primeira página de resultados.

O algoritmo de PageRank apresenta bons resultados na obtenção da importância da página, e dos algoritmos utilizados, é o que mais contribuiu para a eficiência do TUMBA, já que sem ele observamos um grande decréscimo na precisão. A função de TFxIDF de Salton disponibilizada pelo Oracle Intermedia mostrou bons resultados como esperado, já que o modelo vectorial sempre obteve bons resultados. A função de ranking por análise dos termos dos títulos aumenta também significativamente a qualidade dos resultados, confirmando a nossa intuição que os documentos com os termos de pesquisa contidos no título sugerem que estes são relevantes para o utilizador. Os dados extraídos das páginas Web sobre a visualização do tamanho e formato dos termos, e a sua forma de pesagem eficiente e simples baseado na relatividade dos seus tamanhos, fornecem informação que ajuda em muito no melhoramento da precisão do TUMBA, já que deste modo obtemos para cada documento os termos que mais o descrevem.

A função global de ranking mostrou como a conjunção dos diferentes tipos de ranking fez com que as fragilidades de alguns tipos de algoritmos fossem complementadas pelos pontos fortes dos outros tipos, conseguindo-se assim muito melhores resultados do que se tivéssemos apenas uma função global

sem junção de algoritmos de diferentes tipos de ranking.

## 6 Conclusões e trabalho futuro

Através da implementação de alguns algoritmos e mecanismos de ranking baseados em três tipos diferentes de análise, construímos a função de ranking de documentos do motor de busca TUMBA, que combina os resultados dos diferentes tipos. Estes baseiam-se na avaliação do conteúdo dos documentos, na estrutura de links da WWW e na interação do utilizador com o motor de busca. A sua conjugação origina que muitas fragilidades de um tipo de algoritmos sejam complementadas pelos pontos fortes de outros tipos, obtendo-se assim melhores resultados do que com apenas um tipo.

A literatura existente, apesar de não ser escassa, é muito limitada no tema de conjugação de diferentes tipos de algoritmos de ranking. Foca principalmente na apresentação de novos algoritmos e na comparação de resultados entre algoritmos do mesmo tipo. Foi por isso difícil conseguir uma função global de ranking que conjugasse os diferentes tipos e apresentasse bons resultados.

O pré-processamento dos documentos, como a extracção de informação da visualização do tamanho e formato dos termos, apresentou sérias dificuldades. Foi utilizada uma ferramenta construída por nós para extracção dessa informação nos documentos HTML, baseada na sintaxe publicada para esta linguagem [18]. Mas o número de documentos com sintaxe incorrecta é elevado, obrigando-nos a dispendir muito tempo na adaptação da ferramenta para suportar um número elevado de incorrecções de sintaxe.

Foi também difícil conseguir que o TUMBA apresentasse respostas num reduzido intervalo de tempo, já que a informação a ser processada é muita. Por isso pré-processámos o máximo de informação possível e eliminámos todos os cálculos possíveis na função global de ranking.

A análise inicial dos dados de utilização do TUMBA no domínio .FC.UL.PT da Faculdade de Ciências da Universidade de Lisboa sobre o nosso ranking permite-nos concluir que este apresenta um bom grau de precisão.

Na próxima versão do TUMBA o ranking será melhorado com mais mecanismos e algoritmos de análise de:

**conteúdos:** utilizaremos informação adicional na computação do ranking, como a data de criação do documento, para avaliar o grau de actualidade dos documentos, ou o número links quebrados, ou seja, o número de links que apontam para documentos que já não estão acessíveis. Faremos também a detecção de termos da pesquisa nos URLs de documentos. O texto das âncoras dos documentos pode servir também para análise dos documentos que referenciam. Zhu e Gauch realizaram um estudo sobre o efeito de análise de alguns destes conteúdos para aumentar a precisão na pesquisa e reportaram bons resultados [12].

**estrutura de links:** no algoritmo PageRank, podem-se atribuir maiores valores de PageRank inicial às páginas com maior probabilidade de serem vistas, devido à sua importância e popularidade. Com o nosso conhecimento especializado de Web portuguesa, iremos fazer estes PageRanks iniciais reflectir o padrão de acessos dos utilizadores de Portugal, de forma a que os resultados reflectam as suas preferências.

**interacção:** para além de se utilizar a informação processada para obter dados da precisão do ranking, podemos utilizar também essa informação no próprio ranking para dar maior peso aos documentos escolhidos em pesquisas semelhantes. Podemos, por exemplo, dar maior peso aos documentos mais seleccionados pelos utilizadores para determinadas pesquisas ou pesquisas relacionadas com determinado tema, e menor peso aos que têm um alto ranking e nunca são seleccionados. Uma das funcionalidades que o motor de busca TUMBA poderá oferecer é um refinamento das pesquisas ou uma ajuda à pesquisa. À medida que o utilizador vai inserindo uma pesquisa, o sistema vai devolvendo propostas de pesquisas mais específicas, semelhante ao motor de busca Altavista [19]. Um sistema deste tipo é descrito em [20] e o sistema, no caso de não devolver resultados e detectar palavras com ortografia incorrecta, poderá apresentar propostas alternativas de termos a inserir na pesquisa, como faz por exemplo o motor de busca Google. Pode-se inserir para cada página o número de clicks efectuadas sobre ela nas pesquisas em que apareceu, e utilizar este valor na função de ranking, assim como o tempo médio de visualização desses documentos. Um sistema que utiliza este tipo de informação está descrito em [21].

## Referências

- [1] Krishna Bharat and Monika Rauch Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 104–111, 1998.
- [2] Motor de busca TUMBA. <http://xldb.fc.ul.pt/tumba>, Novembro 2001.
- [3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
- [4] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. In *Proceedings of the 9th International World Wide Web Conference, May 15-19, 2000*.
- [5] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analysing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.
- [6] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [7] Allan Borodin Gareth. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th International World Wide Web Conference, May 1-5, 2001*.
- [8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
- [9] Soumen Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. *Journal of the ACM*, pages 211 – 220, 2001.
- [10] Document object model (DOM). <http://www.w3.org/DOM/>, Novembro 2001.

- [11] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
- [12] Xiaolan Zhu and Susan Gauch. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 288–295, 2000.
- [13] I. Silva, B. Ribeiro-Neto, P. Calado, N. Ziviani, and E. Moura. Link-based and content-based evidential information in a belief network model. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, 2000.
- [14] Daniel Gomes and Mário J. Silva. Tarântula - sistema de recolha de documentos da web. In *4ª Conferência sobre Redes de Computadores*, 2001.
- [15] Oracle. <http://oracle.com>, Novembro 2001.
- [16] Taher H. Haveliwala. Efficient computation of pagerank. Technical report, Stanford University, 1999.
- [17] Dell Zhang and Yisheng Dong. An efficient algorithm to rank web resources. *WWW9 / Computer Networks*, 33(1-6):449–455, 2000.
- [18] Hypertext markup language (HTML). <http://www.w3.org/MarkUp/>, Novembro 2001.
- [19] Motor de busca altavista. <http://www.altavista.com>, Novembro 2001.
- [20] Peter Bruza, Robert McArthur, and Simon Dennis. Interactive internet search: Keyword, directory and query reformulation mechanisms compared. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 280–287, 2000.
- [21] Chen Ding and Chi-Hung Chi. Towards an adaptive and task-specific ranking mechanism in web searching. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.