UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

PEDRO ANTONIO GONZALES SÁNCHEZ

APRENDENDO FUNÇÕES DE RANKING BASEADAS EM BLOCOS USANDO PROGRAMAÇÃO GENÉTICA

MANAUS
2013

PEDRO ANTONIO GONZALES SÁNCHEZ

APRENDENDO FUNÇÕES DE RANKING BASEADAS EM BLOCOS USANDO PROGRAMAÇÃO GENÉTICA

Dissertação apresentada ao Programa de Pós Graduação em Informática do Instituto de Computação da Universidade Federal de Amazonas, como requisito parcial para a obtenção do título de Mestre em Informática, área de Recuperação de Informação.

Orientador: Prof. Dr. David B. Fernandes de Oliveira.

MANAUS
2013

Dedico este trabalho a meu bebê que esta em caminho, a minha amada esposa e aos meus pais.

AGRADECIMENTOS

Primeiro agradeço infinitamente a Deus por tudo. A minha amada esposa Ivone pelo apoio constante, por seu amor, sua companhia e por sempre acreditar que tudo vai dar certo, amo você. A Teresa, minha mãe pela confiança, pela força, porque sempre me ensinou a lutar por meu sonho com dedicação e fazer de mim uma boa pessoa. Ao Julian meu pai pelos conselhos. A Sofia minha sobrinha que me trouxe felicidade neste caminho. A minha irmã e seu esposo pela ajuda quando precisei deles. A toda minha família peruana.

Ao meu orientador Dr. David Braga Fernandes de Oliveira pelos ensinamentos, paciência, constante apoio, excelente Professional e boa pessoa. Aos professores Dr. Marco Antônio Pinheiro de Cristo, Dr. Edleno Silva de Moura e Dra. Eulanda Miranda dos Santos pelos ensinamentos, orientações e grandes exemplos de Professional.

A todos meus amigos do PPGI em especial ao Caio, Joyce, Jhonathan, João, Herbert, Rallyson, Éfren, Bruno, Juan Gabriel, Thiago, Anderson, Leandro, Moysés, Janainny, David, Elda, Polyanna, Maria, Osvaldo, Nilmara, seus conselhos e amizade ficam para sempre no meu coração e pensamento, eles representaram minha família em Brasil.

A Cristian Rossi e André Carvalho pela importante ajuda em tudo momento.

A todos os professores do PPGI. Ao IComp pela oportunidade de fazer o mestrado. Ao CNPq pela ajuda no mestrado na UFAM nestes dois anos.

Finalmente meu infinito agradecimento a Brasil, um pais maravilhoso, que me deu a oportunidade de crescer como Professional e a todas as pessoas que fizeram que eu me sinta como em casa neste lindo pais.

RESUMO

Na atualidade, a Internet é considerada uma poderosa ferramenta de comunicação e informação. Seu impacto na sociedade está aumentando cada vez mais, o que significa que está se tornando indispensável. Neste contexto, sistemas de busca por informação tornam-se cada vez mais importantes. Neste trabalho, propomos um novo método de busca capaz de aprender funções de ranking que exploram a estrutura em bloco das páginas Web, usando programação genética. Diferentemente de trabalhos anteriores, nosso método permite combinar evidências tradicionais em recuperação de informação com evidências derivadas da estrutura das páginas. Para validar o método proposto, utilizamos três coleções reais de páginas (IG, CNN e BLOG). Os resultados experimentais mostram que nossa abordagem é capaz de superar os resultados de um *baseline* que usa informações de blocos sem aprendizagem de máquina, apresentando ganhos de precisão (MAP) de 9,38% na coleção IG, de 7,13% na CNN, e 25,87% na coleção de BLOG. Em relação a nosso segundo *baseline*, que usa programação genética a partir de evidências tradicionais de recuperação de informação, nosso método conseguiu ganhos de 5,25% na coleção IG, 10,37% na CNN e 4,37% na coleção de BLOG.

Palavras-chave: Programação Genética. Estrutura de Bloco das Páginas Web. Funções de Ranking.

ABSTRACT

Today, the Internet is considered a powerful tool of communication and information. Its impact on society is increasing more and more, which means that it is becoming indispensable. In this context information searching systems are becoming increasingly important. In this paper, we propose a new search method capable of learning ranking functions that explore Web pages structure in blocks, using genetic programming. Different from previous works, our method allows combining traditional evidence in information retrieval with evidence derived from the structure of Web pages. To validate the proposed method, we use three real collections of pages (IG, CNN and BLOG). Experimental results show that our approach is able to overcome the results of a baseline of information which uses blocks information without learning machine, presenting precision benefits (MAP) of 9.38% in the IG collection, from 7.13% in CNN, and 25.87% in collection BLOG. Regarding our second baseline, which uses genetic programming out of traditional evidence in information retrieval, our method achieved benefits of 5.25% in the IG collection, 10.37% and 4.37% on CNN in collection BLOG.

Keywords: Genetic Programming. Block Structure of the Web Pages. Ranking Functions.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PG	Programação Genética.
RI	Recuperação de Informação.
TF	Frequência de uma palavra no documento.
ICF	Frequência Inversa de uma classe.

SUMÁRIO

Capítulo 1

Introdução

Recuperação de informação (RI) é uma área da ciência da computação que se dedica ao estudo e ao desenvolvimento de técnicas de organização, indexação, e busca por informação em coleções de itens armazenados digitalmente, tais como páginas Web e artigos online. A área ganhou grande destaque a partir do surgimento e crescimento da Web, onde máquinas de busca online (tais como Google, Yahoo Search e Microsoft Bing) representam um instrumento indispensável de busca e acesso a todo tipo de conteúdo presente nessa enorme e diversificada massa de informação.

Um processo de busca inicia quando um usuário informa sua necessidade de informação, sob a forma de uma consulta, para um sistema de busca. De posse da consulta, o objetivo da máquina de busca é retornar para o usuário um conjunto de documentos ordenados de acordo com a relevância para com a consulta. Os modelos mais populares de ordenação (ou ranking) de documentos são o modelo vetorial [28,29], o BM25 [9,30], e os modelos de linguagem [8,10,31]. Tais modelos adotam diferentes estratégias para representar consultas e documentos, e diferentes fórmulas para se calcular o ranking.

A maioria das técnicas de ranking (incluindo os modelos citados anteriormente) representam as páginas Web como unidades indivisíveis e monolíticas, isto é, trata as diferentes porções de uma página de forma equivalente. Embora esta abordagem seja apropriada para busca em coleções de documentos planos e sem estrutura, ela é muito simplista para coleções Web. Conforme demonstrado em Fernandes et al. [34], as páginas Web podem ser divididas em diferentes segmentos (tais como menus, rodapés,

conteúdo principal, notas de copyright, título, etc.), cada qual com tipos de conteúdo e propósitos específicos. Além disso, em Moura et al. [33], foi demonstrado que os métodos de ranking para a Web podem ser mais efetivos quando consideram as posições (isto é, os blocos) em que os termos das consultas ocorrem dentro das páginas. O propósito deste trabalho é dar seguimento a essas pesquisas sobre o uso da estrutura das páginas em sistemas de recuperação de informação para a Web.

Não obstante os bons resultados já obtidos nesses trabalhos, esta dissertação parte da hipótese que uma melhoria ainda mais significativa na qualidade de ranking pode ser alcançada a partir do uso da estrutura das páginas. O método publicado em de Moura et al. foi derivado de um modelo inicialmente elaborado para documentos não estruturados (modelo BM25), que foi adaptado para que este passasse a representar os documentos como um conjunto de segmentos. Nesta dissertação, pretendemos usar métodos supervisionados de aprendizagem de máquina para desenvolver uma função efetiva de ranking a partir de bases de treino segmentadas, através de programação genética (PG).

Programação Genética (PG) é uma técnica de aprendizagem de máquina inspirada na teoria de evolução de Darwin para encontrar soluções otimizadas para problemas de alto nível de complexidade. A técnica é implementada através de uma simulação de computador em que uma população inicial de indivíduos (no nosso caso, possíveis modelos de ranking que consideram a estrutura das páginas) gerados aleatoriamente precisa evoluir para indivíduos capazes de produzir resultados otimizados para o problema em questão. No contexto desta dissertação de mestrado, cada indivíduo é uma fórmula de ranking que será avaliada de acordo com sua efetividade no cálculo da ordenação dos documentos.

Para compor cada fórmula de ranking, foram consideradas evidências tradicionais de sistemas de recuperação de informação, tais como TF e IDF, bem como evidências derivadas da estrutura das páginas. Exemplos de evidências baseadas nos segmentos que foram exploradas nesta dissertação são o ICF (inverse class frequency, que é o número de vezes que um termo ocorre em uma classe de segmentos) o SPREAD (que representa o número de segmentos de uma página que possui um determinado termo), o AICF (average inverse class frequency, que é a média de ICF de uma classe), e o ASPREAD (average SPREAD, que representa a média de SPREAD de um segmento),

todas propostas em Fernandes et al. [31]. Nessa dissertação de mestrado, trabalhamos sob a suposição de que PG será capaz de aprender características intrínsecas do problema de cálculo de ranking a partir da estruturadas páginas, e então usar esse aprendizado para melhorar a efetividade do ranking.

Para validar o método proposto, utilizamos três coleções reais de páginas (IG, CNN e BLOG). Os resultados experimentais mostram que nossa abordagem é capaz de superar os resultados de nosso *baseline:1* [33] que usa informações de blocos sem aprendizagem de máquina, apresentando ganhos de precisão (MAP) de 9,38% na coleção IG, de 7,13% na CNN, e 25,87% na coleção de BLOG. Em relação a nosso *baseline:2* [30], que usa programação genética a partir de evidências tradicionais de recuperação de informação, nosso método conseguiu ganhos de 5,25% na coleção IG, 10,37% na CNN e 4,37% na coleção de BLOG.

1.1 Organização do Trabalho

Essa dissertação está composta como segue. No Capítulo 2 são apresentados alguns trabalhos relacionados, No Capítulo 3 são introduzidos alguns conceitos básicos de recuperação de informação, programação genética, e de estruturação de páginas Web. Tais conceitos são fundamentais para o entendimento deste trabalho. O Capítulo 4 mostra o uso de técnicas de PG para aprender funções de ranking baseadas nas estruturas em blocos das páginas e nas evidências tradicionais de recuperação de informação. O Capítulo 5 mostra os experimentos realizados e os resultados alcançados. No Capítulo 6 apresentamos uma análise dos resultados obtidos por PG. Finalmente, no Capítulo 7 apresentamos as conclusões e os trabalhos futuros que poderão ser desenvolvidos a partir de nossos resultados.

Capítulo 2

Trabalhos Relacionados

A grande quantidade de informação na Web e, a importância dos sistemas de recuperação de informação têm motivado atualmente inúmeras pesquisas sobre métodos para se aprimorar a qualidade da informação recuperada a partir das consultas dos usuários. Nesse capítulo, apresentamos algumas técnicas de ranking baseadas na estrutura em bloco das páginas, bem como alguns trabalhos que adotaram técnicas de PG para obter boas novas funções de ranking.

Trabalhos que usam técnicas de ranking baseadas na estrutura em bloco das páginas como em Song et al. [14] apresenta-se um método para estimar valores de importância para blocos de páginas Web. Considerando que muitas vezes a importância dos blocos em uma página Web não é equivalente, o método proposto atribui automaticamente valores de importância dos blocos com o objetivo de encontrar funções para descrever as correlações entre os blocos de páginas web e os valores de importância. Primeiro segmentam a página Web, em seguida extraem as características espaciais e de conteúdo, para a construção de um vetor de características para cada bloco. Com base nessas características, algoritmos de aprendizagem como SVM-Ranking e Redes Neuronais Artificiais são usados para treinar um modelo para atribuir importância a diferentes segmentos na página web com bons resultados.

Em Cai et al. [18] mostra-se uma estratégia de ranking que considera os blocos das páginas Web e adota funções de ranking previamente propostas por Callan et al. [23] para bases de documentos segmentados em parágrafos e passagens. Os resultados desse trabalho mostram que o uso da estrutura em blocos por sistemas de recuperação de informação pode resultar em qualidades de ranking superiores aos obtidos quando a estrutura é ignorada.

Em Fernandes et al. [31], se considera o problema de usar a estrutura de blocos das páginas Web com o objetivo de melhorar os resultados de ranking de busca em Web sites. Entre os quatro métodos apresentados nesse trabalho, o que obteve os melhores resultados foi um método combinado que integra segmentação de páginas baseados em visão e propriedades de tamanho fixo das páginas, conseguindo melhorias significativas na qualidade de ranking em comparação com sistemas que não utilizam informação de estrutura das páginas.

Em Moura et al. [33] também se propõe métodos para melhorar os resultados de ranking de sistemas de busca da Web com base em informações disponíveis na estrutura em bloco das páginas. Os autores apresentaram nove funções capazes de estimar os pesos dos blocos baseadas em informações complementares sobre a ocorrência de termos dentro dos blocos, e um método de ranking capaz de usar tais pesos. Os resultados experimentais foram comprovados usando duas linhas de base (baselines). O primeiro foi com um ranking BM25 aplicado a páginas inteiras, e o segundo foi um ranking de um BM25, que leva em conta os melhores blocos. Desta forma os resultados comprovaram a eficácia da técnica adotada em quatro coleções Web usadas (IG, CNN, BLOG e CNET). Esse trabalho é usado como um baseline sem aprendizagem de máquina para o método proposto nesta dissertação.

Em Fernandes et al. [34] é proposto um método de segmentação de páginas Web, que pode ser usado em modelos de ranking baseados em estrutura. O método combina as árvores DOM (Document Object Model) de todas as páginas de um site em uma única estrutura de dados chamada SOM (Site Object Model). Os resultados apresentados nesse trabalho mostram que, o método de segmentação proposto obteve resultados próximos dos obtidos quando se utiliza uma abordagem de segmentação baseada na intervenção manual.

Trabalhos que usam técnicas de PG para obter novas funções de ranking com ganho significativo como em Fan et al. [15], se aplica PG para descobrir funções de ranking. Os autores propõem um método que consegue gerar automaticamente estratégias de ranking para contextos diferentes baseados em PG. Os resultados foram avaliados usando dados em TREC com bons resultados. De forma similar, PG também foi utilizada com sucesso em um trabalho apresentado pelo mesmo autor em [16] para a descoberta de funções de ranking aproveitando a informação estrutural dos documentos HTML e avaliados em TREC.

Em Trotman et al. [25] os autores fazem uso de técnicas de aprendizagem de máquina para estimar valores de importância para blocos em coleções de documentos com a mesma estrutura. Cada bloco desta estrutura compartilhada é representado em um vetor como uma sequência de características em uma simulação de aprendizagem de Algoritmos Genéticos.

Em Lacerda et al. [28] apresenta-se um método para associar anúncios com páginas Web baseado em PG. O método tem como objetivo obter boas funções de ranking que consigam colocar propagandas em páginas web, baseado no conteúdo das páginas para evitar propagandas irrelevantes. Os autores consideraram evidências estatísticas em PG obtidas de uma coleção real de propagandas e uma coleção de páginas da Web de um jornal brasileiro com resultados muito bons para o problema de propaganda direcionada.

Em Jen-Yuan et al. [29] apresenta-se um método para gerar automaticamente com aprendizagem de máquina uma função de ranking chamada RankGP. Para isso, utilizou-se programação genética para gerar tal função de ranking, através da combinação de vários tipos de evidências de recuperação de informação. Para a função de fitness usou-se a métrica de avaliação MAP (*Mean Avarege Precision*). O método proposto pelos autores é avaliado usando os conjuntos de dados de benchmark LETOR. Os resultados obtidos mostram que RankGP supera BM25 e SVM Ranking significativamente.

Em Mosrri et al. [30] é proposto um método de geração de fórmulas de ranking específicas para cada tipo de coleção, através de Programação Genética (PG). O diferencial desse trabalho é que os autores não utilizaram apenas informações estatísticas tradicionais derivadas das coleções de documentos, mas também adotaram fórmulas de eficácia já comprovada, tais como o Okapi BM25 [32] e o TF-IDF, como terminais do processo evolutivo. Os resultados obtidos mostram que o uso de terminais significativos em vez de terminais com simples informação estatística melhora a qualidade do processo de descoberta de funções de ranking com PG. Esse trabalho é usado como um *baseline* com aprendizagem de máquina para o método proposto nesta dissertação.

Na literatura não encontramos pesquisas orientadas na descoberta de funções de ranking utilizando PG, que consigam combinam evidências baseadas em informações de estrutura em blocos das páginas Web com informações tradicionais em recuperação de informação de eficácia já comprovada. Por tal motivo, é importante o aporte de nosso

trabalho na área de recuperação de informação e de fato, este é a principal diferença de nossa abordagem e as demais que foram citados.

Capítulo 3

Conceitos Básicos

Neste capítulo são apresentados os fundamentos teóricos e os conceitos básicos relacionados com o problema a ser tratado neste trabalho. As seções 3.1 e 3.2 apresentam conceitos básicos da área de recuperação de informação. A seção 3.3 apresenta conceitos relacionados com segmentação e estruturação de páginas Web. Finalmente, a seção 3.4 faz uma breve introdução sobre programação genética.

3.1 Recuperação de Informação

Recuperação de informação é a área da computação que lida com a representação, armazenamento, organização e busca de itens de informação [4, 5, 6]. O objetivo dos sistemas de recuperação de informação é recuperar uma lista de documentos (*ranking*) relevantes para as consultas feitas pelos usuários.

A recuperação de informação apareceu como resposta às necessidades dos usuários de bibliotecas tradicionais ou digitais, mas ganhou grande destaque desde o surgimento da Web. Nesse contexto, recuperar informação de grandes bases de dados começou a se tornar muito complexo. Várias investigações têm sido feitas ao longo dos anos, visto que desenvolver mecanismos de busca eficazes para obter informações relevantes para os usuários na Web se tornou um problema extremamente difícil.

Na atualidade, a tarefa de busca é considerada um dos problemas mais desafiadores e interessantes da Ciência da Computação, porque as ferramentas de busca na Web estão reinventando os negócios, a forma como comunicamos e compartilhamos informação. Em outras palavras as ferramentas de busca na Web estão transformando a vida da sociedade moderna.

Em recuperação de informação uma questão fundamental é como ordenar os documentos de uma base de acordo com seus graus de relevância para com as consultas. Tal questão está diretamente relacionada com a função de ranking, que é uma função utilizada pelos motores de busca (*Google, Yahoo Search!, Bing, etc*) para classificar documentos potencialmente relevantes, de modo que eles possam estar nas primeiras posições da lista (*ranking*) que é entregue ao usuário pelo sistema de recuperação de informação após da uma consulta. A função de ranking geralmente faz esse processo através da atribuição de uma pontuação numérica.

Dentre esses modelos de recuperação de informação mais conhecidos, destacam-se o Okapi BM25 [32], o Modelo Vetorial [26], e os modelos baseados em modelos de linguagem (*languages models*) [8, 10, 31]. Para fazer essas ordenações (*ranking*) os métodos tradicionais de recuperação de informação usam apenas um pequeno número de evidências estatísticas (por exemplo, frequência de termos de consultas dentro dos documentos, frequência inversa de termos da consulta na coleção, tamanho do documento, etc).

Figura 3.1 Esquema do Processo de recuperação de Informação.

A Figura 3.1 ilustra uma operação de busca por informação. O processo de busca se inicia a partir da submissão de uma consulta para um sistema de recuperação de informação. O sistema, de posse da consulta retornará para o usuário um conjunto de documentos ordenados de acordo com uma função de ranking. É importante ressaltar que nem todos os documentos retornados são relevantes para o usuário, que deverá percorrer a lista de documentos retornados para obter as informações desejadas.

3.2 Aprendendo a fazer Ranking

Dentre os mecanismos comumente empregados para se chegar a novos métodos de ranking estão as técnicas de aprendizagem de ranking (*learning to raking*). O propósito de tais mecanismos é gerar automaticamente funções de ranking a partir de dados de treino, usando técnicas de aprendizagem de máquina [1, 2, 3, 5, 27]. Como forma de melhor entendimento, consideramos o seguinte:

A ideia geral por trás desses mecanismos é mostrada na Figura 3.2.

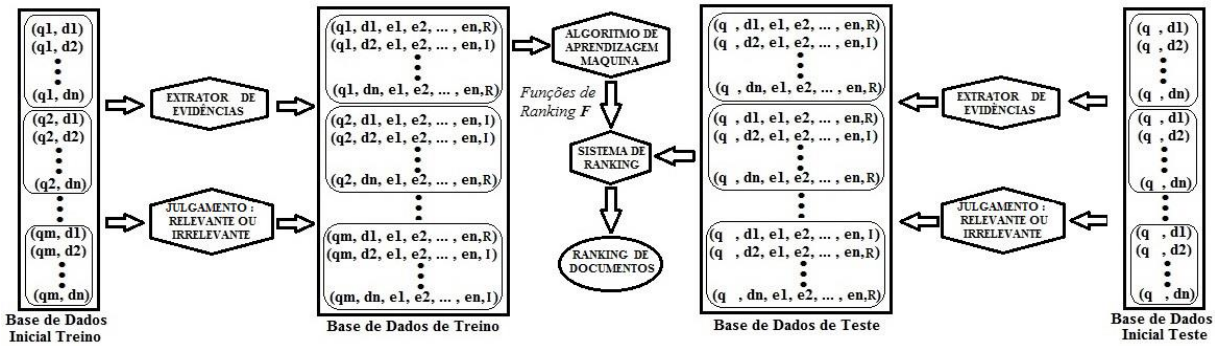

Figura 3.2 Paradigma geral de Aprendizagem de ranking (RI).

O paradigma mostrado na figura 3.2 é utilizado pela maioria dos métodos de aprendizagem de máquina (*learning to ranking*) [29]. Como forma de melhor entendimento, consideramos o seguinte:

Seja uma base de treino conforme mostrado na figura, o processo requer um conjunto de pares ordenados formado por uma consulta q e um documento d como segue: $(q_i\ ,\ d_j)$ a partir das coleções de consulta $Q = \{q_1, q_2, \dots, q_m\}$ e dos documentos $D = \{d_1, d_2, \dots, d_n\}$. Tais pares são divididos em dois grupos de dados, chamados de base de treino e base de teste. A cada par de ambos os grupos é atribuído um rótulo indicando a relação entre a consulta q_i e seu respectivo documento d_j. Esse rótulo, que é atribuído manualmente por um especialista, indica se o documento d_j é relevante *R* ou irrelevante *I* para a consulta q_i .

Para cada instância $(q_i\ ,\ d_j)$ das bases de treino e de teste é aplicado um extrator de características, que irá identificar e extrair evidências sobre a forma como o documento d_j se relaciona com os termos das consultas q_i bem como sobre como os termos da consulta q_i se relacionam com a base de documentos como um todo. Exemplos de evidências comumente usadas em *learning to ranking* são a frequência dos termos da

consulta nos documentos, e o número de documentos que possuem os termos das consultas.

Em seguida, a base de treino é usada por um algoritmo de aprendizagem de máquina, procurando estabelecer que características um documento *d* deve possuir para que este seja relevante para uma consulta *q*. A partir dessas informações as funções de ranking são geradas, que são finalmente avaliadas através da base de teste.

A função de ranking mais efetiva é escolhida através de métricas de avaliação em recuperação de informação (*P@10*, *MAP*, *Bpref10*, entre outros) que buscam avaliar cada método de ranking com base em sua capacidade de retornar documentos relevantes nas primeiras posições do ranking.

3.3 Estrutura em Blocos das Páginas Web

As páginas Web têm componentes visíveis que facilitam seu uso quando interagimos com elas, tais como: a barra de menu, o título, as propagandas, as barras de navegação, entre outros. Cada um desses componentes, que neste trabalho são chamados de blocos, desempenha uma determinada função dentro de suas respectivas páginas.

Em [33], o conceito de bloco é definido como uma região lógica auto-contida dentro de uma página Web que: (i) não está aninhada com qualquer outro segmento e (ii) é representada por um par ordenado *(P,C)*, onde *P* é o *path* (caminho) do bloco, representado pelo caminho entre a raiz da página e a raiz do bloco na árvore DOM (Documento Object Model); e *C* é a porção de texto que o bloco possui.

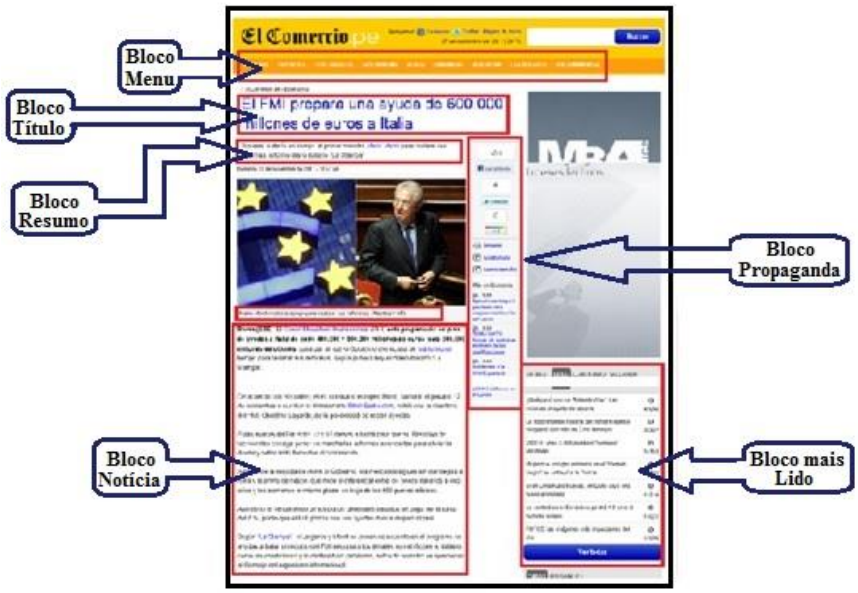

Figura 3.3. Página com marcações dos blocos, extraída do site El Comercio Perú.

A Figura 3.3 apresenta uma página Web dividida por quadros vermelhos que indicam as regiões que são consideradas como blocos.

Os blocos das páginas de um mesmo site podem ser organizados em classes, de acordo com a função que desempenham dentro de suas respectivas páginas. Por exemplo, na Figura 3.4, apresentamos duas páginas Web *ρ1* e *ρ2* que possuem uma estrutura em blocos muito similar. Dizemos que dois ou mais blocos formam uma classe de blocos quando (i) pertencem a páginas distintas de um mesmo site, e (ii) possuem o mesmo *path*, isto é, possuem o mesmo caminho entre a raiz da página e o início dos blocos.

Por exemplo, os blocos Título das páginas dispostas na Figura 3.4 representam blocos de uma mesma classe, visto que eles estão situados na mesma posição dentro de suas páginas (e, portanto possuem o mesmo caminho, ou *path*, na árvore DOM das páginas). Além dos blocos Título, as duas páginas da Figura possuem uma série de blocos em comum. Por exemplo, ambas as páginas possuem um bloco menu, que também pertencem a uma mesma classe por possuírem o mesmo *label*. Observe que blocos de uma mesma classe tendem a exercer uma mesma função dentro de suas respectivas páginas. Neste trabalho, iremos usar estatísticas sobre os termos das consultas nas classes de blocos como evidências para o processo de aprendizagem de máquina.

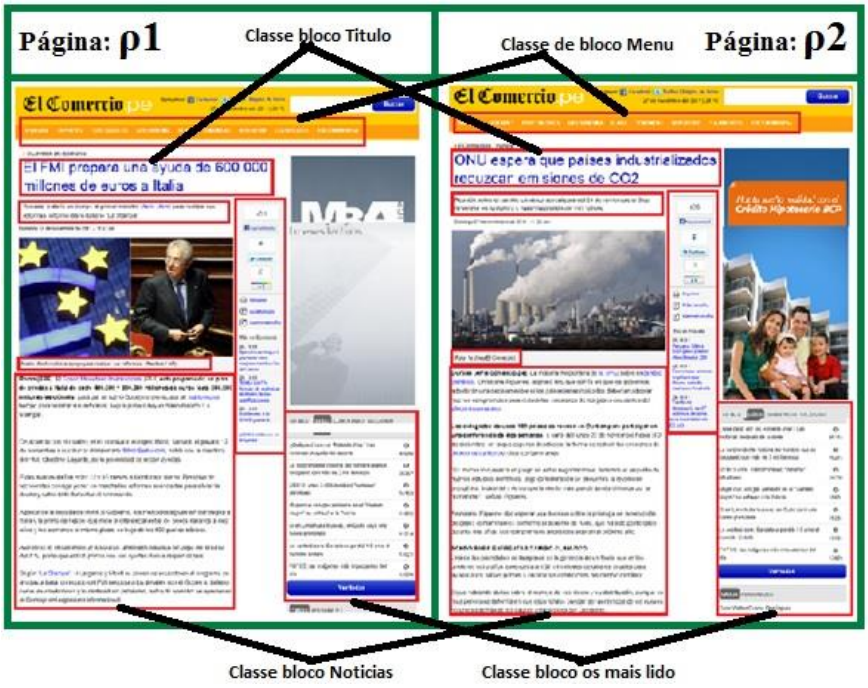

Figura 3.4. Duas páginas "ρ1" e "ρ2" que indicam as classes de blocos.

3.4 Programação Genética (PG)

Programação Genética é uma técnica de aprendizagem indutiva introduzida por Koza em [26] como uma especialização de algoritmos genéticos onde cada indivíduo é um programa de computador. É uma metodologia baseada em algoritmos evolutivos inspirados pela ideia da seleção natural para gerar programas de computador capazes de executar determinadas tarefas. PG é usada para aperfeiçoar uma população de programas de computador de acordo com uma função de aptidão (função de fitness) determinada pela capacidade de um programa para executar uma determinada tarefa computacional.

O processo da PG começa com uma população inicial composta por um conjunto de indivíduos (programas de computador, ou funções matemáticas) criados aleatoriamente [30]. Cada indivíduo é composto por uma estrutura em árvore e representa uma solução para o problema tratado (vide Figura 3.4.1).

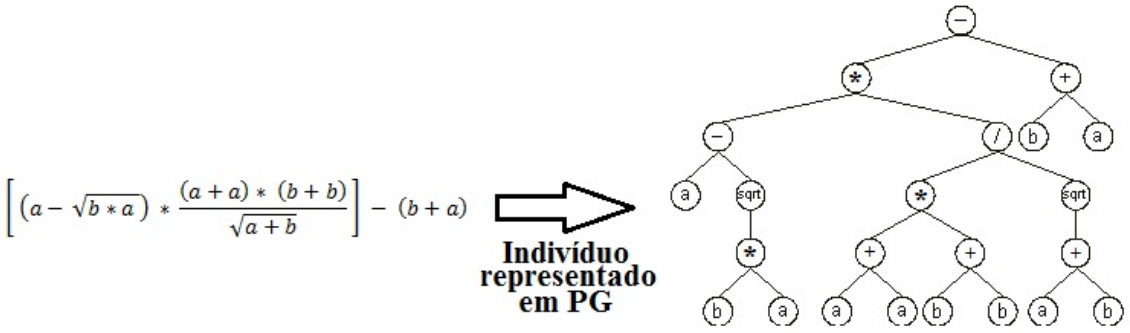

Figura 3.5. Indivíduo representado em uma estrutura de árvore em PG.

Para cada indivíduo é associado um valor de "fitness", que determina o quão bem o indivíduo soluciona o problema considerado. Este valor é determinado por uma função de avaliação, ou função de fitness. Em PG, esse este valor é utilizado para eliminar das populações todos os indivíduos que não estão próximos do objetivo desejado. Os bons indivíduos vão evoluir de geração em geração através dos operadores genéticos.

Os operadores genéticos atuam diretamente sobre as árvores dos indivíduos e devem ser aplicados de modo que apenas árvores válidas sejam formadas. O cruzamento é o operador genético mais usado em PG. Ramos selecionados são trocados entre duas árvores para gerar novos indivíduos, como mostra a figura 3.4.2.

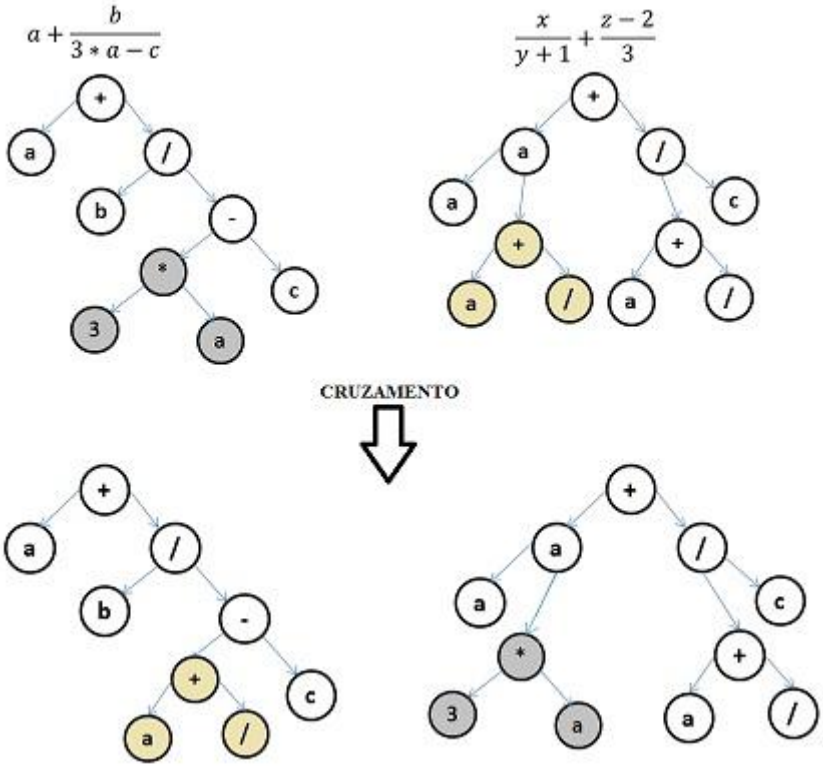

Figura 3.6. Árvore de Cruzamento em PG.

O segundo operador genético usado em PG é a mutação. Ele faz uma seleção de um nó na árvore do indivíduo e o substitui por um novo ramo gerado aleatoriamente como se mostra na figura 3.4.3.

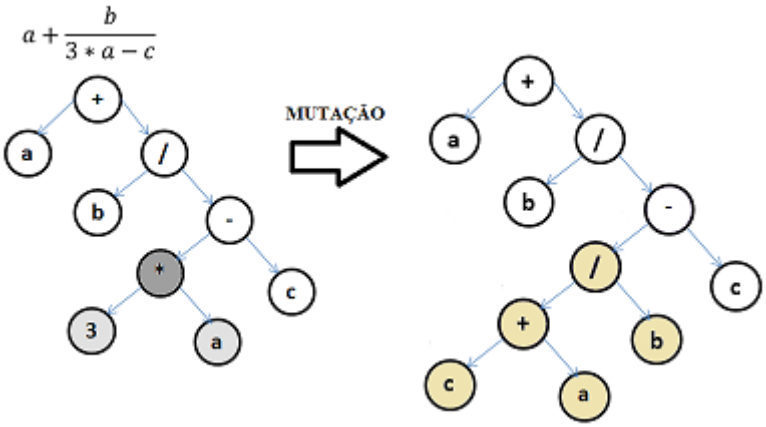

Figura 3.7. Árvore de Mutação em PG.

O terceiro operador genético é a reprodução. Um indivíduo da geração atual é selecionado, e copiado sem alteração alguma, a fim de ser passado para a seguinte geração.

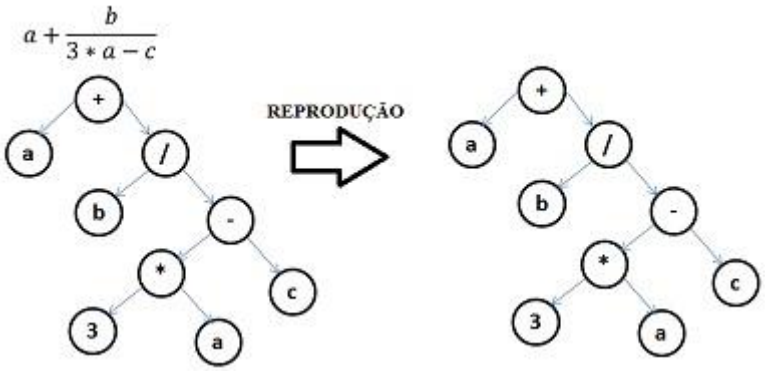

Figura 3.8. Árvore de Reprodução em PG.

Assim, a cada geração os operadores genéticos são aplicados, com o objetivo de substituir uma população atual por uma nova população. A função de fitness é usada para avaliar cada novo indivíduo, e o processo é repetido ao longo de todas as gerações até que um critério de parada seja satisfeito.

Capítulo 4

Aprendendo Funções de Ranking baseados em Blocos usando PG

Conforme dito em capítulos anteriores, alguns trabalhos têm reportado bons resultados ao se empregar programação genética para gerar funções eficazes de ranking [30]. Além disso, também existem estudos que mostram que o uso de informações estruturais das páginas em sistemas de recuperação de informação para a Web pode resultar em melhorias significativas na qualidade do ranking desses sistemas [33]. Neste trabalho, combinamos o uso de programação genética com o uso de informações estruturais das páginas para obter uma função de ranking ainda mais eficaz que os trabalhos anteriores.

A intuição deste trabalho é que a Programação Genética pode explorar o espaço de busca de possíveis soluções para o problema ranking baseado em estrutura, aprender as melhores combinações de diferentes evidências estruturais, gerando indivíduos capazes de estimar a relevância dos documentos para as consultas com grande eficácia. Para isso, adotamos como terminais do processo evolutivo algumas funções que buscam estimar a importância de cada termo de uma página com base na frequência do termo nos blocos dessa página.

4.1 Identificação dos blocos das páginas e geração dos índices de busca

Para viabilizar a estratégia proposta neste trabalho, é necessário identificar os blocos de todas as páginas a serem consideradas nos experimentos. Além disso, nossa estratégia requer que os blocos sejam agrupados em classes, seguindo as diretrizes expostas na Seção 3.5. Conforme descrito no Capítulo 2, em [26] é apresentado duas

abordagens de segmentação, uma automática e outra semiautomática, capazes de identificar os blocos e suas respectivas classes. Ambas as abordagens podem ser aplicadas para satisfazer os requisitos de nosso método.

Uma vez que os blocos foram identificados e classificados, o próximo passo é gerar as estruturas de dados necessárias para o cálculo dos terminais baseados em blocos. Seguindo os moldes do modelo vetorial (vide Capítulo 3.2), são necessários dois tipos de estruturas: o *vocabulário*, e os *índices invertidos* baseados em blocos. O vocabulário é uma estrutura de dados (normalmente uma estrutura *hash*) onde são armazenados todos os termos distintos que ocorrem no conjunto de páginas de uma coleção. Cada termo *t* do vocabulário aponta para uma lista invertida (ou índice invertido) onde cada elemento desta lista armazena a frequência do termo *t* em determinado bloco da coleção. Como podemos observar através da Figura 4.1, cada bloco é representado como uma instância de uma classe *C* dentro de um documento *d*.

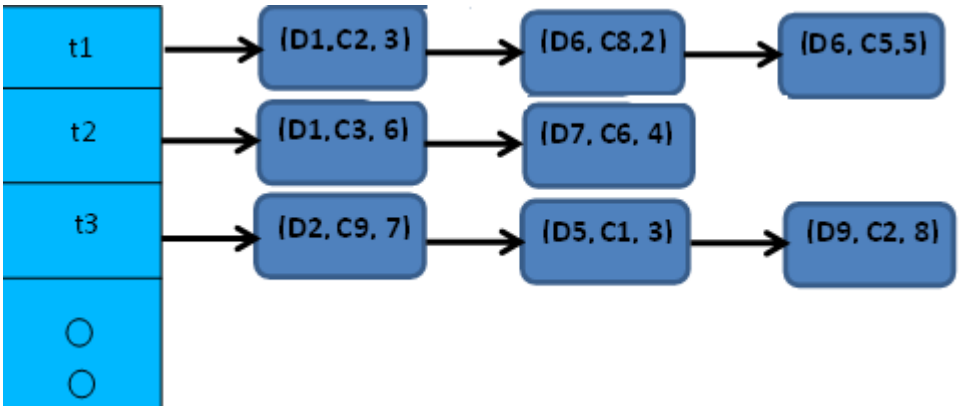

Figura 4.1 Lista invertida contendo informações de blocos. Nesta Figura, C_{bx} representa a classe do bloco b_x presente em dado documento.

4.2 Terminais do processo evolutivo

Através das estruturas de dados descritas na seção anterior, é possível calcular os terminais que serão adotados durante o processo evolutivo (vide tabela 4.1). Escolhemos dois tipos de terminais:

1) *Terminais tradicionais*, que não usam informações de blocos, representados na Tabela 4.1 pelos terminais FT1 a FT17.
2) *Terminais estruturais*, que usam informações de blocos, representados na Tabela 4.1 pelos terminais FT18 a FT24.

Os terminais tradicionais foram selecionados de [30]. Os terminais Ft18 e Ft19 são variações do TF (*term frequency*), apresentado na Seção 3.2. Os terminais Ft20 a Ft22 são variantes do IDF (*inverse document frequency*), também descrito na Seção 3.2. Os terminais Ft23 e Ft24 são variantes da norma dos documentos. Para esses terminais, *tamanho(d)* é a norma do documento *d*, *média[tamanho(d)]* representa a média de normas dos documentos de uma coleção, e k_1 e *b* são constantes cujos valores ideais podem ser estimados através de técnicas de aprendizagem de máquina.

Os terminais estruturais foram selecionados de [33]. Tais terminais são baseados nas funções *bw*, listadas na Tabela 4.2. Uma função block-weight *bw(t,b)* é uma métrica quantitativa associada com o par termo-bloco *[t,b]* que é usada para computar o peso global do termo *t* em relação à página que contém o bloco *b*. Para compor as funções *block-weight bw(t,b)*, são introduzidas duas medidas estatísticas básicas, o *inverse class frequency* (ICF) e o *spread*, que serão explicadas a seguir.

Definição 1. Dado uma classe de blocos $C=\{b_1,...,bn(C)\}$ contendo *n(C)* elementos, e um termo *t* que ocorre em ao menos um bloco de *C*, o *Inverse Class Frequency* de um termo *t* em *C* é definido como

$$\mathrm{ICF}(\mathrm{t},\mathrm{C}) = \log\frac{\mathrm{n}(\mathrm{C})}{\mathrm{n}(\mathrm{t},\mathrm{C})}$$

onde *n(t,C)* é o número de blocos de *C* onde *t* ocorre. O ICF é uma função muito similar ao IDF, mas considera cada classe como uma "coleção de documentos" separada. Como o IDF, o ICF é uma forma de se estimar a quantidade de informação que carrega a ocorrência de um dado termo em uma dada classe de blocos.

Definição 2. O *Spread* de um termo *t* em uma página Web *p*, *Spread(t,p)*, é o número de blocos em *p* que contém *t*, isto é:

$$\mathrm{Spread}(\mathrm{t},\mathrm{p}) = \sum_{\mathrm{b}\in\mathrm{p}} \mathrm{i_b}\,, \text{onde } \mathrm{i_b} = \begin{cases} 1 \text{ se } \mathrm{t}\in\mathrm{b} \\ 0 \text{ caso contrário} \end{cases}$$

A intuição por trás desta medida é que, dado um termo *t* de uma página *p*, quanto maior o número de blocos de *p* que contém o termo *t*, melhor o termo *t* representa o conteúdo de sua página.

As funções *bw*, listadas na Tabela 4.2, representam diferentes estratégias para se computar os fatores *bw* usando as definições de ICF e *Spread*. Estas estratégias são agrupadas em três categorias: *métodos focados em classes*, que atribuem um único valor

de *bw* para todos os termos de uma classe de blocos, representados pelas funções bw_7 a bw_9; *métodos focados em blocos*, que atribuem um único valor de *bw* para todos os termos de um bloco, representados pelas funções bw_4 a bw_6; e *métodos focados em termos*, que permitem que os valores de *bw* variem para termos distintos de um mesmo bloco ou classe, representados pelas funções bw_1 a bw_3. Em [33] é apresentada uma discussão mais aprofundada sobre o ICF e o Spread, bem como sobre as funções *bw*.

As funções *bw* são métricas associadas ao par termo-bloco *[t,b]*, e sozinhas não representam uma estimativa sobre a importância de um termo dentro de um documento. Por causa disso, os terminais estruturais considerados durante o processo evolutivo são obtidos a partir das funções *bw* através das fórmulas listadas na Tabela 1. Nesta tabela, $tf(t,b)$ representa a frequência de um termo *t* em um determinado bloco *b*, e MAX representa uma função que retorna o maior valor de *tf(t,b)*bw(t,b)* considerando cada bloco *b* de uma determinada página.

Uma vez que os valores dos terminais foram calculados para cada termo presente em cada documento da coleção, nós geramos novamente os índices invertidos sem as informações de blocos, mas contendo os valores de todos os terminais calculados a partir do que fora exposto até então nesta seção (vide Figura 4.2). Os novos índices serão utilizados durante o processo seletivo para estimar a precisão dos indivíduos gerados ao longo das gerações.

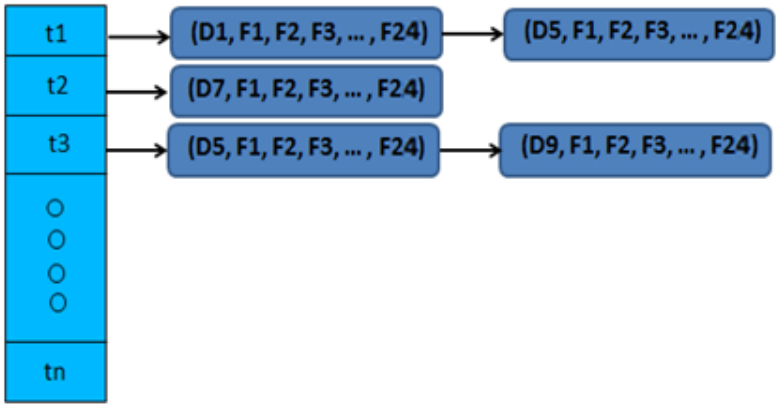

Figura 4.2 Lista invertida que incorpora os valores dos terminais a serem usados durante o processo evolutivo.

4.3 Indivíduos

Conforme exposto no Capítulo 3, os indivíduos são equações geradas ao longo do processo evolutivo, e representam possíveis soluções para o problema que se deseja solucionar. No contexto deste trabalho, os indivíduos representam funções de ranking

geradas a cada geração, que são avaliadas de acordo com sua capacidade de encontrar os documentos relevantes para as consultas propostas. Como todo trabalho envolvendo Programação Genética, representamos as equações dos indivíduos através de uma estrutura em árvore. As funções de adição (+), subtração (-), multiplicação (*), divisão (/) e logaritmo (log) irão compor os nós internos das árvores dos indivíduos, enquanto que os nós folhas serão representados pelos terminais tradicionais e estruturais apresentados na Tabela 4.1.

Tabela 4.1 Terminais Tradicionais usados em PG.

Lista de Terminais		
Id	**Terminal**	**Descrição**
Ft1	$\sum_{b\in\rho} tf(t,b) * bw_1$	Em [33]
Ft2	$MAX[tf(t,b) * bw_1]$	Em [33]
Ft3	$\sum_{b\in\rho} tf(t,b) * bw_2$	Em [33]
Ft4	$\sum_{b\in\rho} tf(t,b) * bw_3$	Em [33]
Ft5	$MAX[tf(t,b) * bw_3]$	Em [33]
Ft6	$\sum_{b\in\rho} tf(t,b) * bw_4$	Em [33]
Ft7	$MAX[tf(t,b) * bw_4]$	Em [33]
Ft8	$\sum_{b\in\rho} tf(t,b) * bw_5$	Em [33]
Ft9	$MAX[tf(t,b) * bw_5]$	Em [33]
Ft10	$\sum_{b\in\rho} tf(t,b) * bw_6$	Em [33]
Ft11	$MAX[tf(t,b) * bw_6]$	Em [33]
Ft12	$\sum_{b\in\rho} tf(t,b) * bw_7$	Em [33]
Ft13	$MAX[tf(t,b) * bw_7]$	Em [33]
Ft14	$\sum_{b\in\rho} tf(t,b) * bw_8$	Em [33]
Ft15	$MAX[tf(t,b) * bw_8]$	Em [33]
Ft16	$\sum_{b\in\rho} tf(t,b) * bw_9$	Em [33]

Ft17	$MAX[tf(t,b) * bw_9]$	Em [33]
Ft18	tf	Em [30]
Ft19	$1 + \log(tf)$	Em [30]
Ft20	$0{,}5 + \frac{0{,}5 + tf}{\max(tf)}$	Em [30]
Ft21	$\log(\frac{N}{n_t})$	Em [30]
Ft22	$\log(\frac{N - n_t + 0{,}5}{n_t + 0{,}5})$	Em [30]
Ft23	$tamanho(d)$	Em [30]
Ft24	$\frac{1}{(k_1 * (1-b) + b * tamanho(d)/média[tamanho(d)]) + tf}$	Em [30]

Tabela 4.2 Terminais com informações de blocos usados em PG.

Lista de Pesos de Bloco		
Id	**Fórmula do Peso**	**Descrição**
$bw_1(t,b)$	$ICF(t, C_b)$	Em [33]
$bw_2(t,b)$	$Spread(t, \rho_b)$	Em [33]
$bw_3(t,b)$	$ICF(t, C_b) * Spread(t, \rho_b)$	Em [33]
$bw_4(t,b)$	$\begin{cases} \frac{\sum_{t' \in b} ICF(t', C_b)}{\lvert b \rvert} & se\ t \in b \\ 0 & outro\ caso \end{cases}$	Em [33]
$bw_5(t,b)$	$\begin{cases} \frac{\sum_{t' \in b} Spread(t', \rho_b)}{\lvert b \rvert} & se\ t \in b \\ 0 & outro\ caso \end{cases}$	Em [33]
$bw_6(t,b)$	$\begin{cases} \frac{\sum_{t' \in b} ICF(t', C_b) * Spread(t', \rho_b)}{\lvert b \rvert} & se\ t \in b \\ 0 & outro\ caso \end{cases}$	Em [33]
$bw_7(t,b)$	$\begin{cases} \frac{\sum_{t' \in v(C_b)} ICF(t', C_b)}{\lvert v(C_b) \rvert} & se\ t \in b \\ 0 & outro\ caso \end{cases}$	Em [33]
$bw_8(t,b)$	$\begin{cases} \frac{\sum_{b' \in C_b} \frac{\sum_{t' \in C_b} Spread(t', \rho_{b'})}{\lvert b' \rvert}}{\lvert (C_b) \rvert} & se\ t \in b \\ 0 & outro\ caso \end{cases}$	Em [33]
$bw_9(t,b)$	$bw_7(t,b) * bw_8(t,b)$	Em [33]

4.4 Função Fitness

Conforme exposto no Capítulo 3, a função de fitness tem como objetivo avaliar cada indivíduo gerado através do processo evolutivo de acordo com sua capacidade de solucionar o problema a ser resolvido. No contexto deste trabalho, a função de fitness avalia cada indivíduo de acordo com a precisão dos documentos retornados para um conjunto de consultas.

Em nossos experimentos, optamos por adotar a métrica Bpref10 [colocar referencia] (*binary preference-based measure*) como função de fitness. O Bpref10 é uma métrica criada para comparar diferentes fórmulas de ranking quando apenas uma parte dos documentos foram avaliados como relevantes ou não para um conjunto de consultas. Desta forma, se um dado indivíduo retornar um documento não avaliado nas primeiras posições do ranking para uma dada consulta, esse documento não será considerado no cálculo da precisão do indivíduo.

O Bpref10 é calculado através da fórmula abaixo:

$$Bpref10 = \frac{1}{R}\sum_{r=1}^{R} 1 - \frac{\mathrm{Irrel}_R(r)}{R+10}$$

onde *R* é o número de documentos julgados como relevantes, $Irrel_R(r)$ é o número de documentos julgados como irrelevantes que ocorrem antes de *r* no conjunto de respostas e que ocorrem entre os primeiros *R+10* documentos julgados como não relevantes recuperados pelo indivíduo.

4.5 Seleção do Melhor Indivíduo

Neste trabalho para selecionar o melhor indivíduo, tomamos como referência a mesma estratégia adotada em nosso *baseline:2* [30], que adota programação genética usando terminais tradicionais de recuperação de informação. No *baseline:2* se definiu uma função de avaliação para ajudar na escolha de boas soluções. Neste trabalho, a escolha do melhor indivíduo é feita considerando o desempenho dos indivíduos nos conjuntos de treino e de validação, para finalmente subtrair o desvio padrão desses valores. Chamamos esta função como: $f_{seleção}$

$$f_{seleção} = (Tr_i + Vl_i) - \sigma_i$$

No *baseline:2* foi mostrado que essa abordagem é bastante efetiva. Formalmente, se define Tr_i como o desempenho de um indivíduo i na base do treino, Vl_i como o desempenho deste indivíduo na base de validação, e σ_i como o desvio padrão dos valores de Tr_i e Vl_i.

Apresentamos uma descrição geral do processo interno na programação genética, para um melhor entendimento de nosso trabalho:

Descrição geral do PG na nossa Proposta

1. PI ← Criar a população inicial (aleatoriamente).
2. MTr ← ø
3. Para cada geração G de NG gerações fazer {
4. Ftr ← ø
5. Para cada indivíduo I que pertence a PI fazer {
6. Ftr ← Ftr + {G, I, Bpref10(I,Tr)} }
7. MTr ← MTr U BomIndivíduo(NI, FTr)
8. PI ← OperaçõesGenéticas(PI, FTr, MTr, G) }
9. MVl ← ø
10. Para cada indivíduo I que pertence a Tr fazer {
11. MVl ← MVl + {I, Bpref10(I,Vl)} }
12. MelhorIndivíduo ← $f_{seleção}$(MTr, MVl)
13. Lista_Ranking ← Func_ranking(MelhorIndivíduo, Te)
14. Melhor_Func-Rankig ← Obtem_Maior_MAP(Lista_Ranking, MelhorIndivíduo)
15. Mostrar(Melhor_Func-Rankig)
16. FIM.

- Tr = Grupo de dados de teste.
- Vl = Grupo de dados de validação.
- Te = Grupo de dados de treino.
- NG = Numero de gerações que vai processar.
- NI = Numero de melhores indivíduos.
- PI = A população inicial de indivíduos feito aleatoriamente.
- MVl = Armazena lista dos melhores indivíduos da validação.
- MTr = Armazena lista dos melhores indivíduos do treino.
- G = Número de Gerações.
- I = Número de individuos.
- FTR = Armazena lista dos resultados do fitness dos dados de treino.

Capítulo 5

Experimentos

5.1 Coleções das Páginas Web

Para confirmar a eficácia de nossos métodos, realizamos experimentos de busca em 3 coleções de páginas Web denominadas IG, CNN e BLOGs. Essas coleções foram segmentadas utilizando a abordagem semi-automática proposta em [34], que identifica os blocos e os classifica de acordo com as funções que desempenham dentro das páginas (vide seção 3.3).

A coleção IG contém 34.460 páginas obtidas a partir do IG (*Internet Group*, www.ig.com.br), um dos maiores portais da Web brasileira. Esta coleção é composta por um site de notícias, um fórum on-line, e um site de receitas culinárias. O processo de segmentação encontrou 407.020 blocos organizados em 104 classes. A coleção dispõe de um conjunto de 50 consultas avaliadas por especialistas através do método de pooling [33,34].

A segunda coleção, CNN, é resultado de uma coleta feita no site internacional da CNN (www.cnn.com). Essa coleção contém um total de 16.257 páginas, onde foram encontrados 25.7156 blocos distribuídos em 158 classes. A coleção CNN também dispõe de um conjunto de 50 consultas, cada qual avaliada por um conjunto de especialistas.

A coleção BLOG é uma coleção que contem 54.055 páginas, 161 classes de blocos e 104.2624 blocos. A coleção foi coletada dos 9 blogs mais populares do mundo em 2008, de acordo com o blog Technorati. A Tabela 5.1 lista os blogs coletados. Essa coleção contém 52 consultas, cada qual avaliada por um conjunto de especialistas.

Tabela 5.1 Descrição das Coleções IG, CNN e BLOG

Coleção	Site	Páginas	Domain
IG	News	26,466	www.ultimosegundo.com.br
	Forum	6,389	www.jornaldedebates.com.br
	Recipe	1,605	www.panelinha.com.br
	Total	34,460	
CNN	News	16,257	www.cnn.com
	Total	16,257	
BLOG	Boing Boing	14,173	www.boingboing.net
	CNET	8,054	news.cnet.comtech-blogs
	Engadget	6,343	www.engadget.com
	Gizmodo	4,454	us.gizmodo.com
	Google	1,050	googleblog.blogspot.com
	Life Hacker	3,997	www.lifehacker.com
	Mashable	7,410	www.mashable.com
	Slash Film	5,376	www.slashfilm.com
	Tech Crunch	3,198	www.techcrunch.com
	Total	54,055	

5.2 Geração das Bases de Treino, Validação e Teste para PG

Para cada uma das quatro coleções listadas na seção anterior, nós executamos os seguintes passos durante a experimentação. Primeiro, submetemos cada consulta da coleção a uma implementação do modelo vetorial tradicional, fato que nos retornou o conjunto de documentos da coleção que possuem alguma similaridade para com a consulta. De posse desses documentos, calculamos o valor de cada terminal da Tabela 4.1 para cada documento retornado, o que nos deu uma lista invertida similar à da Figura 4.2. Quando a consulta possui mais de um termo, o valor de um terminal para esta consulta corresponde à soma dos valores deste terminal para cada um de seus termos. Uma vez terminado esse processo, cada documento retornado para uma consulta possui um vetor de valores (cada qual associado a um terminal da Figura 4.1), bem como possui a informação sobre se este documento é relevante ou não para a consulta considerada.

Adotamos uma técnica de validação cruzada em nossos experimentos, realizando procedimentos de treino, validação e teste em diferentes subconjuntos de cada coleção.

Para tanto, dividimos o conjunto de consultas de cada base em três sub conjuntos, chamados treino, validação e teste. Desta forma cada coleção possui cinco folds, cada uma delas tem uma forma diferente de distribuição das bases de treino, validação e teste.

A base de dados da coleção de IG e CNN estão conformadas por 50 consultas, distribuímos para o treino 25 consultas, para a validação 10 consultas e para o teste 15 consultas. No caso da coleção BLOG está conformada por 52 consultas, distribuímos para o treino 27 consultas, para a validação 10 consultas e para o teste 15 consultas. A Figura 5.2 apresenta a ideia geral do processo para gerar as bases de dados de treino, validação e teste que são usados em PG.

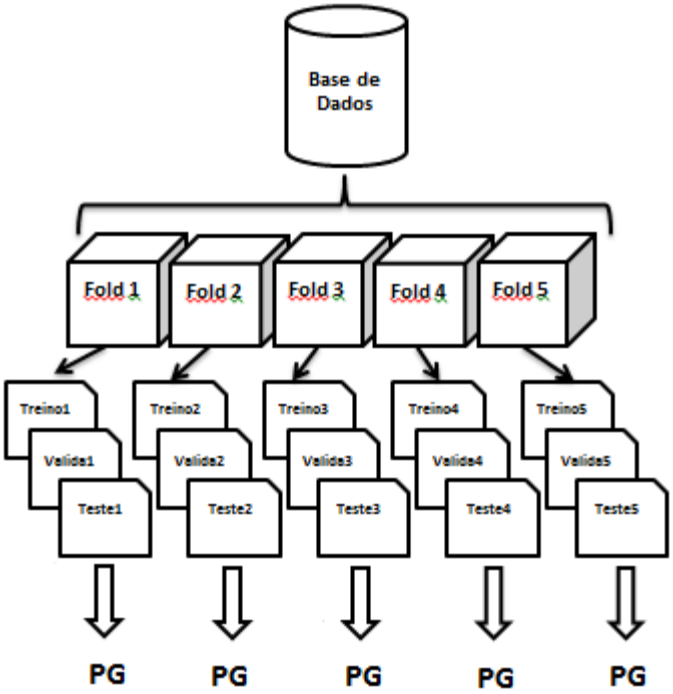

Figura 5.2 Ideia básica da Distribuição dos Folds.

O processo da figura 5.2 é aplicado em neste trabalho para as três coleções (IG, CNN e BLOG).

5.3 Avaliação e Baselines

Foram comparados os resultados de nossa abordagem com outros dois trabalhos. O *baseline:1* [33] que usa informações de blocos sem aprendizagem de máquina e outro *baseline:2* [30] que usa evidências tradicionais em recuperação de informação de eficácia já comprovada em PG.

Em relação ao *baseline:1*, os autores apresentam um método baseado nas informações da estrutura em blocos das páginas para fazer um ranking sem utilizar nenhuma técnica de aprendizagem máquina. Nosso trabalho usa essas mesmas informações em blocos extraídas das mesmas coleções, com objetivo de melhorar o ranking quando essas informações são utilizadas como evidências para PG. No *baseline:1* se usam as métricas de avaliação em recuperação de informação MAP e Bpref-10, como em nosso trabalho, isso para que nossa comparação seja justa. A MAP é a média da precisão obtida após que cada documento relevante é recuperado, definindo a precisão como a relação entre o numero de documentos relevantes recuperados e o numero de documentos recuperados. Para o cálculo da MAP se considera 3 passos fundamentais: o primeiro, quando ainda não se recuperou nenhum documento relevante a precisão é zero. O segundo, cada vez que se obtém um documento relevante, se tem que calcular a precisão. Finalmente o terceiro, que a MAP se calcula como a média aritmética das precisões anteriores. No caso de Bpref-10 já foi explicado na seção 4.4.

No *baseline:2*, os autores apresentam um método baseado em PG baseada em informações tradicionais de recuperação de informação, aplicado na base de dados TREC. É bom ressaltar que o *baseline:2* é o trabalho não supervisionado mais recente na literatura. Nosso trabalho usou as evidências do *baseline:2* para PG, aplicado a nossas coleções, para determinar que os resultados do ranking melhoram quando essas evidências são combinadas com evidências baseados em blocos. Por isso determinamos o impacto que tem as evidências de *baseline:2* em nossas coleções, para fazer uma comparação justa com nosso método.

Finalmente, a diferencia do *baseline:1*, nossa proposta adiciona informações tradicionais de recuperação de informação de eficácia já comprovada e a diferencia do *baseline:2*, nossa proposta adiciona informações baseadas na estrutura dos blocos das paginas. Desta forma, nossa proposta combina as evidências do *baseline:1* e *baseline:2* em nossas coleções usando PG para melhorar significativamente os resultados do ranking das paginas.

5.4 Parâmetros Iniciais de PG

O aprendizado utilizando no conjunto de treinamento aconteceu utilizando diferentes parâmetros. O tamanho da população utilizado em nossos experimentos foi de 750 indivíduos. A profundidade máxima das árvores para representar os indivíduos

foi igual a 17. Em todos os experimentos as populações foram criadas utilizando sementes aleatórias e o processo de evolução ocorreu até alcançarmos 30 gerações. Devido à estabilidade dos resultados após 30 gerações, definiu-se esse valor como o critério de terminação. A semente aleatória utilizada foi 245. Os operadores genéticos foram: 85% de crossover, 10% de mutação e 5% de reprodução. No final de cada geração, a fase de teste foi executada para com os 10 melhores indivíduos descobertos na fase de treino e validação daquela geração. Os terminais foram os apresentados na tabela 4.2.1 e com o conjunto de funções de soma "+", subtrair "-", multiplicação "*", divisão "/" e logaritmo "log".

5.5 Análise dos Resultados

Nesta seção apresentamos os resultados de nossos experimentos. Em todas as coleções (IG, CNN e BLOG) fizemos uma média dos valores do MAP de cada geração de cada fold, que foram aplicados a nosso ranking obtido com PG, como se apresenta na figura 5.5.1 para IG, na figura 5.5.3 para CNN e na figura 5.5.5 para BLOG. Na coleção IG os resultados obtidos se apresentam como segue:

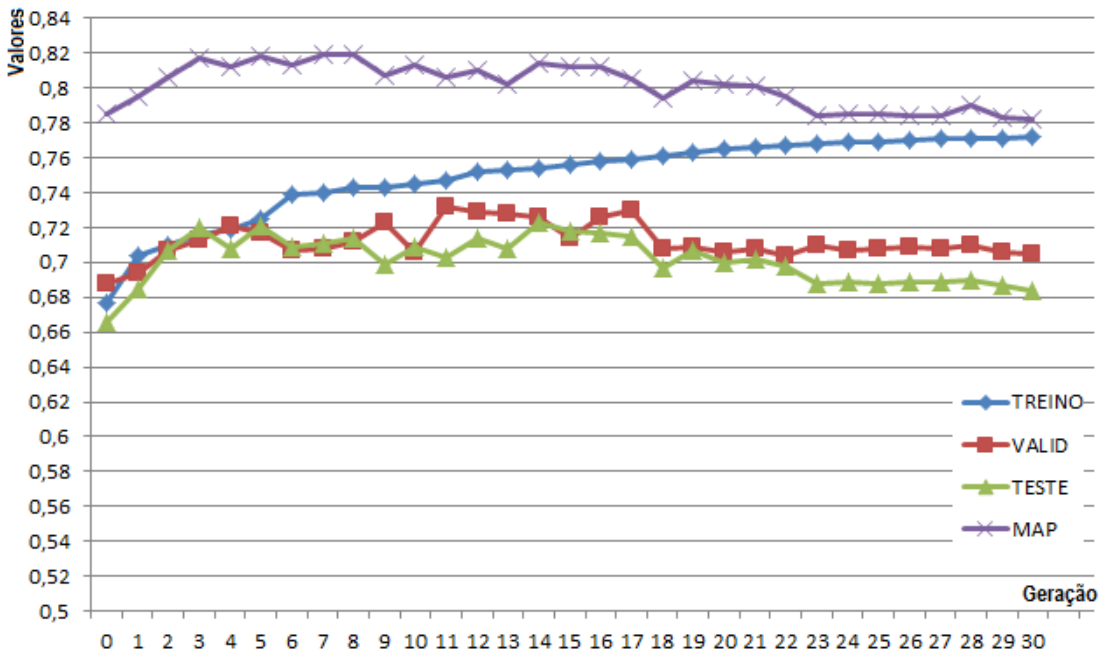

Figura 5.5.1 Resumo dos valores do MAP em cada Geração na Coleção IG.

Nossa função de ranking teve um valor maior do MAP de 0,8192. Esse valor representa um ganho de 9,38% sobre *baseline:1*, de 5,25% sobre *baseline:2* e 31,92% sobre BM25. A figura 5.5.2 apresenta curvas comparativas.

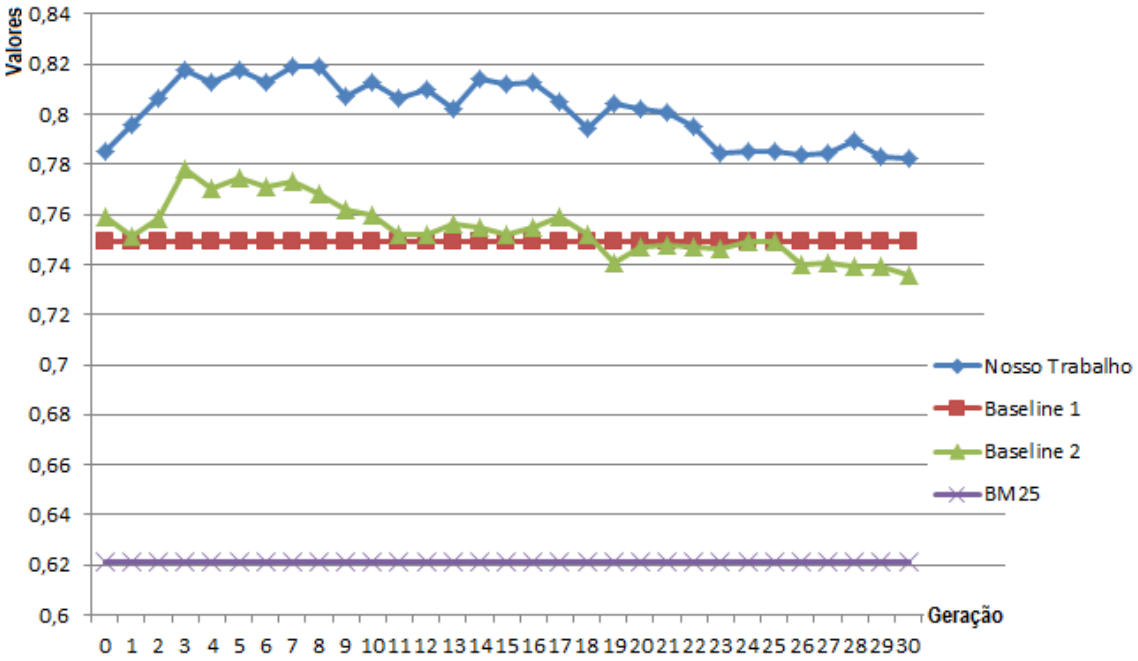

Figura 5.5.2 Resumo Comparativo Coleção IG.

Apresentamos uma tabela comparativa dos resultados obtidos neste trabalho em IG e os resultados obtidos por nossos *baseline:1* e *baseline:2*.

Tabela 5.5.1 Dados Comparativos na Coleção IG.

Resultados Comparativos do MAP na Coleção IG	
MAP em nosso trabalho	0,819223
MAP Baseline1	0,749000
MAP Baseline2	0,778375
MAP BM25	0,621000

Na coleção CNN, os resultados obtidos se apresentam como segue:

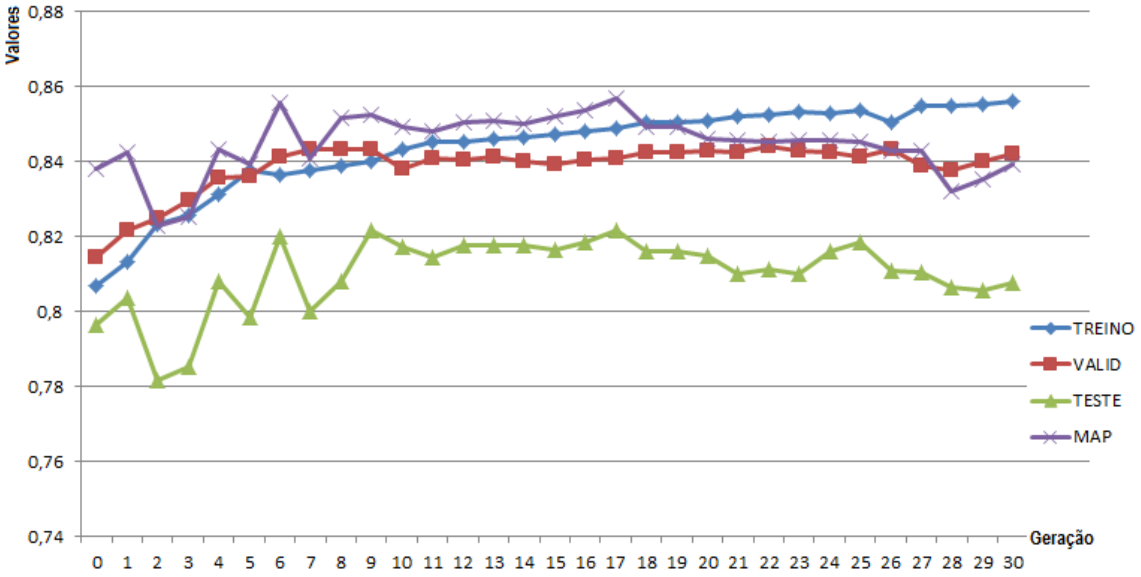

Figura 5.5.3 Resumo dos valores do MAP em cada Geração na Coleção CNN.

Nossa função de ranking teve um valor maior do MAP de 0,85705. Esse valor representa um ganho de 7,13% sobre *baseline:1*, de 10,37% sobre *baseline:2* e 24,03% sobre BM25. A figura 5.5.2 apresenta curvas comparativas.

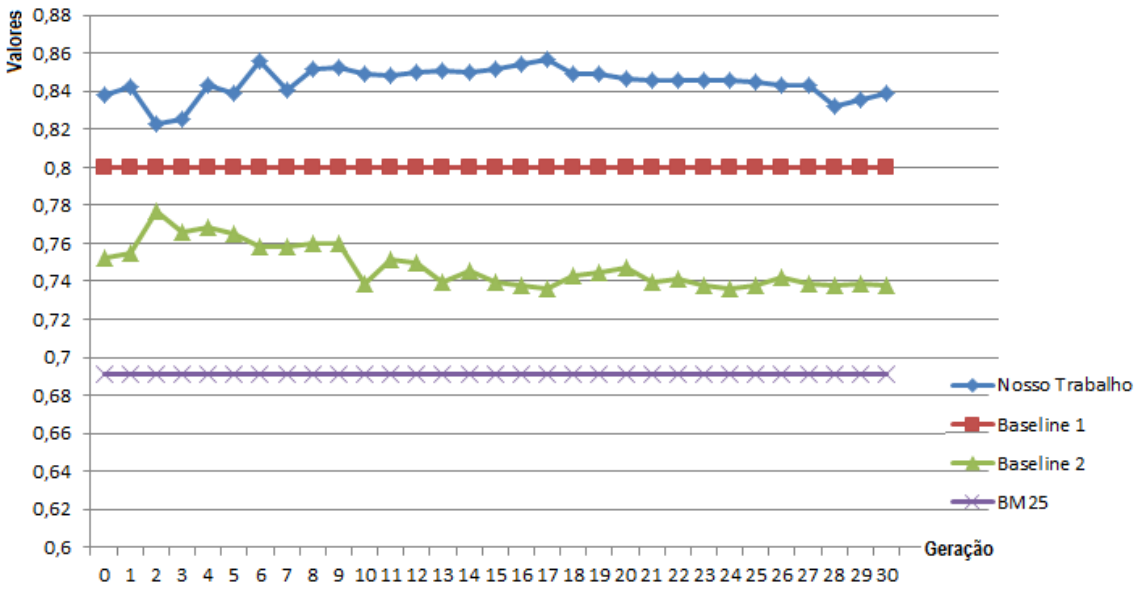

Figura 5.5.4 Resumo Comparativo Coleção CNN.

Apresentamos uma tabela comparativa dos resultados obtidos neste trabalho em CNN e os resultados obtidos por nossos *baseline:1* e *baseline:2*.

Tabela 5.5.2 Dados Comparativos na Coleção CNN.

Resultados Comparativos do MAP na Coleção CNN	
MAP em nosso trabalho	0,857050
MAP Baseline1	0,800000
MAP Baseline2	0,776520
MAP BM25	0,691000

Na coleção BLOG, os resultados obtidos se apresentam como segue:

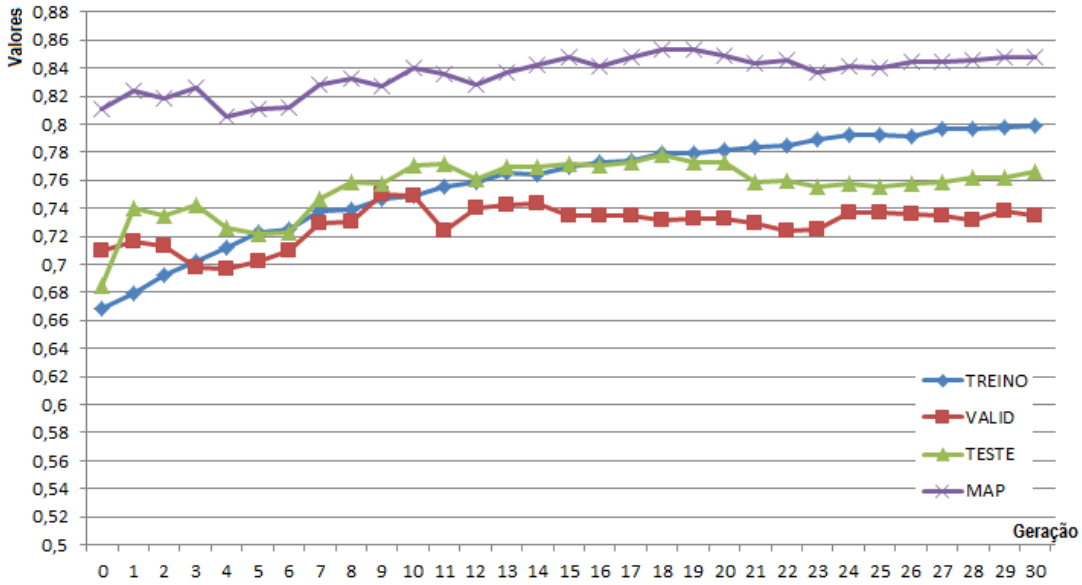

Figura 5.5.5 Resumo dos valores do MAP em cada Geração na Coleção BLOG.

Nossa função de ranking teve um valor maior do MAP de 0,85341. Esse valor representa um ganho de 25,87% sobre *baseline:1*, de 4,37% sobre *baseline:2* e 32,52% sobre BM25. A figura 5.5.2 apresenta curvas comparativas.

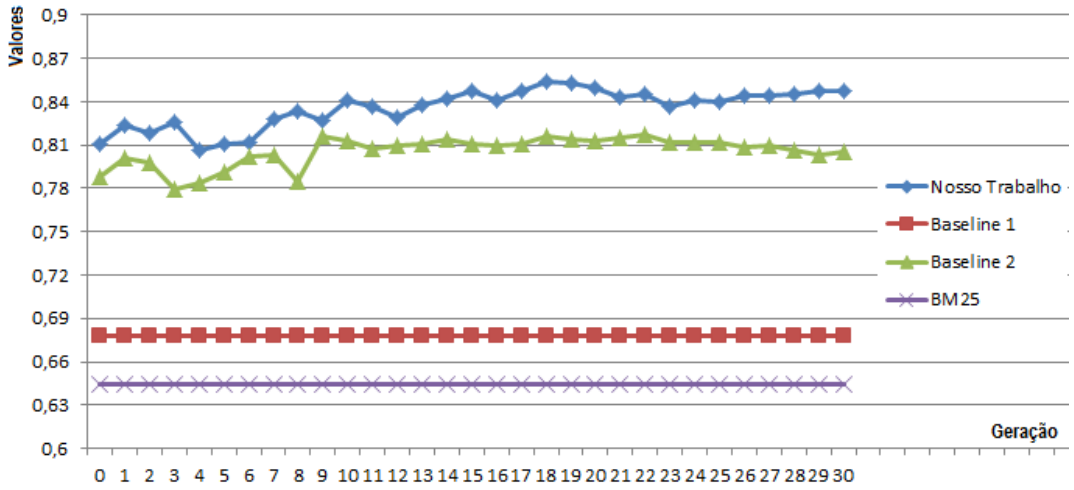

Figura 5.5.6 Resumo Comparativo Coleção BLOG.

Apresentamos uma tabela comparativa dos resultados obtidos neste trabalho em BLOG e os resultados obtidos por nossos *baseline:1* e *baseline:2*.

Tabela 5.5.3 Dados Comparativos na Coleção BLOG.

Resultados Comparativos do MAP na Coleção BLOG	
MAP em nosso trabalho	0,853410
MAP Baseline1	0,678000
MAP Baseline2	0,817690
MAP BM25	0,644000

Os resultados de nosso trabalho superam significativamente os resultados obtidos por nossos *baselines* em cada uma das três coleções. Desta forma fica demostrado que combinar evidências baseados na estrutura em blocos das páginas com evidências tradicionais em recuperação de informação usando PG, têm melhores resultados quando esses não são combinados. Além disso, não se conhece mais pesquisas que combinam essas evidências, por tal razão, se destaca a importância de nossos resultados para futuras pesquisas na área.

Capítulo 6

Análise dos Resultados Gerados pela PG

Os resultados geradores pela PG, podemos observar que todas as funções de ranking com maior valor de MAP em cada fold de todas as coleções, sempre estão presentes terminais baseados em blocos e terminais baseados em informações tradicionais de recuperação de informação. Ou seja, fica comprovado que as melhores funções de ranking aprendidas pela PG contêm uma combinação destes terminais. Por exemplo, podemos visualizar esse comportamento na função de ranking obtida no fold1 na coleção BLOG, na geração 10, com o valor de MAP no teste de 0,950503. Lembrando que os terminais com informações de blocos estão definidos entre ft1 até ft17 e os terminais com informações tradicionais de recuperação de informação entre ft18 até ft24.

```
(- (/ (- (/ (- (* ft20 ft19) (+ (+ (log (/ (- ft20 ft1) (- ft4 ft2))) (/ ft23 ft9)) (log ft17))) (- (- ft13 ft14) (- (+ ft12 ft13) (log ft9)))) (log (/ (- ft20 ft1) (- (- ft20 ft12) (/ ft19 ft4))))) (log ft21)) (log (/ (+ ft22 ft9) (log (+ ft14 ft23)))))
```

Também acontece o mesmo comportamento na função de ranking obtida no fold4 na coleção CNN, na geração 24, com o valor de MAP no teste de 0,921819.

```
(- (* ft9 (- ft6 (* (log (+ ft9 ft7)) (* ft9 ft16)))) (/ (-
(* (* ft14 (- (- ft12 ft5) (log ft19))) (/ ft10 ft15)) (-
(+ (* (log (log (+ ft9 ft7))) (* (log ft9) (- (- ft20 ft16)
(* ft20 (* (+ (* (* (/ ft1 ft8) (* ft9 ft16)) (* ft9 ft16))
ft5) (log ft3)))))) (/ (* (/ ft16 ft8) (* ft9 ft16)) (- (*
(log ft9) (- ft11 ft10)) (log ft4)))) (+ (log ft19) (log
ft4)))) ft18))
```

O mesmo comportamento na função de ranking obtida no fold3 na coleção IG, na geração 19, com o valor de MAP no teste de 0,872049.

```
(/ (- (- (* (- ft20 ft16) (/ ft16 ft18)) (/ ft4 ft18)) (-
(+ ft23 ft7) (log ft12))) (- (- (+ ft23 ft7) ft10) (/ (* (-
ft20 ft16) (/ ft16 ft18)) (- (- (+ ft23 ft7) ft10) (/ (-
ft15 ft16) (- (log ft4) (log (/ (+ (* (/ (log ft8) (- ft18
ft1)) (* (log ft11) (/ ft16 ft9))) (log (log (+ ft12
ft10)))) ft5)))))))))
```

No caso das funções soma “+”, subtração “-“, multiplicação “*”, divisão “/” e logaritmo “log” também estão presentes em todas as funções de ranking geradas pela PG, o que significa que tais funções são importantes para obter boas funções de ranking. Neste trabalho tomamos em conta a função subtração “-“ que a diferencia de outros trabalhos que usam PG, não foi considerado.

Capítulo 7

Conclusões e Trabalhos Futuros

Neste trabalho apresentamos uma nova abordagem baseada na combinação de evidências extraídas da estrutura em blocos das páginas com evidências extraídas de conhecidas fórmulas de ranking de recuperação de informação, usando programação genética. Nós mostramos que a nossa abordagem melhora o desempenho de recuperar informação em relação aos nossos *baselines* [11, 12]. Usamos as coleções IG, CNN e BLOG para validar nossa abordagem.

Nossos experimentos mostram que nosso método leva a melhorias significativas na eficácia de recuperar informação baseadas em nossas funções de ranking geradas pela PG. Apresentando ganhos de precisão (MAP) de 9,38% na coleção IG, de 7,13% na CNN, e 25,87% na coleção de BLOG em relação ao *baseline* [33] que usa informações de blocos sem técnicas de aprendizagem máquina. Em relação a nosso segundo *baseline* [30] que usa programação genética a partir de evidências tradicionais de recuperação de informação, de comprovada eficácia, nosso método conseguiu ganhos de 5,25% na coleção IG, 10,37% na CNN e 4,37% na coleção de BLOG.

Os resultados obtidos neste trabalho nos levam a concluir que combinar esses dois tipos de evidências como terminais de entrada para PG melhora a qualidade do processo de descoberta de funções de ranking.

7.1 Trabalhos Futuros

No futuro pretendemos estudar se as funções de ranking obtidas neste trabalho em uma coleção são estáveis para fazer bons rankings em outras coleções. Por exemplo, as funções de ranking obtidas na coleção IG consegue fazer um bom ranking nas coleções

CNN e BLOG. Da mesma forma as funções de ranking obtidas na coleção CNN é estável para fazer um bom ranking nas coleções IG e BLOG.

Outro trabalho que se pretende estudar é combinar as informações baseadas na estrutura em blocos das páginas com outras técnicas de aprendizagem máquina para determinar o grau de importância que realmente tem essas informações na área da recuperação de informação.

REFERÊNCIAS

[1] SALTON, GERARD; LESK, Michael E. Computer evaluation of indexing and text processing. Journal of the ACM (JACM), v. 15, n. 1, p. 8-36, 1968.

[2] SALTON, GERARD. The SMART retrieval system—experiments in automatic document processing. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[3] J. R. Koza. Genetic programming as means for programming computer by natural selection. MIT Press, Cambridge, 1992.

[4] CALLAN, J. P. Passage-level evidence in document retrieval. In SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA. Springer-Verlag New York, Inc. p. 302-310, 1994.

[5] B. T. BARTELL, G. W. COTTRELL, and R. K. BELEW. Automatic combination of multiple ranked retrieval systems. In Proceedings of the 17th ACM SIGI, p. 173–181, 1994.

[6] ROBERTSON, STEPHEN E.; WALKER, STEVE. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer-Verlag New York, Inc., p. 232-241, 1994.

[7] J. PONTE and W. B. CROFT. A language model approach to information retrieval. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. p. 275-281, USA, 1998.

[8] Hawking, D. and Craswell, N. (1998). Overview of TREC-7 very large collection track. In Proc. of the Seventh Text Retrieval Conf., pages 91--104.

[9] R. BAEZA-YATES and B. RIBEIRO-NETO. Modern Information Retrieval. Addison-Wesley-Longman, Boston, MA, 1999.

[10] BERGER, A; LAFFERTY, J. Information retrieval as statistical translation. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, p. 222-229, 1999.

[11] Hawking, D., Craswell, N., Thistlewaite, P., and Harman, D. (1999). Results and challenges in web search evaluation. Comput. Netw., 31(11-16):1321-1330.

[12] J. LAFFERTY., C. ZHAI. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p. 111-119, USA, 2001.

[13] MARTÍNEZ MÉNDEZ F. J., Propuesta y Desarrollo de un Modelo para la Evaluación de la Recuperación de Información en Internet. Tesis Doctoral, Universidad de Murcia. España. 2002.

[14] SONG, R., LIU, H., WEN, J. R., & MA, W. Y. Learning Block importance Models for Web Pages. In: Proceedings of the 13th international conference on World Wide Web. ACM, p. 203-211, 2004.

[15] FAN, W., GORDON, M. D., & PATHAK, P. Discovery of context-specific ranking functions for effective information retrieval using genetic programming. Knowledge and Data Engineering, IEEE Transactions on, v. 16, n. 4, p. 523-527, 2004.

[16] FAN, W., GORDON, M. D., PATHAK, P., XI, W., & FOX, E. A. Ranking function optimization for effective Web search by genetic programming: An empirical study. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on (pp. 8-pp). IEEE. Janeiro 2004.

[17] FAN, W., FOX, E. A., PATHAK, P., & WU, H. The effects of fitness functions on genetic programming-based ranking discovery for Web search. Journal of the American Society for Information Science and Technology, v. 55, n. 7, p. 628-636, 2004.

[18] CAI, D., YU, S., WEN, J.-R., and MA, W.-Y. Block-based Web Search. In SIGIR ’04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA. ACM. p. 456-463, 2004.

[19] FAN, W.; GORDON, M. D.; PATHAK, P. A generic ranking function discovery framework by genetic programming for information retrieval. Information Processing & Management, v. 40, n. 4, p. 587-602, 2004.

[20] BURGUES C. J.C. Ranking as Learning Structured Outputs. Microsoft Research One Microsoft Way Redmond,WA 98052-6399. USA. 2005.

[21] BRINKER K., HÜLLERMEIER E., Calibrated Label-Ranking. Data and Knowledge Engineering Otto-von-Guericke-Universität Magdeburg. Germany. 2005.

[22] CHU W., GHAHRAMANI Z., Extensions of Gaussian Processes for Ranking: Semi-supervised and Active Learning. Gatsby Computational Neuroscience Unit University College London. London. 2005.

[23] GRANGIER D., BENGIO S., Exploiting Hyperlinks to Learn a Retrieval Model. IDIAP Research Institute, Martigny, Switzerland, Ecole Polytechnique Federale de Lausanne (EPFL). Switzerland. 2005.

[24] RAJARAM S., AGARWAL S., Generalization Bounds for k-Partite Ranking. in Proceedings of the NIPS 2005 workshop on Learning to Rank, Vancouver 2005.

[25] A. Trotman. Learning to rank. Information Retrieval, v. 8, n. 3, p.359–381, 2005.

[26] B. PÔSSAS, N. ZIVIANI, J. WAGNER MEIRA, and B. RIBEIRO-NETO. Set-based vector model: An efficient approach for correlation-based ranking. ACM TOIS, v. 23, n. 4, p. 397–429, 2005.

[27] GRANGIER, D., & BENGIO, S. Exploiting hyperlinks to learn a retrieval model. 2005.

[28] A. LACERDA, M. CRISTO, M. A. GONÇALVES, W. FAN, N. ZIVIANI, and B. RIBEIRO-NETO. Learning to advertise. In Proceedings of the 29th ACM SIGIR, p. 549-556, 2006.

[29] JEN-YUAN YEH, JUNG-YI LIN, HAO-REN KE, WEI-PANG YANG, Learning to Rank for Information Retrieval Using Genetic Programming. SIGIR 2007 Workshop. Taiwan.2007.

[30] H. MOSRRI DE ALMEIDA, M. GONÇALVES, MARCO CRISTO, P. CALADO., A Combined Component Approach for Finding Collection-Adapted Ranking Functions based on Genetic Programming. SIGIR 2007 Proceedings. Amsterdam, 2007.

[31] FERNANDES D., DE MOURA E. S., RIBEIRO-NETO B., DA SILVA A. S., GONÇALVES M. A., Computing block importance for searching on Web sites. CIKM 2007. p. 165-174, Lisboa, Novembro, 2007.

[32] CHRIS J.C. BURGES., KRYSTA M. SVORE., A Machine Learning Approach for Improved BM25 Retrieval. Conference on Information Knowledge Management (CIKM). Hong Kong, Novembro 2009.

[33] DE MOURA E. S., FERNANDES D., RIBEIRO-NETO B., DA SILVA A. S., GONÇALVES M. A., Using Structural Information to Improve Search in Web Collections. Journal of the American Society for Information Science and Technology. p. 2503-2513, USA, Dezembro, 2010.

[34] FERNANDES D., DE MOURA E. S., DA SILVA A. S., RIBEIRO-NETO B., & BRAGA E. A Site Oriented Method for Segmenting Web Pages. In Proceedings of the 34th international ACM SIGIR 2011 conference on Research and development in Information Retrieval. p. 215-224, Beijing, Julho, 2011.