ANÍSIO MENDES LACERDA

USO DE PROGRAMAÇÃO GENÉTICA PARA PROPAGANDA DIRECIONADA BASEADA EM CONTEÚDO

Belo Horizonte
07 de março de 2008

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

USO DE PROGRAMAÇÃO GENÉTICA PARA PROPAGANDA DIRECIONADA BASEADA EM CONTEÚDO

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ANÍSIO MENDES LACERDA

Belo Horizonte
07 de março de 2008

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Uso de Programação Genética Para Propaganda Direcionada Baseada em Conteúdo

ANÍSIO MENDES LACERDA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. Nivio Ziviani – Orientador
Universidade Federal de Minas Gerais

Ph. D. Marcos André Gonçalves – Co-orientador
Universidade Federal de Minas Gerais

Ph. D. Edleno Silva de Moura
Universidade Federal do Amazonas

Ph. D. Marco Antônio Pinheiro de Cristo
FUCAPI - Fundação Centro de Análise, Pesquisa e Inovação Tecnológica

Belo Horizonte, 07 de março de 2008

Resumo

A Internet é hoje uma das principais mídias de veiculação de anúncios, dentre outros motivos, devido a possibilidade de exposição global de produtos e serviços e aos baixos custos envolvidos nessa exposição. Com isso cada vez mais recursos são destinados a publicidade na rede. O principal tipo de anúncio na Internet é a publicidade de busca, na qual um dado anunciante obtém uma posição de destaque na lista de propagandas mediante um valor pago. Segundo previsões do *eMarketer*, o crescimento dos ganhos com publicidade de busca será da ordem de 148%, saltando de US$ 16.9Bi em 2006 para US$ 42Bi em 2011. O tipo mais popular de publicidade de busca é conhecido como *keyword-targeted advertising* (propaganda direcionada baseada em palavra-chave), no qual as propagandas exibidas para o usuário são escolhidas a partir dos termos de sua consulta. O sucesso da propaganda baseada em palavra-chave motivou o surgimento de outro tipo de anúncio conhecido como *content-targeted advertising* (propaganda direcionada baseada em conteúdo). Neste caso, a lista de propagandas é determinada a partir do texto da página na qual as propagandas serão exibidas.

Neste trabalho propomos e testamos um novo arcabouço baseado em programação genética para o problema de propaganda direcionada baseada em conteúdo. Diferentemente de trabalhos anteriores nosso método permite combinar as evidências estatísticas disponíveis para melhorar a sugestão de propagandas para páginas da Web. Assim, nosso trabalho propõe uma nova técnica de sugestão de propagandas levando em consideração uma grande quantidade de evidências disponíveis o que leva a resultados superiores ao estado-da-arte na área.

Para validar o arcabouço proposto utilizamos uma coleção real de páginas e uma coleção real de propagandas. Os resultados experimentais mostram que o arcabouço baseado em programação genética superou em mais de 60% o melhor método da literatura em termos de precisão média. Além disso, programação genética mostrou-se bastante eficaz em evitar a sugestão de propagandas não-relevantes em páginas da Web. Esse fato é muito importante dado o impacto negativo da sugestão de propagandas não-relevantes para os usuários. Por fim, realizamos uma análise das árvores que representam os indivíduos gerados utilizando programação genética, concluindo que existe grande variabilidade entre os melhores indivíduos em termos da estrutura das árvores. Além disso, vimos que apenas uma pequena parcela das evidências disponíveis foi utilizada pelos melhores indivíduos encontrados.

Abstract

Internet has become one of the most important media for advertising nowadays. It represents the possibility of global exposure to large audiences at very low cost, which attracts great amounts in investments in advertising. In search advertising, an advertiser company is given prominent positioning in ad lists in return for a placement fee. Because of this, such methods are called *paid placement strategies*. According to *eMarketer*'s predictions the search advertising market will grow from US$ 16.9Bi in 2006 to US$42Bi in 2011. The most popular paid placement strategy is a non-intrusive technique called keyword-targeted advertising. In this technique, keywords extracted from the user's search query are matched against keywords associated with ads provided by advertisers. The success of keyword-targeted advertising has motivated information gatekeepers to offer their ad services in different contexts. We refer to the problem of matching ads to a web page that is browsed as *content-targeted advertising*.

In this work, we propose and test a new approach to content-targeted advertising based on genetic programming. Previous work in the literature did not answer important questions such as how to combine the available pieces of evidence or how much importance should be given to each evidence. So, we design a ranking strategy for displaying ads according to their relevance by effectively leveraging all the available evidence.

To validate our genetic programming method we performed experiments using a real ad collection and web pages extracted from a Brazilian newspaper. The results obtained show that our genetic programming approach provided gains over a state-of-the-art method of approximately 60% in average precision. Further, the genetic programming was able to learn functions that successfully avoid the placement of irrelevant ads by calculating thresholds based on the page where the ads should be placed. This is very important because of the negative impact of irrelevant ads on credibility and brand of publishers and advertisers. Finally, we perform an extensive and comprehensive analysis of genetic programming individuals in order better understand the results. We realize that there is a great variability between the best genetic programming individuals besides the similarity on the performance of the best individuals. Further, our best genetic programming individuals used only part of all available evidences.

À minha família, meu pai, Sebastião, minha mãe, Lilia e minha irmã, Pollyana, as pessoas mais importantes da minha vida.

Aos amigos, que sempre estiveram do meu lado.

...

Agradecimentos

Em primeiro lugar, agradeço à minha família, meu pai, minha mãe e minha irmã. Devo a eles tudo e quero lhes ser grato por todo amor, apoio incondicional e sacrifício.

Aos amigos de todas as horas, Bárbara, Louback, Marco Cristo e Mário Sérgio.

Aos companheiros e amigos de república, Modesto e Gustavo, por toda ajuda e companheirismo.

Aos amigos de longe, Allan, Aguillar, Gustavo e Lucas.

À Lara, pelo apoio, atenção e carinho, nos mais diversos momentos.

Aos amigos do LATIN, Charles, Claudine, Daniel, Davi, Denílson, Guilherme, Hendrickson, Pável, Wesley e Wladmir, com os quais aprendi muito. Especialmente, ao Fabiano e ao Thierson, que me acompanharam desde o início.

Ao pessoal da secretaria do departamento, em especial à Sônia, pela torcida e por sempre tentar ajudar nos momentos difíceis.

A todos os amigos do departamento de Ciência da Computação da Universidade Federal de Minas Gerais, em especial, ao Guidoni, pelo sincero interesse.

Obrigado aos professores que me orientaram e muito me ensinaram, Prof. Nivio Ziviani e Prof. Marcos André Gonçalves. Ao Prof. Nivio, em especial, por toda preocupação e ajuda na superação dos desafios, tanto técnicos quanto pessoais. Também agradeço aos membros da banca que muito contribuiram para este trabalho, Prof. Edleno de Silva de Moura e Prof. Marco Antônio Pinheiro de Cristo.

A todos vocês, novamente, muito obrigado, pois sei que sem sua ajuda nada disto teria acontecido.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Internet está se tornando uma das mais importantes mídias para anúncios hoje em dia. Entre os principais fatores deste sucesso podemos destacar a possibilidade de exposição global de serviços e produtos para um grande público a custos relativamente baixos. Com isso passamos a assistir um crescimento acelerado na adoção de publicidade de busca que, por sua vez, tem atraído grandes somas de dinheiro e permitido o financiamento de conteúdos e serviços na Internet. Porém, tal situação era bastante diferente bem pouco tempo atrás, quando a quebra de várias empresas que operavam na Web levou a uma diminuição no volume de capitais de risco e a uma considerável redução dos investimentos em publicidade [34, 35]. De acordo com o *Internet Advertising Bureau* (IAB) [22] tal redução causou consecutivos declínios nos rendimentos quadrimestrais das empresas no mercado americano. Tal queda iniciou-se no primeiro quadrimestre de 2001 e se repetiu até o fim de 2002.

A recuperação do mercado coincidiu com a adoção de um novo formato de propaganda na web, a propaganda de busca. Hoje em dia tal formato é líder na web e deve ser responsável por uma receita de aproximadamente 42 bilhões de dólares em 2011, segundo o eMarketer [11]. No gráfico da Figura 1.1 apresentamos os gastos anuais com propaganda na Internet. Esses dados referem-se ao mercado dos Estados Unidos. Podemos ver uma previsão de crescimento de 148% para o ano de 2011 (US$ 42Bi) em relação ao ano de 2006 (US$ 16.9Bi). Na Tabela 1.1 apresentamos, além dos dados utilizados no gráfico citado anteriormente, os gastos absolutos com mídia no mercado norte-americano e o percentual desses gastos realizados com mídia na Internet. Conforme podemos notar a cada ano o percentual dos gastos com mídia referentes a Internet tem aumentado consideravelmente. Além disso, há previsões de que a diversificação e a introdução de novos serviços permitirão um aumento da influência da publicidade de busca. Somente avanços tanto comerciais quanto tecnológicos permitirão a exploração eficaz de tais possibilidades.

Conforme citamos anteriormente, publicidade de busca é o principal tipo de mídia na Internet atualmente. Na Tabela 1.2 podemos ver que essa liderança ocorre desde o ano de 2006 e as previsões dizem que a publicidade de busca continuará sendo o principal tipo de anúncio na Internet alcançando 60% do mercado em 2011.

Em publicidade de busca, a um dado anunciante é dada uma posição de destaque na lista de propagandas mediante um valor pago. Por esta razão tais estratégias são conhecidas como estratégias de colocação pagas (*paid placement strategies*). A estratégia de colocação de propagandas mais conhecida é a propaganda direcionada baseada em palavras-chave (*keyword-targeted advertising*) [35]. Nesta técnica, as palavras utilizadas na consulta dos usuários são casadas com as que foram associadas às propagandas pelos anunciantes. A partir do casamento, uma lista ordenada de propagandas é construída de tal forma que, ou relevância das propagandas e a quantia que cada anunciante está disposto determinam que propagandas são exibidas e em que ordem. Nesta estratégia, as propagandas são apresentadas, em geral, do lado das respostas para a consulta do usuário.

O sucesso da propaganda direcionada baseada em palavras-chave motivou a oferta de anúncios em diferentes contextos. Podemos citar como exemplo a exposição de propagandas em páginas de portais de informação. A motivação é tirar vantagem do interesse informacional

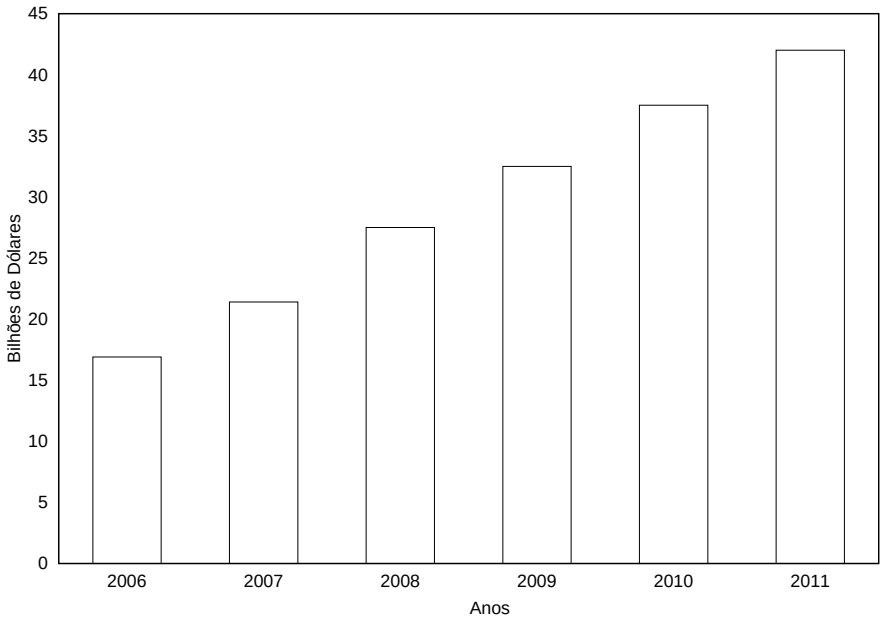


Figura 1.1: Gastos com propaganda na Internet. Dados referentes ao mercado norte-americano.

Ano	Gastos com mídia		
	Internet	Total	Percentual da Internet sobre o total gasto
2006	16,9	281,6	6,0%
2007	21,4	287,5	7,4%
2008	27,5	295,5	9,3%
2009	32,5	301,5	10,8%
2010	37,5	309,0	12,1%
2011	42,0	316,0	13,3%

Tabela 1.1: Gastos com mídia na Internet e gastos totais para o período compreendido entre os anos de 2006 e 2011, inclusive. Os valores para o intervalo de tempo a partir do ano de 2008 são projeções. Os dados referem-se ao mercado norte-americano e os valores absolutos estão expressos em bilhões de dólares.

imediato dos usuários enquanto eles navegam. Tal problema de realizar o casamento de propagandas e páginas Web é conhecido na literatura como publicidade direcionada baseada em conteúdo (*content-targeted advertising*) [26]. Cabe ressaltar que esta estratégia é diferente da anterior no sentido em que, ao invés de utilizarmos as palavras-chave fornecidas pelo usuário no momento da consulta, passamos a utilizar o conteúdo da página na qual tais propagandas serão apresentadas.

Neste trabalho, investigamos o uso de técnicas de programação genética (PG) para melhorar a relevância em publicidade de busca, especificamente publicidade direcionada baseada em conteúdo, isto é, propaganda associada ao conteúdo de páginas Web. Um trabalho anterior da literatura [32] mostrou que o uso de diferentes fontes de evidências,

Tipo de anúncio	2006	2007	2008	2009	2010	2011
Busca	6.799	8.624	11.000	12.935	14.906	16.590
Display ads	3.685	4.687	5.913	6.663	7.500	8.190
Classificados	3.059	3.638	4.675	5.493	6.281	6.930
Rich media	1.192	1.755	2.613	3.575	4.463	5.481
Lead generation	1.310	1.733	2.269	2.795	3.281	3.675
E-mail	338	428	481	553	600	630
Patrocínios	496	535	550	488	469	504
Total	**16.879**	**21.400**	**27.500**	**32.500**	**37.500**	**42.000**

Tabela 1.2: Gastos com mídia na Internet por tipo de anúncio. Os dados referem-se ao mercado norte-americano. Os valores estão expressos em milhões de dólares e são previsões a partir do ano de 2008. Dados obtidos a partir de [10].

tais como evidências estruturais e da página do anunciante, podem impactar a relevância das propagandas selecionadas. Entretanto, tal trabalho não respondeu a importantes questões, como: quais evidências utilizar e qual importância deve ser dada a cada evidência. Isto nos levou aos seguintes questionamentos: como podemos desenvolver uma estratégia de ordenação de propagandas de acordo com sua relevância que permita o melhor uso das evidências utilizadas? Além disso, dado o impacto negativo da sugestão de propagandas irrelevantes na credibilidade das marcas dos anunciantes e dos responsáveis pelas páginas onde elas são veiculadas, como podemos desenvolver funções que minimizem a sugestão de propagandas irrelevantes, especialmente quando não existem propagandas relevantes disponíveis?

1.1 Definição do Problema

O problema investigado neste trabalho é o da descoberta de funções de ordenação de propagandas que sejam mais eficazes que as funções conhecidas e, ao mesmo tempo, evitem a sugestão de propagandas irrelevantes. Apesar de ser conhecido na literatura [32] um trabalho que estuda o impacto de diferentes fontes de evidência para o casamento de propagandas e páginas da Web, ele não avalia como as fontes de evidência podem ser combinadas e como o problema da sugestão de propagandas irrelevantes pode ser tratado. Para tanto utilizamos tanto informação da estrutura das propagandas, como o título, a descrição e as palavras-chave, quanto informação das páginas dos anunciantes.

1.2 Objetivos

Dada a definição do problema apresentada na Seção 1.1, este trabalho levanta a seguinte questão de pesquisa: *"É possível sugerirmos funções de ordenação de propagandas mais eficazes que as funções conhecidas na literatura?"*.

Assim, a partir da hipótese de pesquisa, define-se o principal objetivo deste trabalho como sendo a proposição de uma abordagem para encontrar funções de ordenação que utilizem as

evidências disponíveis e evite a sugestão de propagandas irrelevantes. Em particular utilizamos uma estratégia baseada em PG [24].

Programação genética é uma técnica evolutiva, proposta por John Koza [23], inspirada na evolução biológica que permite encontrar soluções otimizadas para certas características do problema. Nossa suposição é que PG é capaz de combinar as evidências disponíveis e encontrar boas funções em termos de relevância das propagandas. Uma descrição de PG é apresentada na Seção 2.2.

Utilizamos PG pois essa técnica é capaz de realizar busca em grandes espaços, fato que ocorre em nosso problema uma vez que temos literalmente infinitas funções de sugestão de propagandas dadas as evidências disponíveis. Além disso, PG é uma técnica flexível para a combinação não-linear das evidências, é multi-objetivo e permite trabalhar com características intrínsecas ao problema. Por fim, PG foi utilizada também devido ao sucesso de seu uso em trabalhos como deduplicação de dados [8], reconhecimento de imagens [7] e classificação de documentos [37].

1.3 Principais Resultados

Propomos, neste trabalho, e testamos um arcabouço para a descoberta de funções para a associação de propagandas e páginas Web. Nosso método, baseado em programação genética, encontra funções que combinem as evidências disponíveis e permitam evitar a sugestão de propagandas irrelevantes. Os principais resultados da dissertação foram publicados em [24]. Das contribuições teórica e experimentais desta dissertação podemos citar:

- Do ponto de vista teórico, foi proposto um novo arcabouço baseado em programação genética a fim de encontrarmos funções para o casamento de propagandas e páginas Web. Esse arcabouço é baseado na utilização de diferentes fontes de evidência tais como: evidências da estrutura da propaganda – título, descrição e palavras-chave – e evidências não-estruturais – o conteúdo textual da página do anunciante. O arcabouço proposto é detalhadamente explicado no Capítulo 3.

- Diversos experimentos foram realizados com o objetivo de atestar a efetividade da abordagem proposta em comparação com as melhores estratégias conhecidas. A análise dos experimentos permitiu-nos concluir que a abordagem proposta superou os melhores resultados conhecidos da literatura. Dessa forma, os resultados obtidos podem ser utilizados como método base para futuros trabalhos de descoberta de funções de ordenação de propagandas. Tanto os experimentos quanto a discussão dos mesmos são apresentados no Capítulo 4.

- No Capítulo 5 apresentamos detalhes da análise dos indivíduos gerados através de PG. O objetivo dessa análise é tentar entender os motivos que permitiram a maior eficácia dos indivíduos propostos por PG em relação aos melhores métodos conhecidos na literatura para a sugestão de propagandas baseando-se no conteúdo das páginas.

Para validarmos nosso método realizamos experimentos com uma coleção real de propagandas e páginas da Web extraídas de um jornal brasileiro. Os resultados obtidos mostram que PG foi capaz de combinar as diferentes fontes de evidência disponíveis e forneceu boas sugestões para as páginas utilizadas nos testes. Em particular, nossa melhor função forneceu um ganho de 61,7% sobre o estado da arte. A métrica de comparação foi precisão média (*average precision*). Além disso, PG foi capaz de encontrar funções que evitaram a sugestão de propagandas irrelevantes em grande parte dos casos.

1.4 Trabalhos Relacionados

O grande sucesso da publicidade na Web, observado atualmente, tem motivado a pesquisa em muitos tópicos relacionados à publicidade de busca. Exemplos de tais estudos incluem a comparação de estratégias de ordenação [4], a caracterização de tráfego falso a fim de detectar fraudes [12], a proposta de ferramentas para sugestão de palavras-chave [5] e o projeto e implementação de sistemas de publicidade direcionada em larga escala [1].

Em particular, o aspecto da relevância da estratégia de ordenação de propagandas tem atraído atenção. Isto não é surpreendente uma vez que muitos trabalhos de pesquisa em publicidade tem enfatizado a importância de associações relevantes para consumidores [30]. Por outro lado, propagandas não-relevantes podem fazer com que os usuários percam o interesse ao passo que propagandas relevantes têm maior probabilidade de serem acessadas [4]. Como resultado, alguns trabalhos tentam determinar como tirar vantagem das evidências disponíveis para melhorar o grau de relevância das propagandas sugeridas. Por exemplo, estudos de casamento de palavras-chave mostram que a natureza e tamanho das palavras-chave têm impacto na probabilidade de uma propaganda ser acessada [29].

A relevância também é o foco dos autores em [32] que propõem várias estratégias para a ordenação de propagandas em publicidade baseada em conteúdo. Essas estratégias levam em consideração o conteúdo das partes estruturais da propaganda e a informação adicional obtida de páginas da Web além da página-alvo. Exemplos são a página do anunciante ou páginas Web obtidas através de modelos probabilísticos. Os autores mostram que considerar o conteúdo da parte estrutural e as páginas externas pode aumentar a chance de propagandas relevantes. Diferentemente deste trabalho, propomos *aprender* as melhores estratégias de ordenação a fim de efetivamente utilizar todas as evidências disponíveis enquanto minimizamos a sugestão de propagandas não-relevantes. Para isto utilizamos PG.

PG tem sido aplicada em vários tópicos de RI, tais como indução de consultas, representação e otimização [6, 21, 27], agrupamento e classificação de documentos [18, 36] e ordenação de documentos [17, 31]. A partir destes, muitos trabalhos [14, 15, 16, 13] têm aplicado PG para descobrir funções de ordenação. Por exemplo, ganhos foram obtidos na aplicação de PG com o objetivo de encontrar funções de ordenação para consultas específicas no roteamento de consultas [14]. De forma similar, PG também foi utilizada com sucesso em recuperação *ad-hoc* [16]. De fato, nosso trabalho é inspirado em pesquisa anterior na descoberta de funções de ordenação, porém, difere significativamente em vários aspectos

importantes. Dado que prentendemos encontrar funções de ordenação para publicidade direcionada baseada em conteúdo, lidamos com características específicas deste problema não encontradas em problemas clássicos de RI anteriormente estudados. Por exemplo, publicidade baseada em conteúdo apresenta diferentes tipos de evidências, a possibilidade de utilização de estatísticas de campanhas e características específicas, tais como restrições de sugestão de propagandas de campanhas diferentes e o impacto de propagandas não-relevantes.

1.5 Organização da Dissertação

Essa dissertação é dividida como segue. No Capítulo 2 são introduzidos os conceitos básicos de Programação Genética. Tais conceitos são fundamentais para o entendimento deste trabalho. O Capítulo 3 mostra como o arcabouço de PG foi utilizado para tratar o problema de propaganda direcionada baseada em conteúdo. O Capítulo 4 mostra os experimentos realizados para mensurar a qualidade das funções encontradas. No Capítulo 5 apresentamos a interpretação dos resultados produzidos por PG. Finalmente, no Capítulo 6 apresentamos as conclusões e os trabalhos futuros que poderão ser desenvolvidos a partir dos resultados desta dissertação.

Capítulo 2

Conceitos Básicos

O objetivo deste capítulo é apresentar os conceitos fundamentais para o melhor entendimento do problema tratado nessa dissertação: propaganda direcionada baseada em conteúdo. Além disso, apresentamos os conceitos básicos do arcabouço proposto, com ênfase nos fundamentos da teoria de programação genética.

2.1 O Problema: Propaganda Direcionada Baseada em Conteúdo

Propaganda direcionada baseada em conteúdo consiste em apresentar uma lista de propagandas em uma página Web, denominada *página-alvo*. É esperado que as propagandas associadas sejam relevantes para os usuários e adequadas e rentáveis para os anunciantes e divulgadores. Logo, os fatores que contribuem para a ordem na qual as propagandas são mostradas são: (i) a relação e adequação das propagandas ao conteúdo da página e (ii) a quantia que o anunciante está disposto a pagar pelos acessos a suas propagandas.

Neste trabalho consideramos que uma propaganda é composta de três partes estruturais: um título, uma descrição e um apontador. De fato, estes são os componentes de propagandas comumente encontrados em sistemas de propagandas comerciais. O apontador aponta para a página, chamada *landing page*, onde a transação pode ser iniciada. Nessa página, o usuário pode também encontrar mais informação relacionada à propaganda ou empresa, seus produtos e serviços. A Figura 2.1 mostra uma lista de propagandas, com duas propagandas, ao lado direito da página. No caso da propaganda no espaço superior, o título é "RFID Alternative", a descrição é "Single contact 1-Wire memory with 64-bit unique serial number.", e o apontador (*hyperlink*) é direcionado para a url "www.maxim-ic.com".

Além das partes visíveis, um conjunto de palavras-chave $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$ é associado a cada propaganda. Uma palavra-chave pode ser compostas de uma ou mais palavras e são utilizadas pelos anunciantes para descrever os tópicos que devem existir na página Web na qual tal propaganda pode aparecer. Por exemplo, para a propaganda na parte superior da Figura 2.1 as palavras-chave poderiam ser: "RFID" ou "RFID Alternative".

Para associar uma certa palavra-chave k e uma de suas propagandas, o anunciante precisa fazer uma oferta (um lance) para k em um sistema do tipo leilão. Quanto maior a oferta que o anunciante fizer pela palavra-chave k, maiores são as chances de que sua propaganda seja mostrada na lista de propagandas de páginas nas quais o tópico k está presente. Note que os anunciantes pagarão somente pelas ofertas quando os usuários acessarem suas propagandas. Além disso, um anunciante pode associar várias propagandas ao mesmo produto ou serviço. Tal grupo de propagandas é conhecido como *campanha*. Somente uma propaganda por campanha pode ser atribuída a uma página a fim de garantirmos um uso justo do espaço destinado à publicidade e aumentarmos a probabilidade que o usuário encontre uma propaganda interessante.

Neste trabalho estamos particularmente interessados no aspecto da relevância quando tratamos o problema da publicidade direcionada baseada em conteúdo.

Figura 2.1: Exemplo de propaganda baseada em conteúdo na página de uma empresa que oferece empregos na área de saúde. O conteúdo da página é um texto sobre uma tecnologia de identificação chamada RFID. No lado direito da páginas podemos ver propagandas relacionadas a este conteúdo propostas pelo sistema de propagandas do Google.

2.2 Programação Genética

Programação genética (PG) [23] é um conjunto de algoritmos de Inteligência Artificial que segue os princípios de herança genética e evolução das espécies. PG é normalmente utilizada para aproximar relacionamentos de funções não-lineares complexas [23]. Devido ao intrínseco mecanismo de busca paralelo e à capacidade de explorar o espaço multi-dimensional de forma bastante eficaz, PG tem sido utilizada para resolver uma grande variedade de problemas de otimização que frequentemente não possuem boas soluções conhecidas. Uma visão geral do arcabouço de PG, dados os conjuntos de teste e validação, é apresentado em 2.1.

Programa 2.1: Visão Geral do Arcabouço de PG.

```
1  Seja 𝒯 a coleção de documentos de treino;
2  Seja 𝒱 a coleção de documentos de validação;
3  Seja N_g o número de gerações;
4  Seja N_t o número de indivíduos;
5  𝒮 ← ∅;
6  𝒫 ← População inicial de indivíduos gerada de forma aleatória;
7  Para cada geração g de N_g gerações Faça {
8      Para cada indivíduo i ∈ 𝒫 Faça
9          fitness_i ← fitness(i, 𝒯);
10     𝒮_g ← Pegue N_t melhores indivíduos da geração g de acordo com seu valor de adaptação;
11     𝒮 ← 𝒮 ∪ 𝒮_g;
12     𝒫 ← Nova população criada aplicando-se os operadores genéticos aos indivíduos em 𝒮_g;
13 }
14 ℱ ← ∅;
15 Para cada indivíduo i ∈ 𝒮 Faça
16     ℱ ← ℱ ∪ {i, fitness(i, 𝒱)};
17 MelhorIndividuo ← MetodoDeSelecao(ℱ, 𝒮);
```

Em PG, a solução para um dado problema é representada como um indivíduo (isto é, um cromossomo) em uma população. Tais indivíduos são representados por estruturas de dados complexas como árvores, listas encadeadas e/ou pilhas [25]. O tamanho ou comprimento dessas estruturas não é fixo, embora valores máximos de tais medidas possam ser impostos. Inicialmente, um conjunto de indivíduos é criado de forma aleatória como podemos ver no Programa 2.1 (linha 6). A seguir, tais indivíduos evoluem geração após geração através dos operadores genéticos (linhas 7-13). Uma função de adaptação ($\mathit{fitness}(i, \mathcal{T})$) é usada para determinar um valor de adaptação para cada indivíduo (linha 9, a qual é detalhada no Programa 3.1). O valor de adaptação indica quão bem um indivíduo se comporta quanto testado no conjunto de treino e este valor pode ser utilizado para selecionar os melhores indivíduos (linha 10).

Todo o processo de evolução é altamente paralelo, localmente controlado e descentralizado. Além disso, o estado do processo de evolução depende somente da população atual. Durante o processo de evolução operadores genéticos são aplicados sobre o conjunto de melhores indivíduos com o objetivo de gerar indivíduos mais diversos e mais eficazes (linha 12). Como exemplos de tais operadores podemos citar reprodução, *crossover* e mutação. A seguir, detalhamos a função de cada operador:

- Operador de Crossover: A operação de crossover, ou reprodução sexual, permite a variação da população produzindo novos descendentes que consistem de partes tomadas de cada um dos pais. Ambos os pais são escolhidos utilizando-se o mesmo método de seleção.

- Operador de Mutação: A operação de mutação introduz mudanças aleatórias na estrutura da população. Assim tal operador simula os desvios que ocorrem no processo evolutivo. O operador de mutação atua sobre um indivíduo apenas e esse indivíduo é selecionado com uma probabilidade proporcional à sua adaptação ao problema. Ou

seja, quanto melhor a solução fornecida por um indivíduo maior a chance desse ser selecionado para sofrer uma mutação.

- Operador de Reprodução: A operação de reprodução consiste basicamente de duas etapas. Primeiro, um indivíduo é selecionado seguindo algum critério de seleção. Segundo, o indivíduo selecionado é copiado, sem alteração, da população atual para a população seguinte.

A última etapa do arcabouço apresentado no Programa 2.1 consiste em determinar o melhor indivíduo, o qual será aplicado sobre o conjunto de teste. A escolha natural é o indivíduo com melhor desempenho no conjunto de treino. Entretanto, tal indivíduo pode não ser genérico o suficiente devido a um problema conhecido como *overfitting*[1]. Com o objetivo de amenizar o problema, os melhores indivíduos que evoluiram no decorrer de N_g gerações são aplicados a uma segunda coleção, a qual recebe o nome de conjunto de validação (linha 15). Dessa forma é possível selecionar o indivíduo com melhor eficácia em ambos os conjuntos, o conjunto de treino e o conjunto de validação (linha 17) utilizando a função MetodoDeSelecao($\mathcal{F}$, $\mathcal{S}$). Pode-se esperar que tal indivíduo seja genérico uma vez que foi eficaz em dois conjuntos distintos de dados.

Logo, uma estratégia inicial para a seleção do melhor indivíduo pode ser escolher o indivíduo com melhor eficácia média nos conjuntos de treino e validação. Entretanto, uma vez que a média não garante que o indivíduo selecionado tenha uma eficácia balanceada em ambos conjuntos de dados, é interessante considerar o desvio padrão com o objetivo de evitar tal distorção.

De maneira mais formal, o seguinte método foi utilizado para a identificação do melhor indivíduo. Seja $\bar{f}_i$ a eficácia média do indivíduo i nos conjuntos de treino e de validação e $\sigma(f_i)$ o desvio padrão correspondente. O melhor indivíduo é dado por:

$$\underset{i}{argmax}(\bar{f}_i - \sigma(f_i)) \tag{2.1}$$

[1] Trata-se de uma situação na qual o indivíduo ajusta-se a características muito específicas do conjunto de treino de modo que a eficácia ainda melhora no conjunto de treino porém em conjuntos de dados não vistos a performance piora.

Capítulo 3

Uso de Programação Genética em Propaganda Direcionada Baseada em Conteúdo

Com o objetivo de aplicarmos PG ao problema de propaganda direcionada baseada em conteúdo, três conceitos importantes relacionados ao arcabouço de PG serão explicados neste capítulo: os indivíduos, os operadores genéticos e a função de adaptação.

3.1 Formalização do Problema

Dada uma coleção de páginas Web $\mathcal{D}$ e uma coleção de propagandas $\mathcal{A}$, nossa tarefa é selecionar propagandas $a_i \in \mathcal{A}$ relacionadas ao conteúdo de uma página $p \in \mathcal{D}$ e ordená-las de acordo com a relevância em relação às páginas. A lista de propagandas é então construída de maneira que mais propagandas relevantes são colocadas nas posições iniciais e, sempre que possível, somente uma propaganda por campanha é selecionada. A seguir, definimos tal restrição formalmente.

Seja $\mathcal{C} = \{C_1, C_2, ..., C_n\}$ uma partição de $\mathcal{A}$ que representa o conjunto de campanhas $C_1, C_2, \ldots, C_n$. Seja $r(a_i, p)$: $\mathcal{A} \times \mathcal{D} \rightarrow \mathbb{R}$ uma função que indica o grau de relevância de uma propaganda a_i dada uma página-alvo p. Seja δ_{ijp}: $\mathbb{N} \times \mathcal{C} \times \mathcal{D} \rightarrow \mathbb{R}$ uma função que representa o valor de relevância da i-ésima propaganda da campanha C_j de acordo com a função r. Por exemplo, se a_s é a segunda propaganda, em relação ao topo da lista, de uma campanha C_5, $\delta_{25p} = 0.5$ indica que $r(a_s, p) = 0.5$. Estamos interessados em encontrar a função $rank(a_i, p)$: $\mathcal{A} \times \mathcal{D} \rightarrow \mathbb{R}$ que possa ser usada para construir a lista ordenada de propagandas que satisfaça a restrição:

$$\forall_{i,j,k|j\neq k} \ (\delta_{ijp} > 0 \wedge \delta_{(i+1)kp} > 0 \Rightarrow \delta_{ijp} > \delta_{(i+1)kp}) \tag{3.1}$$

Conforme mencionado anteriormente, sistemas de sugestão de propagandas devem minimizar a possibilidade de exibir propagandas não-relevantes. Tais situações de erro ocorrem particularmente em dois casos. Primeiro, apesar de a propaganda e a página estarem relacionadas ao mesmo assunto, a associação é inapropriada. Por exemplo, este é o caso da sugestão de propagandas em páginas sobre catástrofes, publicidade ilegal ou anti-ética. Segundo, uma sugestão inapropriada de propagandas ocorre quando a página-alvo é sobre um tópico para o qual é difícil encontrar propagandas relevantes. A fim de minimizar tais situações, especialmente a segunda, uma boa função de ordenação deve ser capaz de estimar relevância tal que seja possível distinguir níveis aceitáveis de relevância de níveis não-aceitáveis.

Neste trabalho prentendemos aprender funções de ordenação $rank(a_i, p)$ utilizando PG. Tais funções de ordenação são projetadas com o objetivo de maximizar a precisão e minimizar as situações de erro citadas anteriormente.

3.2 Indivíduos

Uma vez que estamos interessados em encontrar uma boa função de ranking para associarmos propagandas a páginas Web, conforme descrito no Capítulo 2, decidimos representar nossos indivíduos usando uma estrutura de árvore. Conforme observado pelos autores em [23], os

nós internos na árvore ("*", "+", "log" e "/") representam funções aplicadas aos terminais nos nós folha. As funções de adição (+), multiplicação (∗), divisão (/) e logaritmo (log) são usadas na representação dos indivíduos. Tais funções foram escolhidas por fornecerem significado a relações. Por exemplo, funções de casamento utilizadas em RI geralmente fazem uso das funções de adição e multiplicação para reforçar relações em diferentes graus, enquanto a função de divisão é utilizada em relacionamentos inversos e a função logaritmo é utilizada para a suavização de valores.

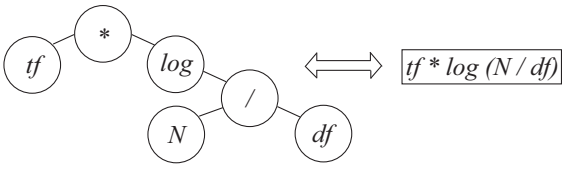


Figura 3.1: Exemplo de representação de uma função como uma árvore, no caso o esquema de pesos `TF-IDF`.

As funções mencionadas anteriormente são aplicadas aos terminais que são os nós folha na árvore. Por exemplo, temos na Figura 3.1 os seguintes terminais: "tf", "N" e "df". Uma vez que o foco deste trabalho é encontrar, utilizando PG, uma única função de ranking que permita encontrar um conjunto de propagandas relevantes em relação a uma determinada página combinando todas as evidências disponíveis ou um subconjunto destas evidências. Em outras palavras, os terminais representam estatísticas sobre as partes que formam a estrutura das propagandas e a informação fornecida pelos anunciantes tais como palavras-chave associadas com as propagandas e o conteúdo da página-alvo. Além disso, utilizamos números reais como terminais para permitir fatores de peso fixo.

Na Tabela 3.1 descrevemos todos os terminais disponíveis. Cabe notar que nesta tabela, P refere-se às partes que formam a estrutura das propagandas e à informação fornecida pelos anunciantes (palavra-chave, título, descrição e página-alvo) e G indica se as propagandas são agrupadas. Por exemplo, a evidência $tf_{ad,title}$ refere-se ao número de vezes que um termo aparece no título de uma propaganda enquanto a evidência $tf_{camp,title}$ representa o número de vezes que um termo aparece nos títulos das propagandas de uma campanha.

3.3 Operadores Genéticos

Os operadores genéticos usados em nosso modelo são largamente utilizados em PG, isto é, mutação, crossover e reprodução. A seguir, considerando a representação de nossos indivíduos por meio de árvores, exemplificamos cada um dos operadores:

- Crossover: O exemplo da Figura 3.2(a) consiste em, dadas duas árvores, trocar sub-árvores dessas árvores de forma a termos dois novos indivíduos.
- Mutação: O exemplo da Figura 3.2(b) ilustra o operador de mutação. Esse operador foi implementado de forma que uma sub-árvore selecionada de forma aleatória seja

Evidências Utilizadas	Significado Estatístico
$tf_{G,P}$	Número de vezes que um termo aparece na parte P de propagandas agrupadas por G.
$tf\_max_{G,P}$	Valor máximo de tf na parte P de propagandas agrupadas por G.
$tf\_avg_{G,P}$	Valor médio de tf na parte P de propagandas agrupadas por G.
$tf\_max\_col_{G,P}$	Valor máximo de $tf_{G,P}$ em toda a coleção.
$length_{G,P}$	Número de termos na parte P de propagandas agrupadas por G.
$n_{G,P}$	Número de termos distintos na parte P de propagandas agrupadas por G.
$df_{ad,P}$	Número de propagandas na coleção que o termo aparece na parte P.
$df\_max_{ad,P}$	Valor máximo de $df_{ad,P}$.
$df_{camp,P}$	Número de campanhas na coleção que o termo aparece na parte P.
$df\_max_{camp,P}$	Valor máximo de $df_{camp,P}$.
N_{ad}	Número de propagandas na coleção.
N_{camp}	Número de campanhas na coleção.
N	Constante real gerada de forma aleatória por GP.

Tabela 3.1: Terminais utilizados no arcabouço baseado em PG para PDBC.

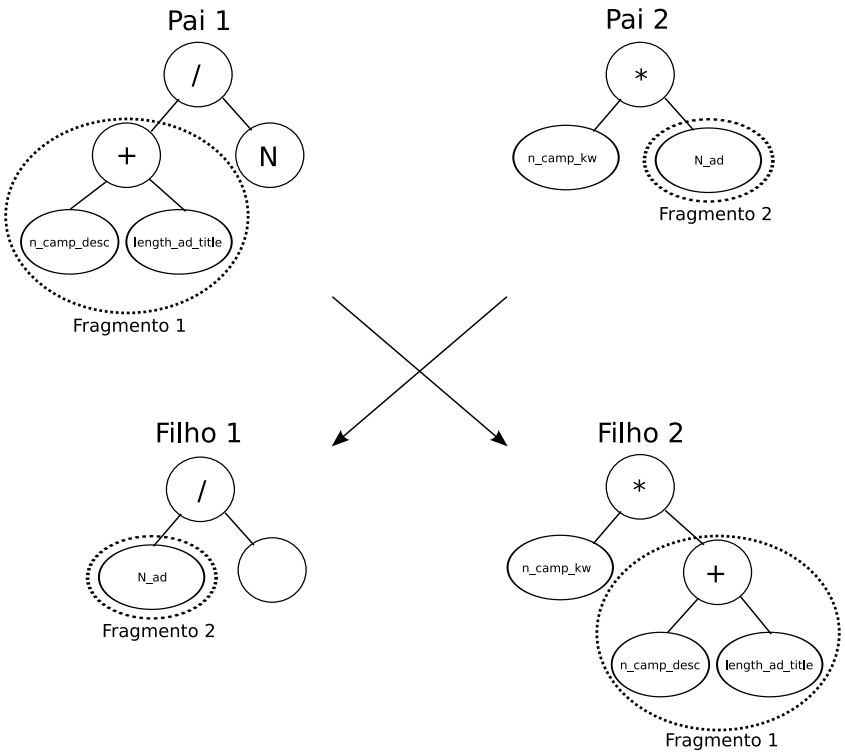
substituída por uma sub-árvore também criada de forma aleatória.

- Reprodução: O exemplo da Figura 3.2(c) ilustra operador de reprodução, o qual refere-se simplesmente a passagem de um dado indivíduo para a próxima população.

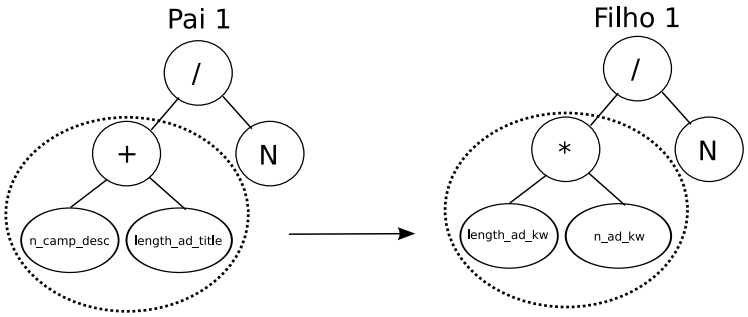
3.4 Função de Adaptação

A função de adaptação é a função objetivo que a PG procura otimizar. O algoritmo descrito no Programa 3.1 detalha a função de avaliação utilizada neste trabalho.

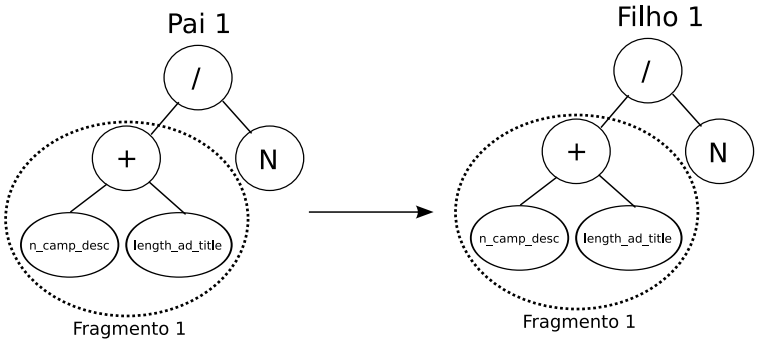
Devemos notar que as listas ordenadas produzidas pelos indivíduos aleatórios não satisfazem a restrição de campanha dada pela Equação 3.1. Logo, com o objetivo de que a função de adaptação (a qual corresponde à função de ordenação na Seção 2.1) possa satisfazer tal restrição, aplicamos o indivíduo i (o qual corresponde a função r na Seção 2.1) a cada campanha na coleção $\mathcal{T}$ utilizando uma estratégia *round-robin*, como descrito a seguir. Para cada campanha, uma ordenação é construída de acordo com a função de similaridade i (linhas 3-4). A propaganda no topo de cada ordenação é então selecionada até que todas as campanhas tenham sidos consideradas (linhas 6-9). Tais propagandas são ordenadas de acordo com o valor dado pela função de ordenação (isto é, pelo indivíduo) i e inseridas na ordenação final (linhas 10-11). O processo é repetido até que nenhuma propaganda permaneça não selecionada (linha 5). Dessa forma, garantimos que a j-ésima propaganda de uma campanha permanecerá sempre acima da *j+1*-ésima propaganda de qualquer outra campanha, satisfazendo a restrição de campanha. O valor de adaptação, correspondente ao indivíduo i, é então obtido através da avaliação da lista ordenada final (linha 12). Note que dependendo da função de avaliação a ser utilizada podemos sugerir diferentes funções de adaptação. A seguir, discutiremos as funções de avaliação e as funções de adapatação correspondentes utilizadas neste trabalho.



(a) Crossover.



(b) Mutação.



(c) Reprodução.

Figura 3.2: Operadores Genéticos.

Programa 3.1: Função de Adaptação.

```
1  function fitness(individuo i, colecao 𝒯)
2      Seja 𝒞 = {C_1, ..., C_n} o conjunto de campanhas em 𝒯;
3      Para toda campanha C_j ∈ 𝒞 faca
4          rlist_j ← Aplica i a C_j;
5      Enquanto existir j tal que |rlist_j| > 0 faca
6          Para j = 1 ate |𝒞| faca
7              Se |rlist_j| > 0 entao
8                  ad_top ← extrai propagandas do topo de rlist_j;
9                  Insira ad_top em rlist_temp;
10         Ordene rlist_temp;
11         Insira propagandas de rlist_temp em rlist_final preservando sua ordem;
12     fvalue ← Avalia rlist_final;
13     retorna fvalue;
```

Uma boa lista ordenada deve maximizar a colocação de propagandas relevantes próximo ao topo da lista uma vez que as propagandas que ocupam as posições do topo possuem uma maior probabilidade de serem clicadas pelos usuários [4]. Logo, a função de avaliação deverá levar em consideração o número de propagandas relevantes e a ordem na qual tais propagandas aparecem. Assim esta função deverá combinar precisão e revocação [2], duas medidas comumente utilizadas em RI. Um exemplo de tal função de avaliação é dada por:

$$pavg@k = \eta \sum_{i=1}^{k} \left(r(a_i) \times \left(\frac{\sum_{j=1}^{i} r(a_j)}{i} \right) \right) \tag{3.2}$$

onde $\eta = \frac{1}{k}$ é uma constante de normalização para garantir que $pavg@k$ seja um valor na faixa entre 0 e 1, k é o número de propagandas a serem apresentadas na página, a_i é a i-ésima propaganda da lista ordenada e $r(d) \in \{0, 1\}$ é o valor de relevância atribuído a uma propaganda, sendo 1 se a propaganda é relevante e 0 caso contrário. A informação de relevância é obtida a partir dos usuários.

Esta métrica é baseada na *precisão média não-interpolada* (PAVG), uma medida muito utilizada nas avaliações da TREC [19]. A diferença entre as métricas PAVG e $pavg@k$ é o valor da constante η, a qual é dada pelo inverso do total de documentos na coleção quando consideramos PAVG. Quando utilizamos $\eta = \frac{1}{k}$ garantimos que uma função de ordenação que apresenta propagandas relevantes em todas as posições de topo da lista receberá um valor $pavg@k$ igual a 1. Dessa forma, somos capazes de avaliar funções que sugerem um número de propagandas menor que o número de posições disponíveis. Neste trabalho, iremos nos referir à função de adaptação que usa $pavg@k$ para avaliar seus indivíduos como $f_{pavg@k}$.

Outro objetivo que desejamos alcançar através de nossas funções de adaptação é recompensar funções que minimizem a sugestão de propagandas irrelevantes. Como mencionado anteriormente, tais propagandas devem ser evitadas uma vez que contribuem para uma percepção negativa da marca e da credibilidade do anunciante por parte dos usuários. Uma possível solução para o problema é considerar os valores de ordenação fornecidos pelos indivíduos de PG como estimativas de quão relevantes as propagandas são em relação à página-alvo. Dessa forma, podemos determinar valores de corte para distinguir níveis de relevância

aceitáveis de níveis de relevância não-aceitáveis.

Logo, nosso problema é encontrar uma função de casamento que forneça estimativas confiáveis na faixa na qual o valor de corte possa ser determinado a fim de separar propagandas relevantes de propagandas não-relevantes. Nossa suposição é que PG é capaz de encontrar tais funções. Assim, dado um certo valor de corte t, iremos modificar nossa função de avaliação tal que ela recompense indivíduos que tendem a sugerir propagandas relevantes acima de t e propagandas não-relevantes abaixo de t. Dessa forma, tal função irá punir indivíduos que tendem a sugerir propagandas não-relevantes acima de t e propagandas relevantes abaixo de t. Nossa segunda métrica de avaliação é dada por:

$$pavg@k_t = \frac{1 + k_1\ r_{at} + k_2\ n_{bt}}{1 + k_3\ r_{bt} + k_4\ n_{at}}\ pavg@k, \tag{3.3}$$

onde k_1, k_3, k_2 e k_4 são os pesos associados ao número de propagandas relevantes acima (r_{at}) e abaixo (r_{bt}) do valor de corte e propagandas não-relevantes abaixo (n_{bt}) e acima (n_{at}) do valor de corte, respectivamente.

Note que em nossos experimentos damos mais peso a n_{at} uma vez que desejamos especialmente evitar a sugestão de propagandas não-relevantes nas posições superiores da ordenação. Em particular, utilizamos $k_1 = k_3 = k_2 = 1$ e $k_4 = 2$.

Resta agora detalhar o processo de cálculo do valor de corte t. Neste trabalho definimos $t = v_{min} + k_t\ (v_{max} - v_{min})$, onde v_{min} e v_{max} são os valores mínimo e máximo dados pela função de ordenação. A constante k_t é a posição relativa na faixa que o indivíduo de PG deve considerar como um ponto de baixa confiança. Em nossos experimentos utilizamos $k_t = 0.3$. Em outras palavras, nossas novas funções de adaptação irão recompensar funções de ordenação para as quais o valor mínimo associado a uma propaganda relevante corresponda a 30% de $(v_{max} - v_{min})$.

Note que, de fato, não é possível conhecer os valores de v_{min} e v_{max} pois estamos trabalhando com funções geradas de forma aleatória. Como consequência definimos tais limites pela inspeção dos valores da ordenação fornecidos por nossos indivíduos aleatórios. Neste estudo adotamos duas diferentes estratégias para estimar os valores limites. Na primeira estratégia utilizamos o valor máximo dado para uma certa página como v_{max} e o valor mínimo como v_{min}. Assim, temos diferentes valores de corte para diferentes páginas. A partir deste momento iremos nos referir a função de adaptação que usa $pavg@k_t$ para avaliar seus indivíduos e calcular valores de corte para cada página como f_{local}. Uma possível desvantangem de f_{local} é que tal estratégia tende a sugerir pelo menos uma propaganda por página. Na segunda estratégia utilizamos o valor máximo dado para um indivíduo como v_{max} e o valor mínimo como v_{min}. Neste caso temos somente um valor de corte para a função. Iremos nos referir a função de adaptação que usa $pavg@k_t$ para avaliar seus indivíduos e calcular os valores de corte para cada indivíduo como f_{global}. Ao contrário de f_{local}, f_{global} permite que nenhuma sugestão de propaganda seja feita para uma dada página.

Capítulo 4

Resultados Experimentais

Neste capítulo descrevemos a coleção de páginas da Web, a coleção de propagandas que utilizamos e apresentamos os principais resultados da dissertação.

4.1 Amostragem e Coleções

Com o intuito de avaliar o arcabouço proposto utilizamos uma coleção composta de 100 páginas extraídas de um jornal brasileiro[1]. Esse conjunto forma nossas páginas-alvo e foi coletado de forma que apenas o conteúdo dos artigos fosse preservado. Uma vez que não temos preferência por nenhum tópico em particular, as páginas tratam de diferentes assuntos como cultura, notícias locais, notícias internacionais, economia, esportes, política, agricultura, carros, crianças, imóveis, computadores e internet, TV, viagens e economia.

O conjunto de propagandas relevantes para nossas coleções de teste foi obtido utilizando a mesma estratégia de *pooling* da coleção TREC Web [20]. Em outras palavras, para cada uma das 100 páginas-alvo selecionamos as primeiras três propagandas da lista de propagandas fornecidas por cada um dos dez métodos de sugestão de propagandas proposto pelos autores em [32]. Estas propagandas foram obtidas a partir de uma coleção real composta por 93.972 propagandas agrupadas em 2.029 campanhas fornecidas por 1.744 anunciantes. Os anunciantes associaram 68.238 palavras-chave a tais propagandas.

Nesta coleção somente uma palavra-chave é associada a cada propaganda. Isto torna o conceito de campanhas muito importante. Os anunciantes utilizam campanhas para associar várias palavras-chave a um produto ou serviço. Como resultado do método de *pooling*, um total de 1.860 propagandas distintas foram selecionadas. Estas propagandas foram colocadas em conjuntos correspondendo a cada uma das páginas-alvo. Cada conjunto continha uma média de 15,81 propagandas. Todas as propagandas foram submetidas a uma avaliação feita por uma grupo de 15 especialistas (entre alunos de graduação e pós-graduação). A cada especialista foi pedido que avaliasse as propagandas selecionadas para cada página de acordo com sua relevância para as páginas. O número médio de propagandas relevantes por página foi 5,15.

Dado que nossos experimentos podem ser qualificados como uma tarefa de aprendizado supervisionado, seguimos o projeto de três conjuntos [14, 28]. Assim, todos os 2.337 pares de propaganda e páginas avaliados foram utilizados para a construção dos conjuntos de treino, validação e teste. Para isto, dividimos os pares avaliados em três conjuntos de forma aleatória. Utilizamos 50 páginas (e suas propagandas correspondentes) para treino, 30 páginas formaram o conjunto de validação e 20 páginas formaram o conjunto de teste. Como mencionado anteriormente, a introdução do conjunto de validação tem o objetivo de amenizar ou evitar o problema de *overfitting* (vide final da Seção 2.2) no conjunto de treino e selecionar o indíduo que melhor generalize. Dados os três conjuntos citados anteriormente, realizamos experimentos utilizando 3 diferentes distribuições de páginas entre esses 3 conjuntos. Em outras palavras, dividimos as 100 páginas da Web entre os conjuntos de treino, validação e

[1] http://www.estadao.com.br/

teste de 3 maneiras distintas. Todos os resultados reportados neste trabalho são baseados no conjunto de teste.

4.2 Parâmetros Iniciais

O aprendizado utilizando o conjunto de treinamento foi realizado com diferentes parâmetros. Notamos que um número pequeno para o tamanho da população e diferentes taxas para as operações genéticas produzem melhores resultados. 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. Este número foi determinado empiricamente. As sementes aleatórias utilizadas foram 245, 37.383, 322.443 e 6.758. Como proposto em [23] utilizamos as seguintes taxas para os operadores genéticos: 85% de crossover, 10% de mutação e 5% de reprodução. Testamos nosso arcabouço usando as três funções de adaptação descritas em 3.4. Realizamos experimentos para cada uma das funções quatro vezes utilizando as diferentes sementes. O melhor resultado entre as quatro execuções é reportado e utilizado para comparação.

4.3 Avaliação e Método Base

Apresentamos os resultados dos experimentos considerando que cada página-alvo oferece 3 posições para colocação de propagandas. Reportamos figuras utilizando $pavg@3$ (Eq. 3.2, com $k = 3$), para o caso no qual os métodos atribuiram exatamente três propagandas por página. Para os casos nos quais é permitido a atribuição de menos de três propagandas utilizamos $pavg@k$ (Eq. 3.2) and $pavg@k_t$ (Eq. 3.3). Em todos os casos, como em [32], reportamos o número de acertos e as propagandas sugeridas por posição. Chamamos de acertos a sugestão de uma propaganda relevante.

Comparamos os resultados de nosso arcabouço baseado em PG com os resultados do método `AAK_H` proposto em [32]. Esse método consiste em utilizar a função de similaridade do cosseno para casar a página-alvo e a propaganda. Além do título e da descrição, o conteúdo da propaganda, conforme utilizados por `AAK_H`, incluem o conteúdo da palavra-chave e o conteúdo da página do anunciante. O método também necessita que todos os termos da palavra-chave da propaganda estejam presentes na página-alvo para que a propaganda seja selecionada. Entre os métodos apresentados em [32], que levam em consideração somente o título, descrição, palavras-chave e página do anunciante, `AAK_H` é o mais eficaz. Dadas as evidências, note que, até onde é de nosso conhecimento, este é o melhor método encontrado na literatura. Isto torna `AAK_H` um método base ideal uma vez que nossos indivíduos fazem uso do mesmo conjunto de evidências.

4.4 Resultados

Nesta seção apresentamos os resultados de nossos experimentos com:

- exatamente três propagandas por página,
- com possivelmente menos de três propagandas por página.

4.4.1 Experimentos com exatamente três propagandas por página

Conforme podemos ver na Tabela 4.1, nosso melhor indivíduo (`GP1`) alcançou eficácia de $50,8\%$ em *pavg*@3. Isto corresponde a um ganho de $61,7\%$ quando comparamos com método base. Uma característica interessante de `GP1` é sua eficácia na primeira posição da lista de propagandas, que é a posição com a maior probabilidade de ser acessada pelos usuários [4].

Métodos	Acertos/Sugestões				*pavg*@3	
	#1	#2	#3	Total	Precisão Média	Ganho
`AAK_H`	9/20	5/20	9/20	23/60	0.314	–
`GP1`	14/20	11/20	7/20	32/60	0.508	+61.7%

Tabela 4.1: Comparação de eficácia entre o melhor indivíduo que evoluiu considerando a otimização de $f_{pavg@k}$ (`GP1`) e o método base (`AAK_H`). A colunas #1, #2, and #3 indicam o total de acertos e sugestões para a primeira, segunda e terceira posições da lista de propagandas, respectivamente.

Na Figura 4.1, apresentamos o processo de evolução ao longo de 30 gerações da população a partir da qual `GP1` foi selecionado. Para cada geração podemos ver os dez melhores indivíduos ordenados de acordo com sua eficácia em termos da função de adaptação (f_{local}). Podemos ver na Figura 4.1 uma diferença considerável entre a eficácia dos indivíduos nos conjuntos de treino e teste. Isto acontece devido ao *overfitting*. Os indivíduos submetidos ao conjunto de treinamento tendem a aprender características muito específicas não encontradas no conjunto de teste. Como consequência, os melhores indivíduos do conjunto de treino não se comportam tão bem no conjunto de teste. Entretanto, selecionando o melhor indivíduo utilizando a Equação 2.1, somos capazes de obter uma boa função de ordenação que é genérica o suficiente.

4.4.2 Experimentos com possivelmente menos de três propagandas por página

Na Tabela 4.2 comparamos a eficácia do melhor indivíduo obtido através de PG que evoluiu para evitar a sugestão de propagandas não-relevantes de acordo com valores de corte. Nesta tabela, `GP2` é o indivíduo que evoluiu a partir da otimização de f_{local}. A linha correspondente a este indivíduo mostra a eficácia para o caso em que o valor de corte não é levado em consideração. Isto é, todas as propagandas do topo da lista selecionadas por `GP2` são avaliadas independentemente de seus valores de ordenação. A linha que começa com `GP2+thr` corresponde ao mesmo indivíduo para o caso oposto, isto é, o valor de corte é levado em consideração. De forma similar, o indivíduo `GP3` evoluiu a partir da otimização de f_{global} e

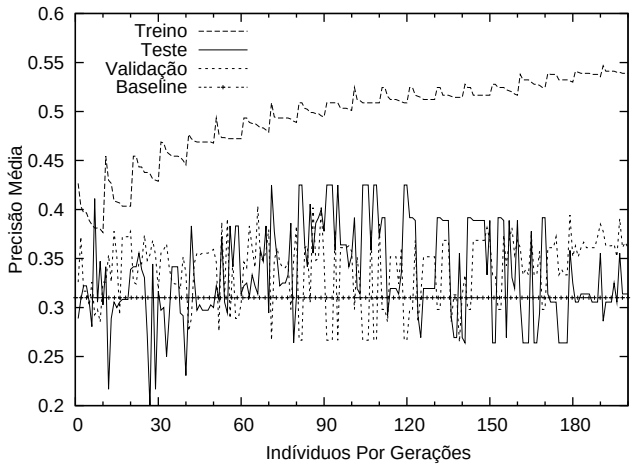


Figura 4.1: Processo de Evolução para 200 indivíduos ao decorrer de 20 gerações. Note que temos o valor de precisão média dos 10 melhores indivíduos de cada geração.

Métodos	Acertos/Sugestões				$pavg@3$		$pavg@k_t$	
	#1	#2	#3	Total	Precisão Média	Ganho	Precisão Média	Ganho
`AAK_H`	9/20	5/20	9/20	23/60	**0.31**	–	–	–
`GP2`	10/20	11/20	8/20	29/60	0.43	+38.7	**1.12**	–
`GP2+thr`	10/20	10/18	3/ 3	23/41	0.49	+58.1	1.30	+16.1
`GP3`	10/20	9/20	5/20	24/60	0.34	+9.6	**0.59**	–
`GP3+thr`	10/20	9/20	5/19	24/59	0.34	+9.6	0.59	0.0

Tabela 4.2: Eficácia dos melhores indivíduos que evoluiram a partir da otimização de f_{local} (`GP2`) e f_{global} (`GP3`). As colunas #1, #2, and #3 indicam o total de acertos e propagandas sugeridas para a primeira, segunda e terceira posições da lista de propagandas, respectivamente. Cabe notar que os valores nas colunas de ganho são relativos aos valores em negrito das colunas correspondentes à esquerda.

sua correspondente eficácia é mostrada para os casos nos quais os valores de corte são usados (`GP3+thr`) e os casos em que tais valores não são usados (`GP3`).

Note na Tabela 4.2 que `GP2` e `GP3` apresentam melhor eficácia que o método base com ganhos de $13,2\%$ e $9,6\%$, respectivamente, para $pavg@k$. Estes resultados, entretanto, são piores que os obtidos com nosso melhor indivíduo, `GP1`. Isto se deve em parte ao fato que indivíduos mais precisos tendem a errar a sugestão de propagandas menos frequentemente e, consequentemente, têm menos oportunidades de serem recompensados pela sugestão correta de propagandas não-relevantes abaixo de um certo valor de corte.

Quando analisamos a eficácia de indivíduos depois de aplicarmos os valores de corte, notamos uma melhora para `GP2+thr` e nenhuma diferença para `GP3+thr`. Por exemplo, o método `GP2+thr` foi capaz de evitar colocar doze propagandas não-relevantes na terceira posição com perda de somente cinco propagandas. Quando consideramos a métrica $pavg@k_t$, o ganho de `GP2+thr` sobre `GP2` foi de aproximadamente 16%. Isto permiti-nos concluir que PG foi capaz de aprender funções que evitam a sugestão de propagandas não-relevantes e apresenta boa eficácia para o caso no qual diferentes valores de corte são obtidos para cada

página. Para o caso no qual um valor de corte global foi utilizado, PG não foi capaz de aprender uma boa função de ordenação.

4.4.3 Variando tamanho do conjunto de treino

Neste conjunto de experimentos analisamos o impacto do tamanho do conjunto de treino sobre a eficácia dos indivíduos gerados. Em outras palavras, selecionamos diferentes porções do conjunto de treino e validação a fim de determinar qual o efeito das sugestões de propagandas feitas pelos indivíduos encontrados através de PG neste novo contexto.

Na Tabela 4.3 apresentamos, da esquerda para a direita, o percentual de páginas utilizadas no conjunto de treino e no conjunto de validação, o número absoluto de páginas utilizadas no conjunto de treino, o número absoluto de páginas utilizadas no conjunto de validação e o número absoluto de páginas utilizadas no conjunto de teste.

O número de páginas de cada conjunto (treino e validação) foi determinado em função do percentual a ser utilizado em relação ao número de páginas utilizadas nos experimentos descritos nas Seções 4.4.1 e 4.4.2. Nesses experimentos utilizamos 50 páginas no conjunto de treino, 30 páginas no conjunto de validação e 20 páginas no conjunto de teste. Logo, por exemplo, teremos para um percentual de 20%, 10 páginas para o conjunto de treino e 6 páginas para o conjunto de validação. Em todos os experimentos dessa seção não variamos o número de páginas do conjunto de teste.

Percentual de páginas utilizadas	Número de páginas utilizadas no treino	Número de páginas utilizadas na validação	Número de páginas utilizadas no teste
20	10	6	20
30	15	9	20
40	20	12	20
50	25	15	20
60	30	18	20
70	35	21	20
80	40	24	20
90	45	27	20

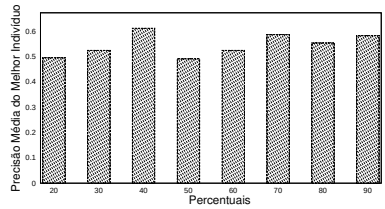
Tabela 4.3: Percentual de páginas utilizadas e respectivos números absolutos de páginas para cada conjunto: treino, validação e teste. Cabe ressaltar que não variamos a quantidade de páginas do conjunto de teste.

Nos gráficos das Figuras 4.2, 4.3 e 4.4 apresentamos os valores de precisão média do melhor indivíduo para cada uma das distribuições de documentos e para cada uma das sementes utilizadas. Para a escolha do melhor indivíduo utilizamos a Equação 2.1 como nos experimentos das Seções 4.4.1 e 4.4.2.

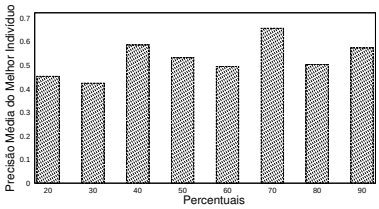
Em todos os gráficos apresentados nas Figuras 4.2, 4.3 e 4.4, independentemente da distribuição e das sementes utilizadas, podemos ver que um percentual menor ou igual a 30% sempre gera resultados piores, em termos de precisão média do melhor indivíduo, que utilizar um percentual igual a 90% dos documentos disponíveis. Assim, podemos notar que quanto maior a quantidade de exemplos para PG, maior a qualidade das funções encontradas.

Outro ponto bastante importante que notamos nos gráficos é que um número maior de páginas utilizadas nos conjuntos de treino e validação não implica em maior eficácia dos indivíduos. Essa tendência existe, porém a relação não é linear na maior parte das vezes. Por exemplo, no caso do gráfico da Figura 4.3d, temos que a precisão média do melhor indivíduo quando utilizamos apenas 40% das páginas disponíveis foi superior a todos os demais percentuais, ou seja, PG encontrou um indivíduo mais eficaz com 40% das páginas apenas.

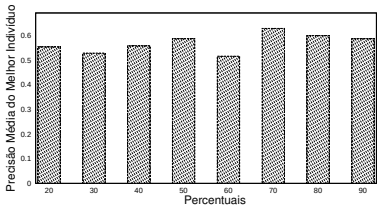
Um resultado muito interessante desta série de experimentos foi que encontramos funções ainda melhores que as encontradas nos experimentos da Seção 4.4.1. Por exemplo, no caso da Figura 4.4(b) PG encontrou uma função que forneceu 0.78 de precisão média utilizando apenas 70% dos documentos disponíveis.



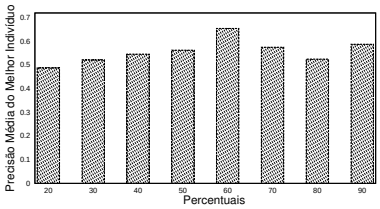
(a) Semente 245.



(b) Semente 322443.

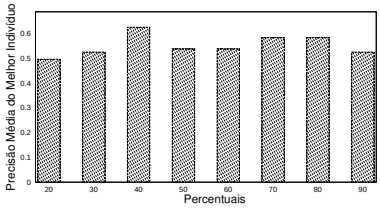


(c) Semente 37383.

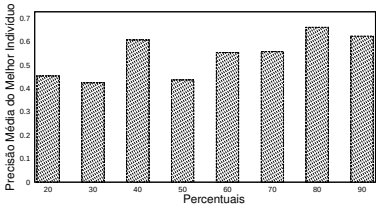


(d) Semente 6758.

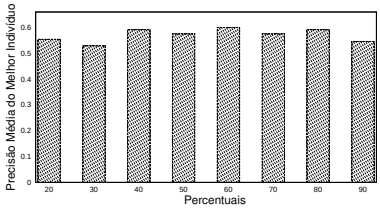
Figura 4.2: Distribuição D_1.



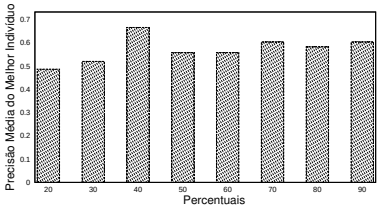
(a) Semente 245.



(b) Semente 322443.

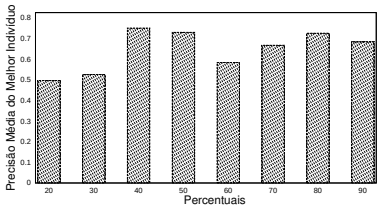


(c) Semente 37383.

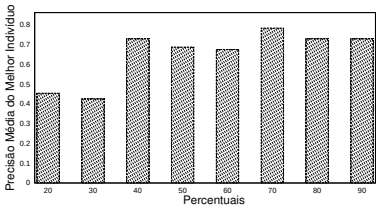


(d) Semente 6758.

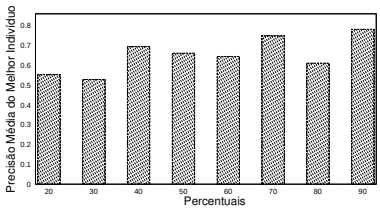
Figura 4.3: Distribuição D_2.



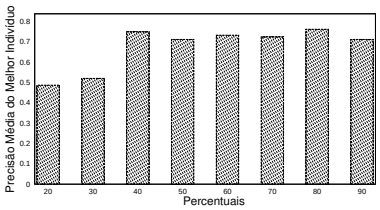
(a) Semente 245.



(b) Semente 322443.



(c) Semente 37383.



(d) Semente 6758.

Figura 4.4: Distribuição D_3.

Capítulo 5

Interpretação dos Resultados Gerados pela Programação Genética

Os resultados experimentais mostraram que é possível encontrar boas funções de ordenação capazes de sugerir propagandas relevantes e evitar propagandas não-relevantes utilizando PG. Neste capítulo procuramos caracterizar as funções encontradas determinando os fatores chave que contribuem para tal sucesso.

Uma das vantagens das soluções descobertas por PG em relação a outras técnicas tais como Redes Neurais [9] e SVM [33] é que tais soluções são interpretáveis. Em outras palavras, pode-se analisar os resultados e tentar entender o motivo para o sucesso. Entretanto, algumas vezes os resultados são muito complexos e de difícil interpretação, mesmo para especialistas no assunto. Um dos problemas conhecidos chama-se Problema do Inchaço – *The Bloat Problem* – em PG [3], o qual refere-se ao aparecimento de um número elevado de evidências. Por exemplo, na Figura 5.1 apresentamos uma árvore que contém 27 nós e, apesar de podermos notar algumas sub-árvores em comum, como *(* tf_camp_desc tf_max_camp_desc)* em negrito, é difícil compreender seu significado.

```
(log (+ (/ (/ (* df_camp_kw (/ n_camp_lp length_camp_kw))
(log (* (* tf_camp_desc tf_max_camp_desc)
(+ tf_max_ad_lp length_avg_desc)))) (* (log tf_avg_ad_title)
(/ tf_avg_ad_title n_camp_title)))
(* (* (+ (/ (* (log (* (* tf_camp_desc tf_max_camp_desc)
(/ df_max_camp_desc tf_ad_title)))
(/ (log tf_camp_kw) (log n_ad_kw))) (* (+ n_ad_kw n_camp_title)
(/ tf_avg_ad_title n_camp_title)))
(* (* (log tf_ad_desc) (log length_avg_kw)) n_ad_kw))
(log length_avg_kw)) (+ (+ n_ad_kw n_camp_title)
(log length_avg_kw)))))
```

Figura 5.1: Exemplo de árvore descoberta.

5.1 Análise Estatística das Populações

O primeiro passo para a análise estatística das populações foi selecionar indivíduos que sugeriram propagandas relevantes a maior parte das vezes. O objetivo era verificar se tais indivíduos eram muito diferentes entre si. Logo, precisamos de uma medida para determinar a distância entre árvores. A distância que adotamos foi a distância de edição, a qual é explicada a seguir.

5.1.1 Distância de Edição entre Árvores

As árvores obtidas através de PG são ordenadas e rotuladas. Em uma árvore um nó particular é referenciado como a raiz. Uma árvore é ordenada quando a ordem das sub-árvores é importante. Uma árvore é rotulada quando seus nós possuem rótulos.

Antes de discutirmos o algoritmo de distância de edição em árvores, iremos apresentar o algoritmo de distância de edição em cadeias de caracteres com o objetivo de evidenciar as diferenças entre esses algoritmos. A *distância de edição* entre duas cadeias de caracteres, s_1 e s_2, é definida como o número *mínimo* de *pontos de mutação* necessários para transformar s_1 em s_2, onde o ponto de mutação é uma das operações de inserção, remoção ou substituição.

As seguintes relações de recorrência definem a distância de edição, $d(s_1, s_2)$, de duas cadeias de caracteres s_1 e s_2:

$$
\begin{aligned}
d(\lambda, \lambda) &= 0, \text{ onde } \lambda \text{ significa cadeia de caracteres vazia} \\
d(s, \lambda) = d(\lambda, s) &= \mid s \mid, \text{ onde } \mid s \mid \text{ significa o tamanha da cadeia de caracteres } s \\
d(s_1, ch_1, s_2 + ch_2) &= \min \left(\begin{array}{l} d(s_1 + ch_1, s_2) + 1, \\ d(s_1, s_2 + ch_2) + 1, \\ d(s_1, s_2) + \left\{ \begin{array}{ll} 0 & \text{se } ch_1 = ch_2, \\ 1 & \text{caso contrário} \end{array} \right. \end{array} \right)
\end{aligned}
$$

As duas primeiras regras são óbvias, assim consideramos a última regra. Aqui, cada cadeia de caracteres tem um último caractere, ch_1 e ch_2, respectivamente. E precisamos editar $s_1 + ch_1$ em $s_2 + ch_2$. Se ch_1 é igual a ch_2, então ch_1 e ch_2 podem casar sem nenhuma penalidade, isto é, 0, e a distância de edição é $d(s_1, s_2)$. Se ch_1 é diferente de ch_2, então ch_1 deve ser transformado em ch_2, isto é, existe uma penalidade igual a 1, resultando em uma distância de edição igual a $d(s_1, s_2) + 1$. Outra opção para editar $s_1 + ch_1$ em $s_2 + ch_2$ é editar $s_1 + ch_1$ em s_2 e então inserir ch_2, resultando em uma distância de edição igual a $d(s_1 + ch_1, s_2) + 1$. A última possibilidade é apagar ch_1 e editar s_1 em $s_2 + ch_2$, resultando em uma distância de edição igual a $d(s_1, s_2 + ch_2) + 1$. Não existe nenhuma outra possibilidade. A fim de obter a distância de edição entre duas cadeias de caracteres, optaremos pela de menor custo, isto é, *min*, das três alternativas.

O Algoritmo 1 detalha como podemos utilizar programação dinâmica para calcular a distância de edição entre duas cadeias de caracteres. Aqui, uma matriz bi-dimensional $m[0...|s_1|, 0...|s_2|]$ é utilizada para lidar com os valores de distância de edição.

Algoritmo 1 Distância de edição entre cadeias de caracteres

Seja $m[0, 0] = 0$
Para $i = 1$ a $\mid s_1 \mid$ **Faça**
  $m[i, 0] = i$
Fim Para
Para $j = 1$ a $\mid s_2 \mid$ **Faça**
  $m[0, j] = j$
Fim Para
Para $i = 1$ a $\mid s_1 \mid$ **Faça**
  Para $j = 1$ a $\mid s_2 \mid$ **Faça**
    $m[i, j] = min(m[i-1, j-1] +$ se $ch_1 = ch_2$ então 0 senão $1, m[i-1, j] + 1, m[i, j-1] + 1)$
  Fim Para
Fim Para

O problema da distância de edição entre árvores é mais difícil que o problema da distância de edição entre cadeias de caracteres mencionado anteriormente. Para duas cadeias de caracteres s_1 e s_2, se $s_1[i] = s_2[j]$, então a distância entre $s_1[1...i]$ e $s_2[1...j]$ é a mesma que a distância entre $s_1[1...i-1]$ e $s_2[1...j-1]$. Entretanto, para calcularmos a distância de edição entre árvores, o relacionamento de ancestral deve ser levado em consideração e uma simplificação análoga não é possível. Os autores propõem em [38] um algoritmo baseado em programação dinâmica para a distância de edição entre árvores similar ao algoritmo de

programação dinâmica para a distância de edição entre cadeias de caracteres. De fato, o algoritmo de distância de edição entre cadeias de caracteres pode ser considerado um caso especial do algoritmo de distância de edição entre árvores. Este algoritmo funciona bem para nossas árvores produzidas por PG, as quais são rotuladas e ordenadas.

A distância entre duas árvores é definida como o número de operações de edição necessárias para transformar uma árvore em outra. As operações de edição podem ser inserção, remoção e substituição. A operação de inserção insere um novo nó na árvore. A operação de remoção remove um nó da árvore. A operação de substituição modifica o rótulo de um nó da árvore.

Para identificar os nós em uma árvore, utilizamos a numeração pós-ordem da esquerda para a direita, conforme ilustra a Figura 5.2. $T[i]$ representa o i-ésimo nó na árvore de acordo com a numeração pós-ordem.

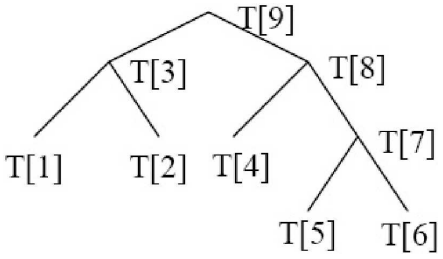


Figura 5.2: Numeração Pós-Ordem.

Um *mapeamento* entre duas árvores, T_1 e T_2, é uma especificação gráfica de quais operações de edição aplicam-se a cada nó das árvores. O mapeamento mostra como transformar a árvore T_1 na árvore T_2. Na Figura 5.3 mostramos um exemplo de mapeamento. Uma linha pontilhada a partir de $T_1[i]$ para $T_2[j]$ indica que uma operação de *substituição* precisa ser aplicada a fim de modificar $T_1[i]$ para $T_2[j]$, caso $T_1[i] \neq T_2[j]$, ou que $T_1[i]$ permanece inalterado se $T_1[i] = T_2[j]$. Os nós de T_1 quando não tocados pela linha pontilhada precisam ser *removidos* e os nós de T_2 não tocados pela linha pontilhada precisam ser *inseridos*. Para construir a sequência de operações de edição, simplesmente precisamos remover todos os nós indicados pelo mapeamento (isto é, todos os nós em T_1 não tocados pela linha pontilhada), a seguir todas as substituições, e finalmente todas as inserções. Neste exemplo, a sequência de operações de edição necessárias para transformar T_1 em T_2 é: remover o nó com rótulo $\surd$ (trata-se de um nó não tocado na árvore T_1) e inserir um nó com rótulo $\surd$ (trata-se de um nó não tocado na árvore T_2).

O mapeamento M de T_1 para T_2 tem um custo associado. Sejam I e J os conjuntos de nós em T_1 e T_2, respectivamente, não tocados por qualquer linha pontilhada em M, assim o custo de M pode ser definido como:

$$\text{Custo}(M) = \sum_{(i,j)\in M} \textit{substituição}\ (T_1[i], T_2[j]) + \sum_{i\in I} \textit{remoção}\ (T_1[i]) + \sum_{j\in J} \textit{inserção}\ (T_2[j])$$

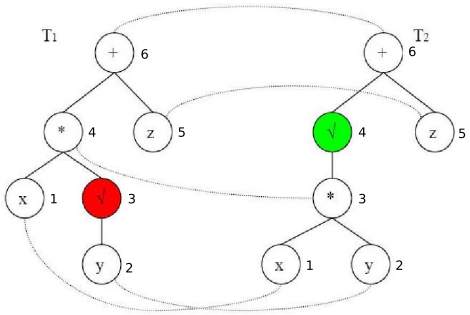


Figura 5.3: Mapeamento entre as árvores T_1 e T_2. Os índices ao lado dos nós referem-se a ordenação pós-ordem.

onde *substituição* é a operação que modifica o rótulo de $T_1[i]$ para $T_2[j]$ se $T_1[i] \neq T_2[j]$, *remoção* refere-se a operação de edição que remove o nó $T_1[i]$ em I e *inserção* refere-se a operação que insere o nó $T_2[j]$ em J.

Conforme mencionado anteriormente, utilizamos numeração pós-ordem dos nós das árvores. O valor de $l(i)$ é definido como o número que identifica a folha mais à esquerda da árvore cuja raiz é $T[i]$, onde $T[i]$ é o i-ésimo nó na árvore da esquerda para a direita considerando-se a numeração pós-ordem. Quando $T[i]$ é uma folha, $l(i) = i$. Nessa numeração, $T_1[1..i]$ e $T_2[1..j]$ serão florestas. Na Figura 5.4 apresentamos as florestas $T[1..7]$ (conjunto de árvores à direita) da árvore T (árvore à esquerda). Mais formalmente temos que $T[i..j]$ é definido como a floresta de T induzida pelos nós numerados de i a j inclusive. Se $i > j$, $T[i..j] = T[1..i]$ é referenciado como $forest(i)$. $T[l(i)..i]$ é referenciado como $tree(i)$. A definição de mapeamento para florestas ordenadas é idêntica à definição para árvores.

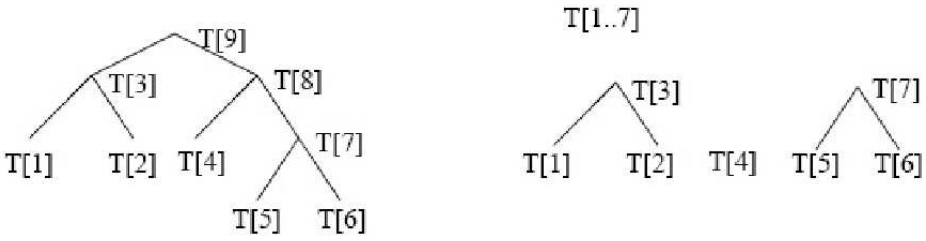


Figura 5.4: Florestas da árvore T.

A distância entre duas florestas $T_1[i'..i]$ e $T_2[j'..j]$ é denotada como $fdist(i'..i, j'..j)$. A distância entre a sub-árvore cuja raiz é $T_1[i]$ e a sub-árvore cuja raiz é $T_2[j]$ é denotada como $tdist(i, j)$. É óbvio que o valor de $tdist$ entre a árvore cuja raiz é a raiz de T_1 e a árvore cuja raiz é a raiz de T_2 nos dá a solução para nosso problema. Outra notação usada no algoritmo é o conjunto $LR\_keyroots$ para a árvore T, definido como segue:

$$LR\_keyroots(T) = k \mid \text{não existe } k' > k \text{ tal que } l(k) = l(k')$$

Intuitivamente, *LR_keyroots* é o conjunto consistindo da raiz e todos os nós com irmãos à esquerda. Por exemplo, para as árvores T_1 e T_2 na Figura 5.3, $LR\_keyroots(T_1) = 3, 5, 6$ e $LR\_keyroots(T_2) = 2, 5, 6$, respectivamente. Para encontramos o conjunto *LR_keyroots* da árvore T_1 da Figura 5.3 precisamos determinar os seguintes valores:

$$l(1) = l(1) = l(4) = l(4) = l(6) = l(6)$$
$$l(2) = l(2) = l(3) = l(3)$$
$$l(5) = l(5)$$

De acordo com a definição acima, temos que $LR\_keyroots(T_1) = 3, 5, 6$ pois o nó 6 é a raiz, fazendo com que faça parte do conjunto por definição. No caso do nó 3 temos que $l(3) = l(2)$, logo, como não existe $k' > k$ tal que $l(k') = l(k)$, onde $k' = 3$ e $k = 2$, o nó 3 faz parte do conjunto $LR\_keyroots(T_1)$. Enfim, para o nó 5 temos que também não existe $k' > k$ tal que $l(k') = l(k)$, onde $k' = 5$ e $k = 5$, fazendo com que o nó 5 faça parte do conjunto $LR\_keyroots(T_1)$. O raciocínio para determinarmos o conjunto $LR\_keyroots(T_2)$ é análogo.

O Algoritmo 1 computa a distância de edição entre árvores, onde a matriz custo c corresponde à matriz custo unitária. Isto é, as operações de inserção, remoção e substituição têm custo unitário. *LR_keyroots* é um vetor de valores lógicos (0 ou 1) onde um valor igual a 1 indica que um elemento está no conjunto. No algoritmo, a rotina principal *DistanciaDeEdicaoEntreArvores* realiza todo o processamento e inicializações e faz as chamadas necessárias para a rotina *treedist*. Os resultados são armazenados no vetor permanente *tdist* o qual no fim contém as distâncias entre todos os pares possíveis de sub-árvores e na posição $(|T_1|, |T_2|)$ a distância entre as duas raízes, isto é, a distância entre as árvores T_1 e T_2. A rotina *treedist* computa os valores de distância e os custos das operações de edição. A rotina armazena os resultados no vetor temporário *fdist* e também preenche os valores para as distâncias entre as sub-árvores no vetor *tdist*.

Usando o algoritmo para distância de edição entre árvores apresentado no Algoritmo 1, analisamos as árvores geradas por PG. Nas Tabelas 5.1(a), 5.1(b) e 5.1(c) apresentamos a distância de edição entre as melhores árvores de cada população. Assim, para cada distribuição de páginas utilizado como treino, validação e teste, temos 4 árvores, ou seja, uma árvore para cada semente testada. Já nas Tabelas 5.2, 5.3 e 5.4 selecionamos os 3 melhores indivíduos de cada população, logo, temos 12 indivíduos (3 indivíduos selecionados para cada uma das 4 sementes).

A partir das Tabelas 5.1, 5.2, 5.3 e 5.4 podemos concluir que, em geral, os melhores indivíduos gerados são bastante diferentes entre si. Este fato é bastante interessante pois diferentes combinações de evidências permitiram alcançar boa eficácia em termos de precisão média. No caso da Tabela 5.1(a) notamos que as árvores 0, 2 e 3 são relativamente parecidas, enquanto a árvore 1 é a árvore mais diferente dentro desta distribuição. Já no caso da Tabela 5.1(b) e 5.1(c) todas as funções são bastante diferentes entre si, o que implica em grande diversidade dos melhores indivíduos desta distribuição.

Algoritmo 2 Algoritmo de Distância de Edição entre Árvores

ENTRADA: Árvores T_1 e T_2
SAÍDA: Distância de edição entre as árvores T_1 e T_2

DistanciaDeEdicaoEntreArvores(T_1, T_2)
Para $i = 0$ a $\mid T \mid$ **Faça**
  Se $LT\_keyroots1[i]$ **Então**
    Para $j = 1$ a $\mid T_2 \mid$ **Faça**
      Se $LR\_keyroots1[j]$ **Então**
        $treedist(i, j, tdist, l_1, l_2, c)$;
      Fim Se
    Fim Para
  Fim Se
Fim Para
Retorna $tdist[i][j]$

$treedist(pos_1, pos_2, tdist, l_1, l_2, c)$ {retorna a distância entre as árvores cujas raízes são pos_1 e pos_2}

$fdist[0][0] = 0$;
Para $i = l_1[pos_1]$ a pos_1 **Faça**
  $fdist[i][0] = fdist[i-1][0] + c[T_1[i]][0]$; {Remoções}
Fim Para
Para $j = l_2[pos_2]$ a pos_2 **Faça**
  $fdist[0][j] = fdist[0][j-1] + c[0][T_2[j]]$; {Inserções}
Fim Para
Para $i = l_1[pos_1]$ a pos_1 **Faça**
  Para $j = l_2[pos_2]$ a pos_2 **Faça**
    Se $l_1[i] == l_1[pos_1]$ e $l_2[j] == l_2[pos_2]$ **Então**
      $fdist[i][j] = min(fdist[i-1][j] + c[T_1[i]][0], fdist[i][j-1] + c[0][T_2[j]]), fdist[i-1, j-1] + c[T_1[i]]T_2[j]])$;
      $tdist[i][j] = fdist[i][j]$;
    else
      $m = l_1[i] - l_1[pos_1]; n = l_2[j] - l_2[pos_2]$;
      $fdist[i][j] = min(fdist[i-1][j] + c[T_1[i]][0], fdist[i][j-1] + c[0][T_2[j]], fdist[m][n] + tdist[i][j])$;
    Fim Se
  Fim Para
Fim Para

5.2 Análise Estatística dos Indivíduos

Nesta seção apresentamos os resultados da contagem de frequências. Conforme citado na Seção 4.1, realizamos experimentos com diferentes distribuições de páginas. Iremos agora identificar cada distribuição através dos identificadores: D_1, D_2 e D_3.

5.2.1 Frequência do Conjunto de Evidências

Na Figura 5.5 apresentamos a frequência das evidências encontradas na melhor árvore de cada geração. Ou seja, selecionamos o indivíduo mais eficaz de cada geração para cada uma das sementes utilizadas na Seção 4.1, assim cada gráfico refere-se a 4 indivíduos uma vez que utilizamos 4 sementes distintas. O número de evidências no eixo x foi de 37, 47 e 47 para os gráficos das Figuras 5.5(a), 5.5(b) e 5.5(c), respectivamente. Assim, vemos

Número da Árvore	0	**1**	2	3
0	0			
1	**69**	**0**		
2	28	**66**	0	
3	29	**69**	23	0

(a) Distribuição D_1.

Número da Árvore	0	1	2	3
0	0			
1	72	0		
2	106	113	0	
3	59	86	110	0

(b) Distribuição D_2.

Número da Árvore	0	1	2	3
0	0			
1	130	0		
2	99	138	0	
3	53	125	92	0

(c) Distribuição D_3.

Tabela 5.1: Distância de edição entre árvores. Neste caso consideramos a melhor árvore, ou seja, a árvore mais eficaz.

Número da Árvore	0	1	2	3	4	5	6	7	8	9	10	11
0	0											
1	41	0										
2	41	0	0									
3	69	84	84	0								
4	73	88	88	5	0							
5	67	82	82	5	10	0						
6	28	61	61	66	69	64	0					
7	28	61	61	66	69	64	0	0				
8	28	61	61	66	69	64	0	0	0			
9	29	58	58	69	72	67	23	23	23	0		
10	29	63	63	72	75	70	25	25	25	10	0	
11	35	60	60	68	72	66	29	29	29	22	24	0

Tabela 5.2: Distância de edição entre árvores referente à distribuição D_1. Neste caso consideramos as 3 melhores árvores.

que o número de evidências apresentadas nos melhores indivíduos não variou muito quando variamos os conjuntos de treino, validação e teste. Isto nos permite intuir que a distribuição das páginas que utilizamos nos experimentos não foi tendenciosa. Outro aspecto interessante sobre o número de evidências encontradas nos melhores indivíduos foi o fato deste número ser pequeno em relação ao número de evidências disponíveis (72). Uma vez que o custo da computação do valor de similaridade entre uma página e uma propaganda está relacionado, entre outros fatores, ao número de evidências envolvidas na função de similariedade, um baixo número de evidências implica em funções mais eficientes.

Na Figura 5.6 apresentamos a contagem das frequências de cada distribuição de páginas quando selecionamos as 3 melhores árvores de cada população. O número de evidências no

Número da Árvore	0	1	2	3	4	5	6	7	8	9	10	11
0	0											
1	18	0										
2	0	18	0									
3	72	78	72	0								
4	72	78	72	0	0							
5	72	78	72	0	0	0						
6	106	100	106	113	113	113	0					
7	62	63	62	89	89	89	89	0				
8	114	108	114	127	127	127	110	53	0			
9	59	58	59	86	86	86	110	76	117	0		
10	59	58	59	86	86	86	110	76	117	0	0	
11	59	58	59	86	86	86	110	76	117	0	0	0

Tabela 5.3: Distância de edição entre árvores referente à distribuição D_2. Neste caso consideramos as 3 melhores árvores.

Número da Árvore	0	1	2	3	4	5	6	7	8	9	10	11
0	0											
1	0	0										
2	15	15	0									
3	130	130	130	0								
4	169	169	167	118	0							
5	169	169	167	118	0	0						
6	99	99	99	138	173	173	0					
7	153	153	153	171	198	198	122	0				
8	153	153	153	171	198	198	124	2	0			
9	53	53	60	125	174	174	92	152	152	0		
10	53	53	60	125	174	174	92	152	152	0	0	
11	53	53	59	127	175	175	94	155	155	19	19	0

Tabela 5.4: Distância de edição entre árvores referente à distribuição D_3. Neste caso consideramos as 3 melhores árvores.

eixo x é igual a 43 para a distribuição D_1, 48 para a distribuição D_2 e 56 para a distribuição D_3. Novamente o número de evidências presentes nos melhores indivíduos não difere muito quando utilizamos diferentes páginas nos conjuntos de treino, validação e testes. De forma similar ao visto nos gráficos da Figura 5.5, apenas uma fração das evidências disponíveis foi utilizada pelos melhores indivíduos.

A partir das evidências encontradas nos melhores indivíduos determinamos o tamanho do conjunto de interseção de evidências entre os melhores indivíduos. Em outras palavras, verificamos o número de evidências em comum entre as três distribuições utilizadas. Conforme vemos na Tabela 5.2.1 uma grande parcela das evidências disponíveis são comuns às diferentes distribuições.

Distribuição	1	2	3
1	0		
2	28	0	
3	25	33	0

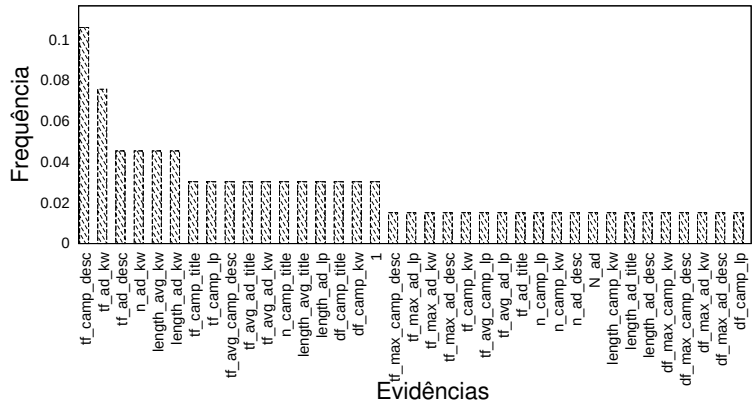
Tabela 5.5: Tamanho do Conjunto de Interseção de Evidências.

5.2.2 Frequência do Conjunto de Funções

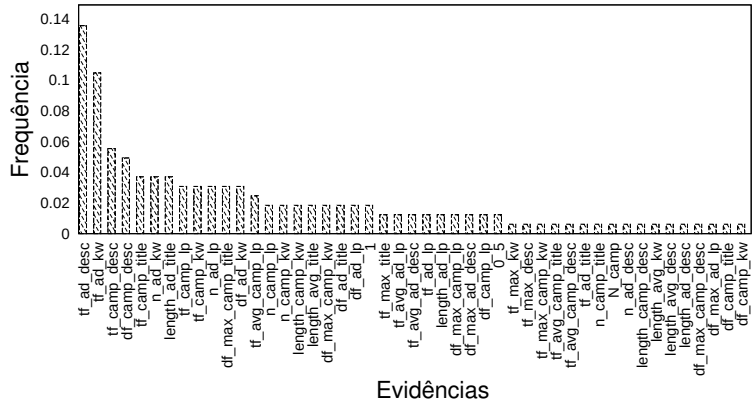
Conforme mencionado na Seção 3.2 as funções que operam sobre as evidências são: "*" (multiplicação), "+" (adição), "log" (logaritmo) e "/" (divisão). Na Figura 5.7 apresentamos as frequências das funções nos melhores indivíduos de cada população. Notamos que não existem grandes diferenças entre o percentual de ocorrência das funções entre as diferentes amostragens.

Na Figura 5.8 apresentamos o percentual de ocorrência das funções quando selecionamos as 3 melhores árvores de cada população. Notamos que o comportamento visto na Figura 5.7 também é observado nesse caso, isto é, não existe grande diferença em termos percentuais das funções encontradas nas diferentes distribuições de páginas.

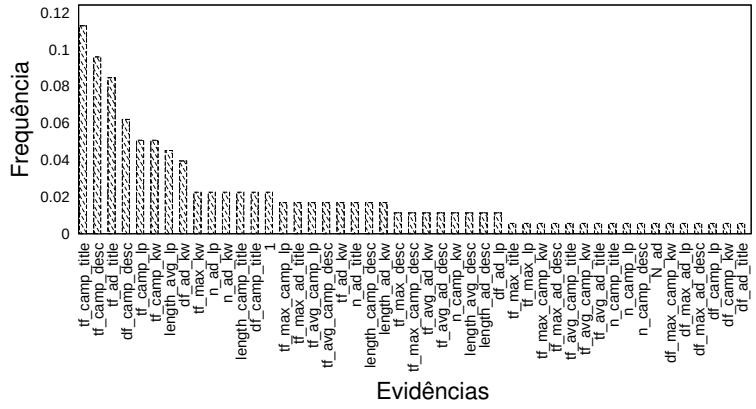
Conforme podemos ver tanto na Figura 5.7 quanto na Figura 5.8, todas as funções foram encontradas nos melhores indivíduos diferentemente do que aconteceu no caso das evidências (Seção 5.2).



(a) Distribuição D_1.

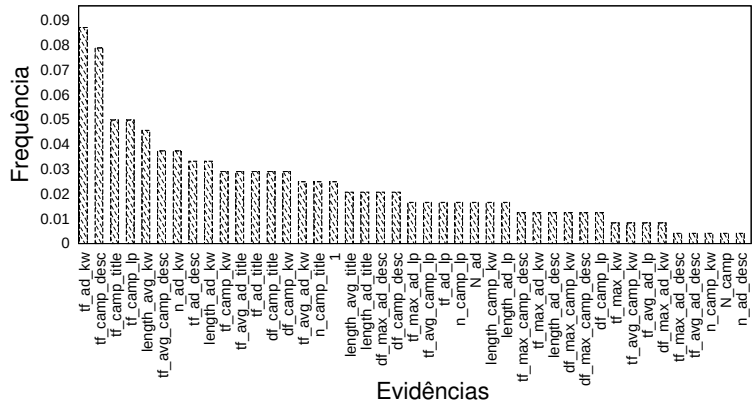


(b) Distribuição D_2.

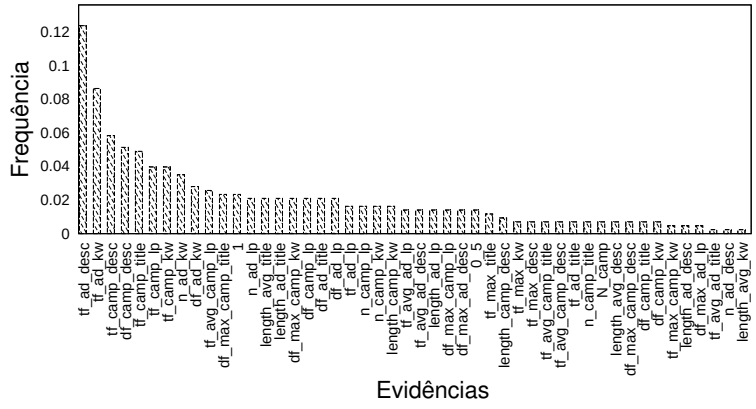


(c) Distribuição D_3.

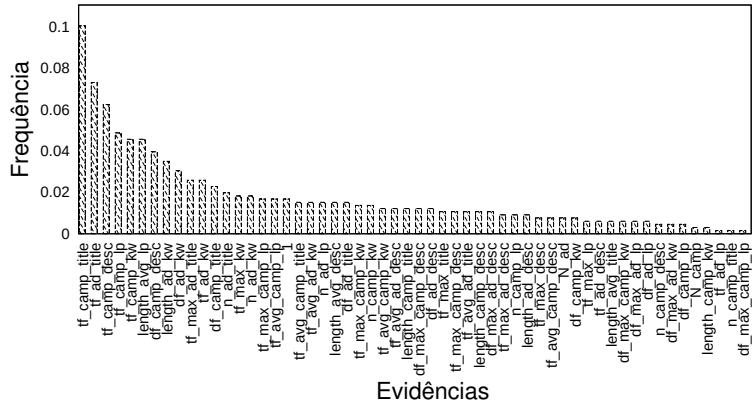
Figura 5.5: Frequências das evidências da melhor árvore de cada distribuição de páginas.



(a) Distribuição D_1.

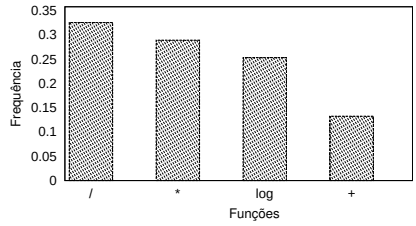


(b) Distribuição D_2.

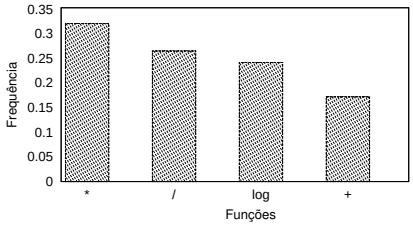


(c) Distribuição D_3.

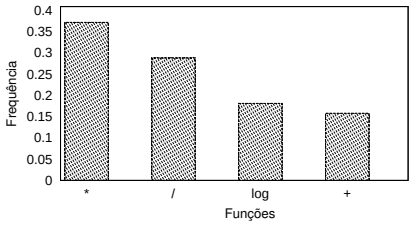
Figura 5.6: Frequências das evidências das 3 melhores árvores de cada distribuição de páginas.



(a) Distribuição D_1.

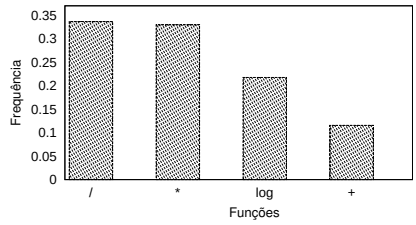


(b) Distribuição D_2.

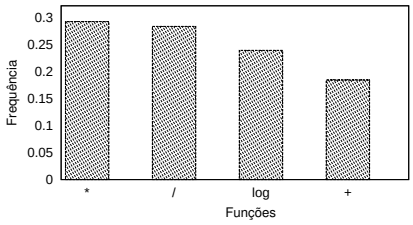


(c) Distribuição D_3.

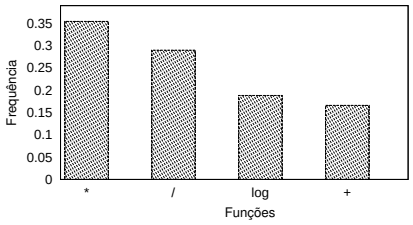
Figura 5.7: Frequências das funções da melhor árvore de cada distribuição de páginas.



(a) Distribuição D_1.



(b) Distribuição D_2.



(c) Distribuição D_3.

Figura 5.8: Frequências das funções das 3 melhores árvores de cada distribuição de páginas.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho propomos e testamos um novo arcabouço para a associação de propagandas e páginas da Web baseado em programação genética. Em particular, dada a importância da relevância para sistemas de propaganda direcionada baseada em conteúdo, nosso método de PG procura aprender funções capazes de selecionar as propagandas mais relevantes utilizando as evidências disponíveis.

Com o objetivo de testarmos nosso método utilizamos uma coleção real de propagandas e uma coleção de páginas da Web de um jornal brasileiro. Os ganhos obtidos por nosso método foram de 61,7% em relação método base. Além de propor funções capazes de fornecer boas estimativas de ordenação de propagandas, PG foi capaz de descobrir funções de ordenação muito eficazes na sugestão de propagandas relevantes e evitar a sugestão de propagandas irrelevantes.

Também procuramos entender qual a influência do tamanho do treino na eficácia dos indivíduos. A análise dos experimentos permitiu ver que, quando analisamos os extremos dos percentuais (quando comparamos os percentuais 20% e 90%, por exemplo), um maior conjunto de exemplos (pares página-propaganda com análise de relevância) leva a obtenção de indivíduos mais eficazes. Entretanto, em várias situações vimos que PG apresentou melhores resultados com apenas uma fração do conjunto de treino disponível. Em outras palavras, vimos que, em várias situações, os indivíduos mais eficazes foram encontrados a percentuais médios, como 40% ou 50% por exemplo. Logo, podemos concluir que, apesar da aleatoriedade inerente ao processo de evolução, existe um limite a partir do qual nem todos os exemplo são necessários. Em nossos experimentos vimos isso acontecer a partir de 40% ou 50% dos conjuntos de treino e de validação.

Uma das vantagens da utilização de PG em relação a outras técnicas como redes neurais e SVM é que as soluções baseadas em PG podem ser analisadas. Assim, realizamos uma série de experimentos a fim de tentar entender os motivos das funções propostas por PG serem tão eficazes. Nossas conclusões são:

- Os melhores indivíduos de cada geração, em geral, são bastante distintos entre si. Isto é, vimos que os melhores indivíduos utilizaram as evidências disponíveis de forma bastante distinta, porém alcançando resultados próximos em termos de eficácia. Ou seja, concluimos que diferentes combinações das evidências geram boas funções de ordenação das propagandas.

- Realizamos experimentos para a análise da frequência de ocorrência das funções matemáticas e da frequência de ocorrência das evidências. No caso das funções matemáticas vimos que todas essas funções foram utilizadas pelos melhores indivíduos. Já no caso da análise da frequência das evidências, vimos que apenas parte das evidências foi utilizada em grande parte dos casos. Isso é bastante interessante pois podemos intuir que apenas parte das evidências já são suficientes para encontrarmos indivíduos eficazes e que o processo evolutivo considerando-se apenas esse número reduzido de evidências pode ser vantajoso do ponto de vista de eficiência.

No futuro pretendemos expandir o arcabouço com a intenção de analisar o impacto da utilização de novas evidências e considerar outros aspectos importantes do problema de propaganda direcionada baseada em conteúdo. Além da utilização de novas evidências, uma possível linha de estudo é a evolução de indivíduos considerando-se apenas as evidências utilizadas pelos melhores indivíduos de populações anteriores. Caso tenhamos um número menor de evidências em relação ao conjunto de evidências disponíveis, como vimos em nossos experimentos, isso pode levar a um processo evolutivo mais eficiente, uma vez que a avaliação de cada indivíduo terá um custo menor.

Outra possível linha a ser investigada é a influência de variáveis do arcabouço de PG nos indivíduos encontrados. Como exemplos de tais variáveis temos o fator de mutação, a profundidade máxima da árvore que define um indivíduo, o tamanho da população e o número de gerações de uma população de indivíduos.

Pretendemos também estudar a eficácia de diferentes estratégias de corte com o objetivo de entendermos qual o impacto dessas alterações no processo de aprendizagem e na eficácia do arcabouço. Também pretendemos realizar testes para a comparação de nosso método e outras estratégias de aprendizado como estratégias baseadas em SVM [15]. Com relação a novos modelos pretendemos propor um arcabouço baseado em PG que utilize a informação de categoria das páginas e das propagandas. Um aspecto muito importante é a expansão do modelo para que esse gere funções capazes de combinar os aspectos monetário e de relevância do problema considerando a quantia que o anunciante está disposto a pagar para a sugestão de suas propagandas.

Bibliografia

[1] Giuseppe Attardi, Andrea Esuli, and Maria Simi. Best bets: thousands of queries in search of a client. In *Proceedings of the 13th international WWW conference on Alternate track papers & posters*, pages 422–423, New York, NY, USA, 2004. ACM Press.

[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, 1st edition, 1999.

[3] Wolfgang Banzhaf, Frank D. Francone, Robert E. Keller, and Peter Nordin. *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[4] Hemant K. Bhargava and Juan Feng. Paid placement strategies for internet search engines. In *Proceedings of the 11th international conference on World Wide Web*, pages 117–123, New York, NY, USA, 2002.

[5] John Joseph Carrasco, Daniel Fain, Kevin Lang, and Leonid Zhukov. Clustering of bipartite advertiser-keyword graph. In *Workshop on Clustering Large Datasets, 3th IEEE International Conference on Data Mining*, Melbourne, Florida, USA, November 2003. IEEE Computer Society Press. Available at `http://research.yahoo.com/publications.xml`.

[6] Oscar Cordon, Felix De Moya, and Carlos Zarco. A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 6(5):308–319, August 2002.

[7] Ricardo da S. Torres, ao Alexandre X. Falc Baoping Zhang, Weiguo Fan, Edward A. Fox, Marcos André Gonçalves, and Pavel Calado. A new framework to combine descriptors for content-based image retrieval. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 335–336, New York, NY, USA, 2005. ACM.

[8] Moisés G. de Carvalho, Marcos André Gonçalves, Alberto H. F. Laender, and Altigran S. da Silva. Learning to deduplicate. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 41–50, New York, NY, USA, 2006. ACM.

[9] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2000. 2nd Edition.

[10] eMarketer. Search engine marketing: User and spending trends, Oct 2007. Available at `http://www.emarketer.com/Reports/All/Emarketer_2000473.aspx?src=report_head_info_sitesearch`.

[11] eMarketer. Us advertising spending, Oct 2007. Available at `http://www.emarketer.com/Report.aspx?code=2000442`.

[12] Elena Eneva. Detecting invalid clicks in online paid search listings: a problem description for the use of unlabeled data. In Tom Fawcett and Nina Mishra, editors, *Workshop on the Continuum from Labeled to Unlabeled Data, 20th International Conference on Machine Learning*, Washington DC, USA, August 2003. AAAI Press.

[13] Weiguo Fan, Edward A. Fox, Praveen Pathak, and Harris Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004.

[14] Weiguo Fan, Michael D. Gordon, and Praveen Pathak. Discovery of context-specific ranking functions for effective information retrieval using genetic programming. *Transactions on Knowledge and Data Engineering*, 16(4):523–527, 2004.

[15] Weiguo Fan, Michael D. Gordon, and Praveen Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing and Management*, 40(4):587–602, 2004.

[16] Weiguo Fan, Michael D. Gordon, Praveen Pathak, Wensi Xi, and Edward A. Fox. Ranking function optimization for effective web search by genetic programming: An empirical study. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4*, page 40105, Washington, DC, USA, 2004. IEEE Computer Society.

[17] M. Gordon. Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988.

[18] Michael D. Gordon. User-based document clustering by redescribing subject descriptions with a genetic algorithm. *Journal of the American Society for Information Science and Technology (JASIST)*, 42(5):311–322, 1991.

[19] Donna Harman. Overview of the fourth text retrieval conference TREC-4. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 1–24, Gaithersburg, Maryland, USA, November 1996. NIST Special Publication 500-236.

[20] David Hawking, Nick Craswell, and Paul B. Thistlewaite. Overview of TREC-7 very large collection track. In *The Seventh Text REtrieval Conference (TREC-7)*, pages 91–104, Gaithersburg, Maryland, USA, November 1998.

[21] Jorng-Tzong Horng and Ching-Chang Yeh. Applying genetic algorithms to query optimization in document retrieval. *Information Process and Management*, 36(5):737–759, 2000.

[22] IAB and PricewaterhouseCoopers. IAB internet advertising revenue report, April 2005. Available at `http://www.iab.net/2004adrevenues`.

[23] John R. Koza. *Genetic programming: On the programming of computers by natural selection*. MIT Press, Cambridge, 1992.

[24] Anísio Lacerda, Marco Cristo, Marcos André Gonçalves, Weiguo Fan, Nivio Ziviani, and Berthier Ribeiro-Neto. Learning to advertise. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 549–556, New York, NY, USA, 2006. ACM.

[25] William B. Langdon. *Data Structures and Genetic Programming: Genetic Programming + Data Structures = Automatic Programming!* Kluwer, Boston, 1998.

[26] Kevin Lee. The SEM content conundrum. ClickZ Experts, July 2003. Available at `http://www.clickz.com/experts/search/strat/article.php/2233821`.

[27] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote, and Fênix de Moya-Anegon. Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology (JASIST).*, 54(2):152–160, 2003.

[28] Tom M. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1996.

[29] OneUpWeb. How keyword length affects conversion rates, January 2005. Available at `http://www.oneupweb.com/landing/keywordstudy_landing.htm`.

[30] Jeffrey Parsons, Katherine Gallagher, and K. Dale Foster. Messages in the medium: An experimental investigation of Web Advertising effectiveness and attitudes toward Web content. In Jr. Ralph H. Sprague, editor, *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6050, Washington, DC, USA, 2000. IEEE Computer Society.

[31] P. Pathak, M. Gordon, and W. Fan. Effective information retrieval using genetic algorithms based matching function adaptation. In *Proceedings of the 33rd Hawaii International Conference on System Science*, pages 523–527, Hawaii, USA, 2000.

[32] Berthier Ribeiro-Neto, Marco Cristo, Edleno Silva de Moura, and Paulo B. Golgher. Impedance coupling in content-target advertising. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 496–500, Salvador, Bahia, Brazil, July 2005.

[33] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[34] Melius Weideman. Ethical issues on content distribution to digital consumers via paid placement as opposed to website visibility in search engine results. In *The 17th ETHICOMP*, pages 904–915. Troubador Publishing Ltd, April 2004.

[35] Melius Weideman and Timothy Haig-Smith. An investigation into search engines as a form of targeted advert delivery. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 258–258. South African Institute for Computer Scientists and Information Technologists, 2002.

[36] Baoping Zhang, Yuxin Chen, Weiguo Fan, Edward A. Fox, Marcos Gonçalves, Marco Cristo, and Pável Calado. Intelligent gp fusion from multiple sources for text classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 477–484, New York, NY, USA, 2005. ACM Press.

[37] Baoping Zhang, Marcos André Gonçalves, Weiguo Fan, Yuxin Chen, Edward A. Fox, Pável Calado, and Marco Cristo. Combining structural and citation-based evidence for text classification. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 162–163, New York, NY, USA, 2004. ACM.

[38] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.