Universidade Federal do Amazonas
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Programa de Pós-Graduação em Informática

Uma Abordagem Evolutiva para Combinação de Fontes de Evidência de Relevância em Máquinas de Busca

Thomaz Philippe Cavalcante Silva

Manaus – Amazonas
Abril de 2008

Thomaz Philippe Cavalcante Silva

Uma Abordagem Evolutiva para Combinação de Fontes de Evidência de Relevância em Máquinas de Busca

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Recuperação de Informação.

Orientador: Prof. Dr. Edleno Silva de Moura

Thomaz Philippe Cavalcante Silva

Uma Abordagem Evolutiva para Combinação de Fontes de Evidência de Relevância em Máquinas de Busca

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Recuperação de Informação.

Banca Examinadora

Prof. Dr. Edleno Silva de Moura – Orientador
Departamento de Ciência da Computação – UFAM/PPGI

Prof. Dr. Marcos André Gonçalves
Departamento de Ciência da Computação – UFAM/PPGI

Prof. Dr. Marco André Christo
Departamento de Ciência da Computação – UFAM/PPGI

Manaus – Amazonas
Abril de 2008

Aos meus pais que guiaram meus caminhos até aqui.

Agradecimentos

A Deus, acima de tudo.

Ao meu orientador, Edleno Silva de Moura, pelo profissionalismo e ensinamentos transmitidos.

Aos meus pais, José Wellington Silva e Sandra Cavalcante Silva, pela dedicação e apoio em todos os momentos desta caminhada.

A minha namorada, Lorisa Simas Teixeira, e seus pais, pela motivação e paciência.

Aos colegas João Marcos Bastos Cavalcanti e Altigran da Silva Soares, pela orientação fundamental para a realização deste trabalho.

Ao colega Moisés Gomes de Carvalho pela contribuição inicial no desenvolvimento desta dissertação.

Aos amigos Roberto Oliveira dos Santos e Filipe de Sá Mesquita pelo companheirismo diário.

Aos amigos André Luis da Costa Carvalho, Klessius Berlch e Bruno Araújo pelas grandes contribuições a este trabalho, além do companheirismo e amizade diária.

Aos amigos do GTI: Eli Cortez, Mauro Rojas, Javier Zambrano e Lisandra Santos pela amizade e convivência.

À Elienai Nogueira, pelo apoio administrativo e pela amizade.

Ao PPGI, pela oportunidade.

A todas as pessoas que me ajudaram na realização dos experimentos de avaliação de relevância de consultas, imprescindíveis para realização desta dissertação, meus sinceros agradecimentos

A todos aqueles que ajudaram de alguma forma na realização deste trabalho, o meu mais profundo agradecimento.

A importante tarefa de evoluir o mundo não pode esperar por homens perfeitos.

George Eliot

Resumo

Máquinas de busca modernas utilizam diferentes estratégias para melhorar a qualidade de suas respostas. Uma estratégia importante é obter uma única lista ordenada de documentos baseada em listas produzidas por diferentes fontes de evidência. Este trabalho estuda o uso de uma técnica evolutiva para gerar boas funções de combinação de três diferentes fontes de evidência: o conteúdo textual dos documentos, as estruturas de ligação entre os documentos de uma coleção e a concatenação dos textos de âncora que apontam para cada documento. As funções de combinação descobertas neste trabalho foram testadas em duas coleções distintas: a primeira contém consultas e documentos de uma máquina de busca real da Web que contém cerca de 12 milhões de documentos e a segunda é a coleção de referência LETOR, criada para permitir a justa comparação entre métodos de aprendizagem de funções de ordenação. Os experimentos indicam que a abordagem estudada aqui é uma alternativa prática e efetiva para combinação de diferentes fontes de evidência em uma única lista de respostas. Nós verificamos também que diferentes classes de consultas necessitam de diferentes funções de combinação de fontes de evidência e mostramos que nossa abordagem é viável em identificar boas funções.

Palavras-chave: Recuperação de Informação, Máquinas de Busca, Web, Programação Genética, Funções de Ordenação de Consultas.

Abstract

Modern Web search engines use different strategies to improve the overall quality of their document rankings. An important strategy to obtain a unique ranking based in the rankings of different sources of evidence may be studed. This work studied an evolutionary approach to derive good evidence combination functions using three different sources of evidences: the textual content of documents, the link structure in the database and the anchor text concatenation. The combination functions discovered by our evolutionary strategies were tested with two distinct datasets: the first one contains a colection of queries and documents extracted from a real case Web search engine with over 12 million documents, the second dataset we use was LETOR benchmark dataset for learning to rank methods. The experiments performed indicate that our proposal is an effective and practical alternative for combining sources of evidence in a single ranking. We also show that different types of queries submitted to a search engine may require different combination functions and that our proposal is useful for coping with such differences.

Sumário

Sumário **i**

Lista de Figuras **iii**

Lista de Tabelas **iv**

1 Introdução **1**

1.1 Objetivos 3

1.2 Revisão da Literatura 4

1.3 Contribuições 8

1.4 Organização da Dissertação 8

2 Conceitos Básicos **10**

2.1 Fontes de Evidência de Relevância 10

2.2 Programação Genética 13

2.2.1 Operadores Genéticos 15

2.2.2 Algoritmo Evolutivo 19

2.2.3 Fases da evolução 20

3 Combinando Evidências Usando PG **22**

4 Experimentos **25**

4.1 Metodologia 25

4.1.1 Fontes de evidência combinadas 25

4.1.2 Classes de consulta 26

4.1.3 Função de Aptidão 27

4.1.4 Coleções de referência 29
4.1.5 Configuração dos experimentos 32
4.2 Resultados 35
4.2.1 WBR03 36
4.2.1.1 Estabilidade em diferentes rodadas 36
4.2.1.2 Estabilidade em diferentes conjuntos de treinamento 40
4.2.1.3 Comparação com métodos de referência 41
4.2.1.4 Impacto de cada fonte de evidência na combinação 43
4.2.2 LETOR 44
4.2.2.1 Estabilidade em diferentes rodadas 44
4.2.2.2 Estabilidade em diferentes conjuntos de treinamento 46
4.2.2.3 Comparação com métodos de referência 48
4.2.2.4 Impacto de cada fonte de evidência na combinação 48

5 Conclusões e Trabalhos Futuros **50**

Referências Bibliográficas **53**

Lista de Figuras

2.1 Exemplo de uma função matemática no modelo de árvore de PG. 14
2.2 Exemplo de um cruzamento entre árvores de PG. 17

4.1 Estabilidade do processo de PG em 20 diferentes rodadas para consultas Navegacionais Populares. 38
4.2 Estabilidade do processo de PG em 20 diferentes rodadas para consultas Navegacionais Não-Populares. 38
4.3 Estabilidade do processo de PG em 20 diferentes rodadas para consultas Informacionais Populares. 39
4.4 Estabilidade do processo de PG em 20 diferentes rodadas para consultas Informacionais Não-Populares. 39
4.5 Estabilidade do processo de PG em 20 diferentes rodadas para consultas sobre a TD2003 do LETOR. 46
4.6 Estabilidade do processo de PG em 20 diferentes rodadas para consultas sobre a TD2004 do LETOR. 46

Lista de Tabelas

4.1 Estatísticas sobre a coleção WBR03. 30

4.2 Fontes de evidência utilizadas extraída das coleções TREC no LETOR. 32

4.3 Parâmetros adotados nos experimentos parciais 33

4.4 Número médio de palavras distintas no conteúdo textual e concatenação de texto de âncora dos documentos relevantes das quatro classes de consulta estudadas. 35

4.5 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas navegacionais populares. Todos os valores estão expressos em MRR. 40

4.6 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas navegacionais não-populares. Todos os valores estão expressos em MRR. 40

4.7 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas informacionais populares. Todos os valores estão expressos em bpref-10 41

4.8 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas informacionais não-populares. Todos os valores estão expressos em bpref-10. 41

4.9 Resultados de MRR para consultas navegacionais quando combinadas as diferentes fontes de evidência com PG, BN, BLC and SIGM. 42

4.10 Resultados da combinação das diferentes fontes de evidência usando PG, BN, BLC and SIGM de acordo com as métricas Bpref-10 e MAP em consultas informacionais. 42
4.11 Resultados atingidos por Programação Genética em MRR de consultas navegacionais usando diferentes combinações de fontes de evidência. Todas significa a combinação usando as 3 fontes de evidência. 43
4.12 Resultados atingidos por Programação Genética de acordo com as métricas Bpref-10 e MAP em consultas informacionais usando diferentes combinações de fontes de evidência. 44
4.13 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas sobre a coleção TD2003. Todos os valores estão expressos em bpref-10. 47
4.14 Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas sobre a coleção TD2004. Todos os valores estão expressos em bpref-10. 47
4.15 Resultados da combinação das diferentes fontes de evidência usando PG, BN, BLC and SIGM de acordo com as métricas Bpref-10 e MAP em consultas informacionais. 48
4.16 Resultados atingidos por Programação Genética de acordo com as métricas Bpref-10 e MAP nas coleções TD2003 e TD2004 usando diferentes combinações de fontes de evidência. 49

Capítulo 1

Introdução

Máquinas de busca se tornaram um importante meio para obter informação de qualidade disponível na Web. As máquinas de busca tentam identificar quais são os documentos mais relevantes, dentre os que atendem a uma consulta, de acordo com a necessidade de informação de cada usuário. Para identificar os documentos mais relevantes as máquinas de busca utilizam fontes de evidência de relevância que são modeladas para atribuir graus de relevância (*similaridade*) aos documentos previamente coletados e armazenados numa base de dados [1]. Dada uma consulta, as fontes de evidência de relevância atribuem valores numéricos aos documentos da base a fim de gerar uma ordenação (*ranking*) dos documentos de modo a tentar colocar os mais relevantes nas primeiras posições. Entre essas fontes de evidência podemos citar: os textos dos documentos, os textos de âncora dos documentos, as estruturas de ligação entre os documentos, a localização geográfica dos documentos, o nível de URL [1], entre outros.

Estudos passados [23, 24] mostram que utilizar fontes de evidência de relevância isoladamente não é uma estratégia adequada para obter as melhores respostas em uma máquina de busca. Isso acontece porque cada fonte de evidência trás uma informação complementar que contribui na identificação de bons documentos. A combinação se dá através de uma função matemática que relaciona os valores numéricos atribuídos pelas fontes de evidência a um determinado documento em um valor único chamado de *similaridade unificada* do documento em relação a uma consulta. Os documentos são então reordenados a partir dos valores de similaridade unificada. Sabendo-se que, dada uma consulta, existe pelo menos uma ordenação ótima dos documentos de acordo com a relevância do mesmo e que as possibilidades de relacionar as fontes de evidência são infinitas,

[1] profundidade da URL no diretório

descobrir a função de combinação que gera a ordenação ótima dos documentos da base é um problema de otimização. Como identificar manualmente uma função de combinação é uma tarefa árdua e de alto custo, estratégias automáticas para combinar quaisquer fontes de evidência de relevância devem ser estudas para melhorar a qualidade das respostas de sistemas de busca.

Trabalhos passados foram realizados buscando alternativas genéricas para combinar fontes de evidência de relevância. Em [35], foi proposto um método para realizar a combinação linear dos valores de similaridade gerados por cada fonte de evidência a cada documento de uma coleção. Outra forma de realizar a combinação, proposta em [31, 4], modela o valor de similaridade como uma probabilidade independente de relevância do documento. A partir daí foram utilizados modelos estatísticos para combinação dos resultados. Os resultados obtidos através desses métodos de combinação apresentaram ganhos sobre as fontes de evidência isoladas. Estes métodos são baseados em modelos que simplificam o problema de combinação de fontes de evidência através de hipóteses como: independência entre as fontes de evidência, compatibilidade nas distribuições de valores de similaridade de diferentes fontes, entre outras. Estes problemas não são tratados por estes métodos, havendo, portanto, a necessidade de estudar com maior profundidade outras abordagens que consigam identificar o máximo de características presentes nos resultados e usá-las de maneira adequada na combinação.

Outro método proposto na literatura apresenta uma técnica para combinar fontes de evidência independentes de consultas, como o Pagerank, com uma fonte de evidência de texto [8]. A abordagem proposta utiliza uma técnica de ajuste de parâmetros que necessita de uma fase de treinamento para ser realizada. Apesar do custo associado à realização do treinamento, abordagens desse tipo têm obtido melhores resultados na geração de funções de combinação. Os resultados mostram que o método melhora a qualidade das respostas, porém, a abordagem é restrita às fontes de evidência independentes de consulta.

Estudos realizados em [20, 33] mostram que não é suficiente encontrar uma única função de combinação que englobe todas as possíveis consultas submetidas por usuários. Diferenças no objetivo dos usuários ao realizar uma consulta afetam diretamente a qualidade da combinação. Para cada tipo de consulta submetida, é necessário que seja criada uma função de combinação única. Como exemplo real, consultas cujo interesse é adquirir conhecimento sobre um determinado tópico de informação devem usar evidências de conteúdo textual com maior impacto na combinação. Por outro lado, consultas associadas a serviços de uma determinada

cidade devem utilizar as fontes de evidência sobre localidade geográfica de um documento com maior importância. Portanto, a abordagem de combinação de fontes de evidência deve ser capaz de reconhecer a importância de cada fonte para diferentes tipos de consultas e modelar esta importância de forma adequada na função de combinação.

1.1 Objetivos

O objetivo desse trabalho foi estudar uma abordagem de aprendizagem automática chamada Programação Genética (PG) para combinar diversas fontes de evidência de relevância em uma máquina de busca. PG foi escolhida como estratégia para realizar a combinação de fontes de evidência devido ao seu sucesso ao descobrir funções de ordenação de documentos em sistemas de busca [14]. Em um trabalho apresentado em [11], foi proposta uma abordagem para combinar fontes de evidência com a finalidade de aumentar a qualidade das respostas de um sistema de busca. Os resultados obtidos indicaram que PG não era uma estratégia adequada para realizar a combinação de fontes de evidência em uma máquina de busca. Por acreditar que a aplicação de PG é viável, realizamos um novo estudo com o mesmo objetivo nesta dissertação. Realizamos mudanças na metodologia de experimentos e na implementação do modelo de PG para validar ou invalidar os resultados obtidos previamente. Com nossa abordagem mostramos que, além de obter bons resultados, a PG permite identificar nas funções matemáticas geradas a importância de cada fonte de evidência na combinação.

Para avaliar a abordagem estudada foram realizados diversos experimentos em uma máquina de busca real da Web. Os experimentos envolviam três fontes de evidência amplamente usadas em diversos trabalhos da literatura. A primeira fonte de evidência está contida nos textos dos documentos da Web e é obtida através da aplicação do modelo de espaço vetorial [30]. A segunda fonte está presente na concatenação dos textos de âncora relativos a um documento e é obtida também com a aplicação do modelo de espaço vetorial. A última fonte de evidência é obtida através da análise das ligações entre os documentos da Web obtida com o cálculo do Pagerank [26].

Como diferentes objetivos para realização de uma consulta exigem diferentes estratégias de combinação, no presente trabalho avaliamos a aplicação da abordagem proposta em quatro cenários de consultas. Primeiro, as consultas foram divididas em duas classes: informacionais e

navegacionais. Em seguida, separamos as consultas de acordo com o número de submissões ao sistema de busca, classificando-as entre populares e não-populares. Os experimentos mostraram que as diferentes classes de consulta[2] afetam diretamente o modo como as fontes de evidência são usadas na combinação. Portanto, analisar a importância de cada fonte de evidência na combinação de diferentes classes de consulta foi um dos objetivos desse trabalho. Os resultados dos experimentos mostram que usar PG obtém bons resultados na combinação de diversas fontes de evidência nos variados cenários.

1.2 Revisão da Literatura

Trabalhos relacionados à combinação de fontes de evidência de relevância em máquinas de busca são discutidos a seguir. São apresentados também trabalhos sobre a aplicação de PG na área de recuperação de informação.

Trabalhos de combinação de fontes de evidência direcionam seus esforços na tentativa de otimizar a qualidade das respostas retornadas a uma consulta. Nesse sentido, Salton [29] desenvolveu um trabalho de recuperação de informação utilizando artigos científicos que obtém ganhos na qualidade dos resultados de consultas submetidas através da combinação de evidências de texto com informações contidas em citações. Ele concluiu que documentos que possuem similaridade entre citações tendem a tratar de assuntos semelhantes.

Dado que o aumento da qualidade das respostas está intimamente ligado à forma como as máquinas de busca modelam a informação, novas fontes de evidência começaram a ser estudadas e novas formas de combiná-las de maneira eficiente também. Lee [23, 24] propõe uma nova maneira para realizar a combinação dos valores de similaridades de documentos retornados por diferentes fontes de evidência e conclui, através de experimentos, que combinar fontes de evidência permite obter ganhos significativos em relação a fontes isoladas.

Em [35], Westerveld propõe uma abordagem empírica para escolha de pesos de cada uma das fontes de evidência disponíveis. Este trabalho analisa uma fórmula de probabilidade linear para combinar fontes de evidência e assim estudar o problema de busca por página de entrada[3], que é a busca por um documento específico na Web. Westerveld conclui que a inserção de

[2] Informacionais populares, informacionais não-populares, navegacionais populares e navegacionais não-populares

[3] The entry page search prolbem

novas fontes de evidência sobre um documento aumenta as chances de resolver o problema da página de entrada. Outros dois trabalhos propostos por Craswell [7] e Chowdhury [6] apresentam resultados bastantes similares ao resultado obtido por Westerveld.

Em Silva et al [31] e posteriormente em Calado et al [4], estendendo o trabalho de Ribeiro-Neto [27], foi proposta uma abordagem de combinação de diferentes fontes de evidência utilizando Redes Bayesianas. Nesses trabalhos, valores de similaridade de documentos foram modeladas como probabilidades independentes e um arcabouço de Redes Bayesianas foi desenvolvido para efetuar a combinação destas similaridades. O objetivo do trabalho era aumentar a qualidade de respostas de uma máquina de busca. Os resultados apresentaram ganhos de até 59% sobre a fonte de evidência baseada em conteúdo textual.

Em [33], Upstill et al estudam o impacto de diferentes fontes de evidência em consultas navegacionais. Diferentes formas de combinar fontes de evidência de relevância são apresentadas e experimentos foram realizados para mostrar o impacto de cada fonte de evidência no resultado final. Upstill concluiu que para o processamento de consultas navegacionais a fonte de evidência encontrada em textos de âncora de documentos deve ser utilizada a fim de aumentar a qualidade dos resultados em sítios populares. Além disso, mostrou-se que fontes de evidência baseadas em texto ajudam a prevenir que páginas pessoais com textos de âncora inadequados não sejam ignoradas. Este trabalho também aponta a necessidade de estudos futuros em busca de melhores estratégias de combinação destas duas fontes de evidência.

Em [8], Craswell propõe um conjunto de métodos de ajuste de parâmetros para atribuir pesos a fontes de evidência independentes de consulta. Esses pesos são utilizados sobre fontes de evidência independentes de consulta para produzir uma combinação linear com fontes de evidências dependentes de consultas como o texto e os textos de âncora. Nossa abordagem pode ser vista como uma generalização da abordagem baseada em ajuste de parâmetros proposta por Craswell por considerar quaisquer métodos de ordenação de documentos existentes numa máquina de busca. Além disso, as funções geradas por nossa abordagem identificam melhor as interdependências existentes entre as fontes de evidência para determinadas classes de consultas, pois são capazes de identificar e utilizar combinações não-lineares entre as fontes de evidência, melhorando a qualidade da combinação.

Em [17], foi proposta uma abordagem de aprendizagem automática utilizando o método conhecido como *Support Vector Machine* (SVM) para combinar fontes de evidência. O trabalho

propõe combinar informações extraídas dos cliques sobre documentos de resposta a uma consulta submetida com fontes de evidência tradicionais como o conteúdo textual de uma página e fontes independentes de consultas.

O presente trabalho para combinação de fontes de evidência de relevância utiliza uma técnica evolucionária chamada Programação Genética (PG). PG foi aplicada em várias pesquisas na área de recuperação de informação. A primeira aplicação de PG em recuperação de informação foi feita por Chen [5]. A partir daí novas aplicações surgiram até o trabalho proposto por Fan et al em [14]. Nesse trabalho, Fan apresenta um sistema de PG com o objetivo de desenvolver funções para cálculo de similaridade de documentos utilizando estatísticas extraídas dos próprios documentos de uma coleção. Nos experimentos realizados por Fan a coleção de referência TREC[4] foi utilizada. Os resultados obtidos por Fan foram superiores às funções tradicionais de ordenação como Okapi [28] e o modelo de espaço vetorial [30].

Fan et al [15] também aplicam PG para gerar funções de ordenação de contexto [5] específico. Fan discute a questão do *superadaptação* nas funções de similaridade geradas pela abordagem, problema comum em métodos de aprendizagem automática. Para reduzir o impacto negativo do superadaptação, Fan usa uma divisão do conjunto de consultas usadas nos experimentos em três partes: treinamento, validação e teste, mesma abordagem utilizada em nossos experimentos. Seu trabalho também mostra que diferenças na formulação de funções de similaridade afetam o desempenho de sistemas de busca. Novamente são mostrados ganhos sobre outras abordagens da literatura. Em [13], os autores exploram fontes de evidência baseadas nas estruturas dos documentos da Web e seus resultados mostram ganhos ao utilizar estatísticas baseadas nessas evidências.

Trotman [32] explora uma abordagem similar àquela proposta por Fan et al [14], com uma pequena mudança. Ele explicitamente inclui funções de similaridade usadas como referência no início do processo evolutivo. Dessa forma é possível iniciar o processo evolutivo partindo de boas funções iniciais. Os resultados mostraram que essa abordagem obtém ganho substancial em relação à proposta por Fan et al.

PG já havia sido estudada como abordagem para combinação de fontes de evidência em [11]. No entanto, o trabalho de Carvalho apresenta resultados que sugerem que PG não é uma boa

[4] http://trec.nist.gov

[5] Por contexto entenda: diferentes tipos de consultas, usuários e coleções de documentos

abordagem para combinação de fontes de evidência. O trabalho de Carvalho apresenta alguns problemas na metodologia de experimentos que podem ter levado a resultados imprecisos. Um dos problemas encontrados foi a não utilização de uma função de aptidão comprovadamente eficiente para evolução das funções de combinação o que pode ter levado a resultados imprecisos. Outro problema foi a falta de experimentos para comprovar a estabilidade e a eficiência da estratégia de PG, como por exemplo: experimentos de validação cruzada e experimentos para análise da importância das evidências na combinação. Carvalho também nao realiza experimentos buscando evoluir funções para diferentes classes de consultas. A fim de reavaliar a abordagem, no presente trabalho nós propomos uma nova metodologia para os experimentos: revisamos a implementação do arcabouço de PG, modificamos os parâmetros necessários para a evolução de funções, discutimos um estudo mais aprofundado a respeito da importância de diferentes fontes de evidência em consultas com diferentes objetivos e diferentes popularidades, realizamos experimentos com uma nova base de dados e utilizamos novas funções de aptidão, a fim de aumentar a eficiência do processo evolutivo, como discutido na Seção 4.1.3. Para uma comparação completa com o trabalho proposto por Carvalho, veja o Capítulo 4.

Em paralelo com o trabalho desenvolvido aqui,Yeh et al. [36] estudaram o impacto da aprendizagem de funções de combinação de fontes de evidência utilizando PG sobre uma base de dados pública. Apesar dos objetivos iguais, tal estudo não investigou em profundidade as características da abordagem utilizando PG na tarefa de combinação. Por exemplo, em nosso trabalho são executados experimentos para mostrar a estabilidade do processo evolutivo em diferentes rodadas e são apresentados também experimentos para verificar a estabilidade do processo em diferentes conjuntos de treino. Diferentemente do trabalho de Yeh et al, realizamos experimentos com diferentes classes de consulta, fundamental para aumentar a qualidade das respostas de um sistema de busca por utilizar funções de combinação mais apropriadas a diferentes cenários de consulta. Por último, nosso trabalho propõe uma análise cuidadosa sobre o impacto das diferentes fontes de evidência na combinação em diferentes classes de consulta, o que nos permitiu comprovar que a importância das fontes de evidência deve variar para diferentes tipos de consultas.

1.3 Contribuições

As principais contribuições desse trabalho, as quais o diferenciam dos demais propostos anteriormente na literatura, são listadas a seguir:

1. Propomos alterações na metodologia de experimentos e mudanças na implementação de um arcabouço de PG voltado à combinação de diferentes e distintas fontes de evidência proposto em [11].

2. Exploramos a PG como ferramenta para análise da importância de cada fonte de evidência de relevância usada em uma máquina de busca para diferentes tipos de consulta que podem ser submetidas ao sistema.

3. Procuramos novas funções objetivo necessárias ao modelo de PG para identificar e separar boas funções durante o processo de treinamento. Também adotamos uma nova configuração de parâmetros do arcabouço de PG que permite redução no custo de processamento e aumento da qualidade das funções de combinação. Além destas alterações, realizamos modificações no código fonte do arcabouço a fim de corrigir erros de programação que poderiam interferir nos resultados.

4. Construímos uma base de dados para experimentos de busca em uma máquina de busca real da Web contendo consultas e respectivas informações de relevância de documentos. As consultas estão divididas de acordo com o objetivo (informacionais e navegacionais) e também de acordo com a popularidade delas (populares e não-populares).

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte maneira. No Capítulo 2 apresentamos os conceitos básicos sobre recuperação de informação usando fontes de evidência de relevância e Programação Genética necessários para compreensão dos termos e métodos utilizados no trabalho. No Capítulo 3 discutimos a abordagem de PG aplicada à combinação de diferentes fontes de evidência de relevância. A metodologia utilizada na execução dos experimentos, resultados da combinação de fontes de evidência, comparações com as próprias fontes isoladas e outros métodos de combinação presentes na literatura, e a discussão a respeito dos resultados obtidos

está presente no Capítulo 4. Finalmente, no Capítulo 5 apresentamos as conclusões e trabalhos futuros.

Capítulo 2

Conceitos Básicos

Nesta seção apresentamos os conceitos básicos necessários para o entendimento do trabalho desenvolvido aqui. Os conceitos estão divididos em dois grupos, o primeiro descreve as fontes de evidência de relevância que podem ser usadas por uma máquina de busca e o segundo apresenta a abordagem evolutiva Programação Genética (PG).

2.1 Fontes de Evidência de Relevância

O desempenho de um sistema de busca pode ser afetado por diferentes fatores como: qualidade da representação de consultas submetidas, abrangência da coleta e armazenagem de documentos, qualidade da indexação de documentos, entre outros. Um dos principais fatores que afetam a qualidade das respostas de uma máquina de busca é o método de ordenação de documentos adotado durante o processamento de consultas. Uma máquina de busca utiliza fontes de evidência de relevância contidas nos documentos de sua coleção a fim de estimar a importância de cada documento encontrado na resposta dada a uma consulta.

Existem diversas fontes de evidência disponíveis nos documentos da Web. Os textos dos documentos, os textos de âncora dos documentos, as estruturas de ligação entre os documentos, a localização geográfica dos documentos e o nível da URL são exemplos de fontes de evidência de relevância adotadas em sistemas de busca. Sobre cada uma das fontes há, também, uma variedade de funções de ordenação que podem ser aplicadas. Dentre as fontes de evidência, três foram utilizadas na presente dissertação:

- *Conteúdo textual.* Possui o conteúdo textual extraído de cada página.

- *Textos de âncora dos documentos.* Esta fonte de evidência é obtida através da concatenação de todos os textos que descrevem apontadores para um documento. Um documento é então representado pelos textos que o descrevem em outros documentos da coleção. Pode-se utilizar sobre esta fonte as mesmas funções de ordenação de documentos que se usa sobre a fonte extraída do conteúdo textual dos documentos.

- *Estruturas de ligação entre documentos.* Esta fonte de evidência é obtida através da análise dos apontadores entre os documentos de uma coleção. Uma conexão entre dois documentos pode ser vista como um voto de confiança do autor do documento que aponta no conteúdo informativo do documento apontado. Métodos foram criados para analisar o grau de importância dos documentos na coleção a partir dessas conexões. O Pagerank [26] é o método mais popular proposto na literatura para calcular o valor desta fonte de evidência.

Estas fontes de evidência foram adotadas nos experimentos de combinação da presente dissertação. A seguir serão apresentadas as definições de dois métodos de ordenação de documentos utilizados para obter as três fontes de evidência utilizadas nos experimentos do Capítulo 4. O primeiro, trata-se do tradicional Modelo de Espaço Vetorial, que pode ser aplicado tanto sobre fontes de evidência do conteúdo textual de documentos quanto sobre a concatenação de textos de âncora de um documento. O segundo, refere-se ao Pagerank, modelo de análise de conexões entre documentos amplamente utilizados em máquinas de busca modernas. As motivações para escolha das fontes de evidência usadas nos experimentos do presente trabalho são apresentadas no Capítulo 4.

Modelo de espaço vetorial

O modelo de espaço vetorial é um modelo que representa documentos em formato de texto como vetores em um espaço. Todos os elementos deste modelo, que podem ser consultas ou documentos, apresentam-se como conjuntos de termos. A dimensão do espaço é dada pela quantidade de termos distintos presentes em uma coleção de documentos na qual o modelo é aplicado. Um elemento é representado por um vetor cujas coordenadas são determinadas pelos termos que o compõem. As coordenadas de um vetor podem ser obtidas de diversas maneiras. Dentre elas, o método *tf-idf* é um dos que obtém melhores resultados e, por ter sido utilizado em nossos experimentos, será explicado nesta seção.

Em sistemas de busca, o modelo de espaço vetorial atribui um grau de relevância a um documento em relação a uma consulta submetida através do cálculo de proximidade entre seus vetores. O valor que representa a proximidade entre o vetor de documento e o vetor de consulta é chamado de valor de similaridade. Dessa forma, é esperado que documentos que apresentam maior valor de similaridade em relação a uma consulta sejam documentos mais relevantes. A função geralmente usada para calcular um valor de similaridade entre dois vetores é o cálculo do cosseno dos ângulos desses vetores utilizando a seguinte fórmula:

$$sim(d,q) = \frac{\sum_{i=1}^{t} w(i,d) \times w(i,q)}{\sqrt{\sum_{i=1}^{t}(w(i,d))^2} \times \sqrt{\sum_{i=1}^{t}(w(i,q))^2}}$$

onde $sim(d,q)$ corresponde ao valor de similaridade do documento d em relação a consulta q, t é o número de termos distintos da coleção e $w(i,e)$ é a função que calcula a importância do termo i no elemento e, que é dada pela seguinte função:

$$w(i,e) = tf(i,e) \times idf(i)$$

onde $tf(i,e)$ é a frequência do termo i no elemento e, $idf(i)$ é a frequência inversa do documento i, calculada por:

$$idf(i) = log(\frac{N}{n_i})$$

onde N é o número total de documentos da coleção e n_i é o número de documentos em que o termo i ocorreu.

Pagerank

O Pagerank é uma métrica criada para identificar a importância dos documentos de uma coleção em uma máquina de busca através da análise das estruturas de ligação da Web. Documentos muito importantes sobre determinados assuntos tendem a atrair muitos apontadores de outros documentos da Web que se relacionam de alguma maneira com aquele assunto. A popularidade de um documento relativa ao número de apontadores incidentes pode ser um indicador de que o documento apresenta uma grande importância na coleção. Além disso, é razoável pensar que documentos importantes tendem a transferir esta importância aos documentos a que apontam.

O Pagerank é um algoritmo que tenta identificar estas características no grafo da Web e mensurar a importância de documentos em valores numéricos.

O algoritmo do Pagerank modela um usuário da Web que navega entre os documentos de uma coleção de maneira aleatória. Para realizar esta navegação o usuário segue os apontadores que interligam os documentos em uma coleção. O valor que um documento obtém através do cálculo do Pagerank é uma estimativa deste usuário chegar ao documento através de uma navegação aleatória. Além de navegar entre as páginas seguindo os apontadores de maneira aleatória, é possível que o usuário realize uma seleção aleatória de qualquer outro documento da coleção. Por isso, considera-se que cada documento apresenta sempre uma probabilidade mínima de ser visitado, sendo esta probabilidade conhecida na literatura por *damping factor*.

O Pagerank de uma página p em um grafo $G = (V, E)$ é calculado da seguinte maneira:

$$PR(p) = \left[(1 - c) \times \sum_{q \in I(p)} \frac{PR(q)}{|O(q)|} \right] + \frac{c}{|V|} \tag{2.1}$$

onde c é o *damping factor*, $|O(q)|$ é o número de páginas apontadas pela página q, $I(p)$ é o conjunto das páginas que apontam para p, e $|V|$ é o número de páginas na coleção.

2.2 Programação Genética

PG é uma técnica de aprendizagem automática cujo objetivo é evoluir programas de computador seguindo processos baseados no conceito de *Seleção Natural* proposto por Darwin [10]. Esta técnica é utilizada para realizar busca por boas soluções para um determinado problema dentro de um grande espaço de soluções. Estas soluções são modeladas como programas de computador.

De acordo com o conceito de seleção natural, os seres vivos estão num processo de seleção constante em que o meio ambiente afeta diretamente a capacidade de sobrevivência. Quando algum organismo[1] apresenta uma modificação que seja útil na competição pela vida ele apresentará maior probabilidade de sobrevivência. Os indivíduos que adquirirem esta modificação perpetuam esta capacidade a seus descendentes através do processo conhecido por herança, na qual as características dos indivíduos pais são propagadas aos filhos através do código genético. Com o passar de gerações, os organismos evoluem e cada vez mais se adaptam ao meio ambiente em que se encontram.

[1] Em biologia um organismo é um ser vivo.

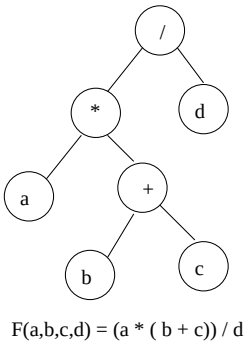

Figura 2.1: Exemplo de uma função matemática no modelo de árvore de PG.

A PG simula o processo de seleção natural que existe na natureza. Em PG, programas de computador que definem soluções para algum tipo de problema são *indivíduos* que precisam evoluir para manter sua existência. Cada etapa do processo evolutivo é denominada *geração* e o conjunto de indivíduos existentes em uma geração é chamado de *população*. A partir de uma população inicial, os diversos indivíduos são combinados entre si e, a partir de um critério de seleção, escolhem-se os que serão perpetuados para a próxima geração. Estes passos são executados por várias gerações até que se atinja um critério de parada em que se escolhe o melhor indivíduo da população como solução para o problema. Estes passos serão melhor descritos na Seção 2.2.2.

O que diferencia PG de outras técnicas evolutivas é o modo como indivíduos são modelados. PG modela soluções para problemas com estruturas de dados variáveis do tipo árvore de sintaxe. Esta forma de modelagem permite uma maior flexibilidade na busca por soluções, pois aumenta a capacidade de informação que um indivíduo pode representar. Cada nó de uma árvore pode conter símbolos que pertencem ao conjunto dos *terminais* ou símbolos que pertencem ao conjunto de *funções*. Os terminais são constantes ou variáveis dos programas e são folhas das árvores de uma PG. As funções são as operações que podem ser executadas sobre elementos do conjunto de terminais como: operações matemáticas ($+$, $-$, log, etc), operadores lógicos ($\wedge$,$\vee$,$\rightarrow$, etc), funções especiais, entre outras definidas previamente de acordo com o problema em questão. O espaço de soluções do problema é, portanto, formado por todas as combinações possíveis entre terminais e funções.

Em nosso trabalho, PG é utilizada para evoluir funções matemáticas que combinam valores de similaridade de documentos obtidos por diferentes fontes de evidência de relevância. A Figura 2.1 mostra um exemplo de uma função matemática modelada como uma árvore de PG.

2.2.1 Operadores Genéticos

PG é construída a partir de uma simulação dos operadores genéticos existentes na natureza: *cruzamento, mutação, reprodução, avaliação e seleção.* Indivíduos são modificados por estas operações seguindo um algoritmo iterativo, também baseado nos conceitos da natureza, chamado de *algoritmo evolutivo.* O objetivo de uma PG é desenvolver indivíduos presentes em uma população utilizando operadores genéticos a fim de obter um conjunto de indivíduos mais capacitados com o passar das gerações.

Criação da população inicial

O primeiro passo a ser dado pelo processo evolutivo da PG é a criação de uma população inicial. Esta população é obtida geralmente através de árvores geradas aleatoriamente, combinando os diversos terminais e funções de modo a atingir um número de indivíduos pré-determinado. Tradicionalmente uma árvore é preenchida inicialmente por um elemento do conjunto de funções, que é a raiz da árvore, e para cada um dos argumentos da função utiliza-se tanto um elemento do conjunto de funções quanto um elemento do conjunto de terminais até uma profundidade máxima pré-estabelecida.

Um trabalho proposto por Daida, em [9], mostra que a escolha das funções e terminais que compõem os indivíduos da população inicial afeta diretamente a capacidade de resolução do problema com PG. Portanto, a qualidade da população inicial deve ser levada em consideração e esta população deve ser composta por soluções que sejam uma amostra representativa do espaço de soluções possíveis. Dessa forma, a PG pode obter melhores resultados e ampliar o espaço de busca por boas soluções.

Existem diversos métodos que podem ser aplicados para criação da população inicial. Dentre eles destacam-se os métodos *grow, full* e *ramped-half-and-half.* O método *grow* permite a criação de árvores de tamanho variável. A escolha de cada nó é feita de maneira aleatória entre funções e terminais respeitando uma profundidade máxima pré-estabelecida. Ao inserir um terminal em um nó da árvore nenhum outro elemento poderá ser inserido a partir desse nó e quando o penúltimo nível de profundidade for atingido por uma função somente poderá ser inserido elementos do conjunto dos terminais.

O método *full* prevê o preenchimento completo de todos os nós da árvore até sua profun-

didade máxima. Para todos os nós anteriores ao último nível permitido são escolhidas funções aleatoriamente. Ao último nível são escolhidos aleatoriamente elementos do conjunto dos terminais para preencher as folhas das árvores.

O método *ramped-half-and-half* é um método híbrido entre os dois anteriores. Este método prevê a criação da população inicial com 50% dos indivíduos pelo método *full* e os restantes 50% de indivíduos pelo método *grow*. Este método é uma tentativa para aumentar o potencial inicial da capacidade de busca por boas soluções aumentando a diversidade inicial dos indivíduos.

Cruzamento

O cruzamento[2] é a operação genética responsável por realizar a combinação entre dois indivíduos de uma mesma população e assim gerar dois novos indivíduos que obtém por herança características de seus pais.

A primeira etapa da operação de cruzamento é a escolha dos pares de indivíduos que sofrerão o processo. Essa escolha pode ser de forma aleatória, proporcional à capacidade de cada indivíduo resolver o problema ou inversa, em que os indivíduos mais capazes fazem par com os indivíduos menos capazes.

A segunda etapa é a operação de troca de material genético em que os pares de indivíduos são escolhidos e combinados a fim de gerar um novo par de programas. A troca de material genético se dá através da troca de sub-árvores escolhidas aleatoriamente nos indivíduos pais. Para que o cruzamento tenha validade é necessário que os elementos do conjunto de funções apresentem a capacidade de receber como argumento qualquer elemento do conjunto de funções e do conjunto de terminais, caso contrário, restrições deverão ser implementadas a fim de evitar programas com erro de sintaxe.

A figura 2.2 apresenta um cruzamento entre dois indivíduos de PG, modelados como árvores de sintaxe. No exemplo, após a seleção dos pares, duas sub-árvores foram escolhidas aleatoriamente e trocadas formando dois novos indivíduos filhos.

Mutação

A mutação é uma propriedade que garante a diversidade de indivíduos em uma população. É executada após a operação de cruzamento sobre uma porcentagem pré-definida de indivíduos es-

[2]Tradução para *crossover*

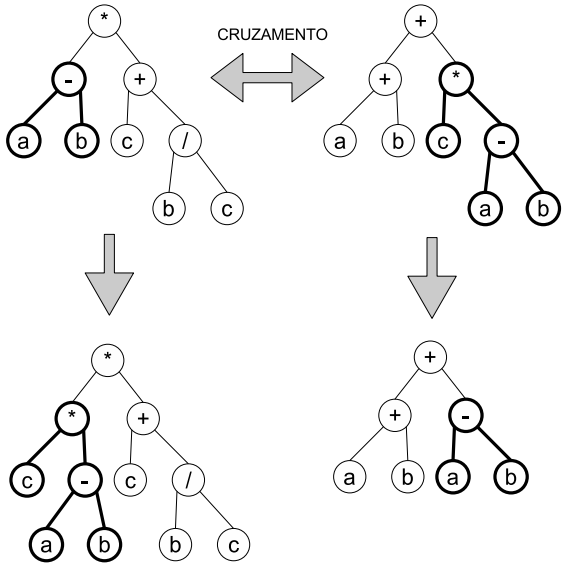

Figura 2.2: Exemplo de um cruzamento entre árvores de PG.

colhidos aleatoriamente. Os indivíduos escolhidos para se submeterem ao processo de mutação têm suas características originais alteradas de maneira aleatória permitindo que novas características possam ser inseridas e propagadas entre os indivíduos de uma população.

Existem três técnicas utilizadas com freqüência em sistemas de PG:

- *Substituição de sub-árvore*:um nó de uma árvore é escolhido aleatoriamente e a sub-árvore é substituída por uma outra gerada aleatoriamente[3].

- *Inserção de sub-árvore*: uma das folhas da árvore é escolhida aleatoriamente e é inserido nessa folha uma sub-árvore gerada aleatoriamente.

- *Troca de sub-árvore*: dois nós de uma mesma árvore são escolhidos aleatoriamente e suas respectivas sub-árvores são trocadas.

A mutação serve para manter a diversidade da população, porém, apresenta um potencial destrutivo muito grande. Com a mutação há a possibilidade de que boas soluções recém criadas sofram deteriorização na sua capacidade devido a mudanças aleatórias em seus componentes. Por esta razão, a porcentagem de indivíduos permitidos para sofrer mutação é tipicamente baixa.

[3]Geralmente os algoritmos de criação de indivíduos da população inicial é utilizado para gerar a nova sub-árvore

Avaliação

A avaliação dos indivíduos em um sistema de PG é necessária para simular o comportamento de competição entre os organismos na natureza. Os indivíduos de uma determinada geração são avaliados de acordo com sua capacidade de resolver o problema proposto. As avaliações são feitas através de uma *função de aptidão* que atribui um valor numérico a cada indivíduo presente na população. Este valor representa quão bem um indivíduo resolve o problema proposto e é chamado de valor de *aptidão* do indivíduo.

A avaliação dos indivíduos de uma população é feita a partir de um conjunto de dados de entrada que são instâncias reais de um problema proposto. Estes dados são chamados de dados de treinamento ou *casos de aptidão*. Os casos de aptidão são um conjunto de instâncias reais, coletados previamente para servirem como entrada para o processo evolutivo e devem ser instâncias estatisticamente representativas do problema que se deseja tratar. Cada instância dos casos de aptidão possui valores de entrada de um problema proposto e os valores corretos esperados na saída dos programas gerados por PG. As funções de aptidão devem atribuir melhores valores a programas que obtém resultados que mais se aproximam da saída esperada.

A função de aptidão de um sistema de PG é um dos mecanismos mais importantes no processo evolutivo, pois é a responsável por identificar e diferenciar a capacidade de resolução dos indivíduos de uma determinada população, guiando o processo de busca. Em sistemas de PG em que a função de aptidão não é boa o suficiente para identificar as capacidades dos indivíduos, pode haver um atraso muito grande na busca por boas soluções, inviabilizando a aplicação de PG como abordagem para encontrar a solução de um problema.

Seleção e Reprodução

Após o cruzamento, mutação e avaliação dos indivíduos de uma PG, é necessário que novos indivíduos sejam selecionados para compor a população da próxima geração. Dentre os métodos de seleção que podem ser aplicados temos:

- Escolha de indivíduos por ordem de aptidão onde os melhores n indivíduos serão passados para a próxima geração.
- Seleção aleatória de indivíduos sem levar em consideração a ordem da aptidão, método conhecido como *roleta-russa*.

- Seleção de indivíduos feita proporcionalmente ao valor de aptidão, método conhecido por *roleta-russa-ponderada*, que dá maior probabilidade de sobrevivência aos indivíduos de acordo com seus valores de aptidão

O método roleta-russa-ponderada foi o escolhido para realização dos experimentos da presente dissertação.

Após a seleção de indivíduos a serem transferidos para a próxima geração ocorre a reprodução. Esta operação genética é responsável por separar os indivíduos que sofreram cruzamento dos indivíduos selecionados e transferir para a próxima geração sem sofrer mudanças na estrutura.

2.2.2 Algoritmo Evolutivo

As operações genéticas são executadas sobre um algoritmo evolutivo. O algoritmo utiliza os operadores genéticos de forma a simular um processo de seleção natural em que os indivíduos são soluções para um determinado problema. Os passos do algoritmo evolutivo são apresentados abaixo:

1. Gera população inicial aleatoriamente ou com indivíduos definidos pelo usuário.
2. Avalia os indivíduos da população através da função de aptidão.
3. Seleciona p indivíduos como melhores pais e copia para a próxima geração.
4. Aplica as operações genéticas sobre os indivíduos da população.
5. Avalia todos os novos indivíduos pela função de aptidão.
6. Seleciona os f melhores filhos e copia para a próxima geração.
7. Caso atinja o critério de parada, passa para a próxima instrução. Caso contrário, substitua a população atual pelos indivíduos da próxima geração e volte para a instrução 2.
8. Apresente os melhores indivíduos como solução para o sistema

O critério de parada pode se dar através da escolha de um número máximo de gerações, definido previamente como parâmetro da PG. Outros parâmetros da PG devem ser configurados de forma adequada para que o sistema trabalhe de maneira eficiente, tanto mantendo a qualidade

das soluções finais quanto reduzindo o tempo máximo de processamento do algoritmo. O número de melhores pais p e de melhores filhos f, o número de indivíduos da população, a porcentagem de indivíduos que sofrem mutação por geração, são exemplos de parâmetros que precisam ser configurados.

2.2.3 Fases da evolução

Existem diversas metodologias para aplicação de técnicas evolutivas no aprendizado de soluções. Essas metodologias devem ser aplicadas com o intuito de aprimorar a qualidade das soluções geradas em diversas execuções do processo e permitir que as soluções geradas correspondam a boas soluções para o problema. Em nosso trabalho a metodologia para evolução de soluções foi dividida em duas fases:

- A fase de *treinamento* é responsável pela evolução de indivíduos. Nesta fase o aprendizado de soluções é realizado sobre um conjunto de dados de entrada para treinamento. Estes dados correspondem a instâncias reais do problema em questão que devem representar fortemente o conjunto total de instâncias. Além disso, os dados de treinamento apresentam as respostas esperadas para cada instância utilizada nesta fase. Usando esse conjunto de dados os indivíduos aprendem quais as principais características que levam a geração de melhores soluções de acordo com a função de aptidão.

- A segunda fase da nossa metodologia é chamada de *validação*, que é responsável por verificar se os melhores indivíduos gerados na fase de treinamento são boas soluções também para um conjunto diferente de instâncias do problema. Os melhores indivíduos do treinamento são aplicados aos dados de validação, que também são instâncias do problema com seus respectivos resultados esperados, e os que obtiverem melhores resultados são escolhidos como resposta do sistema. A avaliação dos indivíduos também deve ser de acordo com a função de aptidão.

Estas duas fases são necessárias para evitar o problema conhecido como *superadaptação*. O superadaptação é um fenômeno que acontece quando indivíduos evoluídos durante o processo de treinamento ficam altamente especializados no conjunto de dados de treinamento. Assim, indivíduos que sofreram superadaptação tendem a apresentar resultados ruins em um conjunto de dados diferente, pois não são soluções genéricas o suficiente para quaisquer conjunto de

dados. Devido a estas questões, é necessário que o conjunto de dados utilizado para treinamento e para validação seja um conjunto estatisticamente representativo do conjunto total de possíveis instâncias do problema.

Capítulo 3

Combinando Evidências Usando PG

Neste capítulo abordamos os detalhes do modelo de combinação de fontes de evidência de relevância através de uma técnica evolutiva. Nós mostraremos como PG pode ser usada para gerar funções de ordenação de documentos que combinam diferentes fontes de evidência. O problema de combinação de fontes de evidência pode ser formalizado da maneira apresentada abaixo.

Considere $E = \{e_1, \ldots, e_k\}$ um conjunto de fontes de evidência de relevância que podem ser usadas para aferir um grau de relevância de um documento da Web d dada uma consulta q em uma coleção de documentos de uma máquina de busca. Tipicamente, cada fonte de evidência e_i produz um valor de similaridade $s_i(q, d)$ que permite construir uma lista ordenada de documentos de acordo com cada uma das fontes de evidência, individualmente. Nós queremos encontrar uma *função de combinação* f que combine os valores de similaridade $s_1(q, d), \ldots, s_k(q, d)$ de tal forma que seja possível gerar uma única lista ordenada de documentos que leve em consideração todas as fontes de evidência. A função de combinação f deve ser capaz de maximizar a qualidade das respostas de acordo com uma métrica de avaliação de qualidade adotada.

Nós acreditamos que é possível obter a função de combinação através de um processo de PG no qual os valores de similaridade individuais $s_i(q, d)$ são dados como terminais e o objetivo é evoluir árvores que representam uma combinação apropriada das diferentes fontes de evidência. As funções de combinação geradas durante o processo são avaliadas através de uma função de aptidão que calcula a qualidade da nova ordenação de acordo com uma medida de qualidade adotada.

Este processo de PG necessita da execução das fases de treinamento e validação, genericamente descritas no Capítulo 2. Para cada fase, foi processado um conjunto de consultas extraídas

de um log de consultas reais, com opções de submissão selecionadas por usuários ao submeterem as consultas, podendo estas ser conjunções, disjunções ou frases. Em seguida, nós ordenamos os resultados de cada consulta de acordo com cada uma das fontes de evidência adotadas e avaliamos os resultados produzidos por cada ordenação individual. Por exemplo, uma consulta conjuntiva é processada inicialmente sem uma ordenação, em seguida o resultado é ordenado por cada uma das fontes de evidência individualmente produzindo diferentes listas de resposta, uma para cada fonte a ser combinada. Nós avaliamos manualmente os resultados produzidos pelas fontes de evidência individuais para cada consulta segundo critérios de relevância e usamos esta avaliação para realizar o processo de seleção das funções de combinação. Para cada fonte de evidência $e \in E$ e cada consulta $q \in Q$ nós solicitamos a usuários reais de máquinas de busca que avaliassem as 50 primeiras respostas produzidas. O conjunto união das respostas avaliadas para cada consulta é usado como casos de aptidão do processo evolutivo. Foi assumido que esta união contém uma porção significativa dos documentos relevantes que podem ser encontrados por qualquer função de combinação que considera as fontes de evidência de relevância presentes em E. Para cada classe de consultas estudadas no presente trabalho, foi gerado um conjunto diferente de casos de aptidão.

É importante observar que este procedimento é conveniente, uma vez que permite que a avaliação das funções de combinação geradas durante o processo evolutivo se dê de forma rápida, pois a lista de resposta de cada função é comparada a um único conjunto de documentos previamente avaliados. Este procedimento evita que haja uma avaliação manual das respostas geradas por cada função de combinação criada no processo. Tal avaliação teria um custo muito alto para gerar funções de combinação, inviabilizando a aplicação de PG neste tipo de problema, pois a mesma gera centenas de funções de combinação diferentes em cada geração do algoritmo. Este tipo de restrição apareceu em trabalhos anteriores, por exemplo, em [14], onde PG foi adotada para gerar funções completas de ordenação, em vez de combinar apenas métodos já existentes para o cálculo de similaridades.

Os casos de aptidão aqui são dados relativos às consultas a serem utilizadas nos experimentos. Para cada consulta, um conjunto de respostas para cada fonte de evidência com seus respectivos valores de similaridade, um conjunto de documentos relevantes e um conjunto de documentos não-relevantes representa compõe um caso de aptidão. Estes casos apresentam um papel fundamental no método de combinação, pois são responsáveis por guiar o processo evolu-

tivo e direcionar a busca no espaço de soluções. A fase de treino seleciona, usando PG, funções de combinação que produzem bons resultados considerando a posição dos documentos relevantes e não-relevantes na ordenação final do conjunto de casos de aptidão separados para a fase de treinamento. A função de aptidão utiliza como parâmetros um conjunto de consultas, que são os respectivos casos de aptidão, e retorna um valor que representa o desempenho da função de combinação.

Durante a fase de treinamento o processo de PG é executado 20 vezes com diferentes sementes e a melhor função da última geração de cada execução é selecionada para ser avaliada na fase de validação. As 20 funções selecionadas pela fase de treinamento são avaliadas utilizando as consultas do conjunto de validação. A que obtiver melhor resultado na fase de validação é selecionada para ser implementada na máquina de busca. Em nossos experimentos esta fase é apresentada como a fase de teste, que contém um conjunto de consultas diferente das utilizadas no treinamento e validação, e representa a performance mostrada no Capítulo 4

Capítulo 4

Experimentos

Neste capítulo apresentamos e analisamos em detalhes os experimentos realizados com o objetivo de verificar a eficácia da abordagem de combinação de fontes de evidência de relevância de máquinas de busca utilizando PG. Para isto, descrevemos as fontes de evidência utilizadas, as classes de consulta estudadas, o processo de avaliação das funções de combinação e as coleções de documentos. Os resultados obtidos são apresentados em tabelas e gráficos na Seção 4.2.

4.1 Metodologia

4.1.1 Fontes de evidência combinadas

Para verificar nossa abordagem, nós realizamos experimentos usando três fontes de evidência diferentes propostas em trabalhos anteriores na literatura:(1) a concatenação dos textos de âncora, que associa a cada documento da Web a concatenação de todos os textos de âncora referentes a este documento na coleção, (2) a informação extraída das estruturas de ligação entre os documentos da Web e (3) o conteúdo textual presente nos documentos da Web.

A primeira fonte de evidência utilizada associa, para cada documento, a concatenação de todos os textos de âncora dos apontadores incidentes no documento. Como argumentado em [12], esta concatenação provê uma boa descrição dos documentos apontados e pode ser usada como uma importante fonte de evidência de relevância. Uma das vantagens desta evidência é que ela provê diferentes descrições, escritas por diferentes autores, a respeito de um único documento da Web. Esta informação pode ajudar, por exemplo, a tratar o problema da polissemia

em máquinas de busca, que consiste na possibilidade de se expressar um conceito de diversas maneiras diferentes.

Outra característica de fundamental importância a respeito da concatenação dos textos de âncora é que ela pode prover informação textual a documentos que não apresentam conteúdo textual. Documentos que não possuem texto normalmente apresentam imagens, aplicações ou gráficos para apresentar informações a seus usuários. A conseqüência direta do uso destas formas de comunicação é que coletores de documentos da Web não estão preparados para lidar com a informação textual contida nestes tipos de dados. Desta forma, a concatenação de textos de âncora que apontam para estes tipos de documentos permite que lhes sejam atribuídos informações textuais de maneira indireta. Para calcular o valor de similaridade desta fonte de evidência foi utilizado o modelo de espaço vetorial.

A segunda fonte de evidência de relevância utilizada pode ser extraída das estruturas de ligação entre páginas da Web que é uma das mais ricas fontes de informação a respeito dos documentos. Atualmente existem diversas metodologias para calcular o valor desta fonte de evidência como: Salsa [25], HITS [21] e o Pagerank [26]. Em nosso trabalho decidimos aplicar o Pagerank por ser o método mais popular. Esta fonte de informação foi utilizada por identificar a importância dos documentos da coleção analisando as conexões entre eles.

A terceira e última fonte de evidência utilizada neste trabalho é o conteúdo textual dos documentos da coleção. Esta fonte apresenta uma importância fundamental para sistemas de recuperação de informação, pois grande parte do conteúdo presente na Web se apresenta através de informação textual. Para calcular o valor de similaridade entre consultas e os documentos que a satisfazem também usamos o modelo de espaço vetorial, que é o modelo mais popular de recuperação de informação.

4.1.2 Classes de consulta

As consultas utilizadas foram divididas de acordo com dois diferentes critérios: popularidade (popular ou não-popular) e por objetivo da consulta (informacional ou navegacional). A idéia é usar o modelo de PG para verificar se a classe das consultas afeta a escolha de parâmetros das melhores funções de combinação. Um dos objetivos de nossos experimentos é verificar como cada classe de consulta afeta a qualidade das respostas de uma máquina de busca. Abaixo estão descritas as motivações para estudar as diferentes classes de consulta e na Seção 4.2 apresentamos

resultados de experimentos com estas classes.

Objetivo da consulta

Nós utilizamos o objetivo das consultas na busca por informação para separar o conjunto total de consultas entre dois tipos: navegacionais e informacionais. Nas consultas navegacionais, os usuários estão interessados em encontrar uma única página específica na Web, enquanto nas consultas informacionais os usuários estão interessados em um tópico de informação, ou seja, em buscar informação a respeito de determinado assunto que o satisfaça.

A classificação de consultas de acordo com o tipo foi realizada de forma manual. As consultas foram selecionadas de um log de um sistema de busca real e classificadas até que fossem encontradas 62 consultas de cada tipo.

Popularidade

A popularidade foi escolhida porque pode afetar a qualidade da informação extraída das fontes de evidência presentes em ambientes colaborativos como a Web, onde documentos apontam para outros documentos que se relacionam de alguma maneira. Como exemplo dessas fontes de evidência citamos as estruturas de ligação entre documentos e os textos de âncora de documentos. A hipótese inicial é que consultas populares tendem a ser relacionadas a assuntos populares ou páginas pessoais populares. Dessa forma, informações contidas nas estruturas de ligação entre documentos e o texto de âncora podem ser usadas de maneira mais apropriada para as consultas populares do que para as não-populares.

4.1.3 Função de Aptidão

A função de aptidão utilizada no processo evolutivo deve ser capaz de selecionar as melhores funções de combinação dentre as presentes na população de uma geração. Esta seleção deve ser feita baseada nos resultados obtidos por um pequeno conjunto de consultas adotadas na fase de treinamento e validação, levando em consideração apenas uma parte dos documentos relevantes avaliados para uma consulta. Estas propriedades são importantes para permitir que o processo evolutivo encontre boas funções de combinação com o mínimo de esforço para realização do treinamento. Métricas diferentes foram adotadas para as consultas informacionais e navegacionais devido às diferenças inerentes a cada tipo de consulta.

Consultas informacionais

A avaliação de funções de ordenação para consultas informacionais na Web tem sido estudada por muitos autores, como [34, 3, 16, 19]. A principal restrição à avaliação de máquinas de busca na Web é a falta de um conjunto de documentos relevantes completo para cada consulta. De fato, o número de documentos relevantes é geralmente bem menor que o total de documentos. Este cenário é similar ao da avaliação de funções de combinação de fontes de evidência durante as fases de treinamento e validação do processo evolutivo. Neste processo, é necessário avaliar antecipadamente um conjunto de respostas para cada consulta e usar avaliações de relevância dos documentos de resposta para selecionar a melhor função de combinação. Dessa forma, funções de combinação geradas durante o processo evolutivo podem inserir novos documentos nas primeiras posições da lista ordenada de respostas que não foram avaliados previamente.

Métricas para avaliar funções de ordenação de respostas neste cenário foram apresentadas em [3, 16]. Tais métricas devem ter a propriedade de proporcionar uma comparação justa entre as funções de ordenação avaliadas, usando somente uma avaliação parcial das respostas para cada consulta. Métricas tradicionais de avaliação, como MAP e precisão no topo das respostas [1] não são boas opções para a avaliação de sistemas neste cenário, pois elas não levam em consideração a ocorrência de documentos não-avaliados no conjunto de documentos de resposta.

A métrica adotada em nossos experimentos para o processo evolutivo foi o *bpref-10* [3], que foi desenvolvido para comparar sistemas de recuperação de informação quando somente uma única parte do conjunto de respostas para uma consulta foi avaliado. Os autores mostraram que esta medida é similar ao MAP para coleções onde todo o conjunto de documentos foi avaliado e que os resultados se mantém estáveis à medida que documentos são retirados da lista ordenada de respostas, simulando a não-avaliação destes documentos. Ao usar bpref-10 com uma parte dos documentos de resposta avaliados, as conclusões a respeito do desempenho relativo do sistema tendem a ser as mesmas que seriam obtidas com a avaliação total dos documentos de resposta. A principal diferença entre bpref-10 e outras métricas de avaliação de respostas como o MAP é que o mesmo não leva em consideração os documentos não avaliados.

O bpref-10 é calculado pela seguinte fórmula:

$$bpref_10 = \frac{1}{R}\sum_{r=1}^{R} 1 - \frac{Irrelevant_R(r)}{R+10}$$

onde R é o número de documentos avaliados como relevantes, $Irrelevant_R(r)$ é o número de documentos avaliados como não-relevantes posicionados acima de r entre os $R+10$ documentos avaliados como não-relevantes retornados pelo sistema.

No intuito de avaliar as novas respostas geradas pelo processo de PG, nós usamos a mesma função usada como função de aptidão. Para manter a confiabilidade dos resultados nós usamos a precisão média, MAP[1], para avaliar a qualidade das respostas finais produzidas. Os resultados foram mostrados na Seção 4.2.

Consultas Navegacionais

Para consultas navegacionais nós usamos a média do valor de MRR para cada consulta como função de aptidão. O MRR(Mean Reciprocal Ranking) é uma métrica adotada para consultas navegacionais em experimentos de trabalhos na conferência TREC. A equação 4.1 mostra como calcular o MRR.

$$MRR(QS) = \frac{\sum_{\forall q_i \in QS} \frac{1}{PosRelAns(q_i)}}{|QS|} \tag{4.1}$$

onde QS é o conjunto de consultas, $PosRelAns(q_i)$ é a posição da primeira resposta relevante na lista ordenada de documentos para uma consulta dada q_i. Esta métrica apresenta valores que variam entre 0 e 1, na qual maiores valores são atribuídos a funções de combinação que colocam o documento correto mais próximo do topo da lista ordenada de respostas e menores valores quando as funções colocam o documento correto mais distante do topo da lista de respostas. A avaliação de consultas navegacionais é mais simples, uma vez que existe apenas uma resposta correta a ser encontrada e usar a posição deste documento na lista de respostas para calcular o MRR.

4.1.4 Coleções de referência

A abordagem de combinação de fontes de evidência utilizando PG foi avaliada neste trabalho utilizando duas bases de dados distintas . A primeira base de dados é composta por páginas da Web Brasileira, chamada WBR03, e consultas submetidas ao TodoBR[2], que é uma máquina de busca real da Web Brasileira. A segunda base de dados utilizada foi o LETOR [18], uma coleção

[1] Mean Average Precision

[2] TodoBR é uma marca da Akwan Information Technologies, que foi adquirida pelo Google em Julho de 2005

de referência publicada pela Microsoft Research Asia para ser usada como referência em estudos de aprendizagem automática de funções de ordenação de documentos.

Coleção WBR03

A WBR03 é uma coleção composta por 12,020,513 de páginas da Web Brasileira usada pelo TodoBR, contendo 139,402,245 de apontadores conectando estas páginas e apresenta aproximadamente 60 Gb de texto puro (sem considerar tags HTML). O TodoBR tem sido amplamente usado em experimentos anteriores relacionados a busca na Web. A respeito de seu tamanho reduzido, se comparada a bases de dados de máquinas de busca atuais, esta coleção apresenta um número de páginas próximo ao da coleção de documentos VLC2, adotada na Web TREC, com a vantagem de possuir um log de consultas de usuários reais.

A Tabela 4.1 apresenta algumas estatísticas sobre a WBR03 que podem ser úteis para entender os resultados apresentados em nossos experimentos. O número de apontadores válidos indica que a WBR03 possui um grande conjunto de páginas conectadas, provendo uma boa quantidade de informação para métodos de análise das estruturas de ligação entre documentos e métodos de análise das informações presentes nos textos de âncora das páginas. Esta coleção representa uma porção considerável da comunidade da Web Brasileira, que provavelmente é tão diversa em conteúdo e estruturas de ligação quanto a Web mundial. Outra vantagem em utilizar o TodoBR é a capacidade de adotar consultas reais e separá-las de acordo com sua popularidade. Dado que nossa base de dados apresenta uma diversidade de conteúdo e de conexões entre páginas tão diversa quanto a Web global, nós acreditamos que os resultados atingidos por nossa abordagem possam ser facilmente aplicados em outras coleções de referência.

Número de Páginas	12,020,513	Número de Hosts	999,522
Número de Domínios	141,284	Número de Apontadores	139,402,245
Média de Texto por Página	5kb		
Tamanho Total de Texto	60 Gb		

Tabela 4.1: Estatísticas sobre a coleção WBR03.

LETOR

O LETOR é uma base de dados de referência para experimentos voltados à aprendizagem automática de funções de ordenação em máquinas de busca publicada em [18]. O LETOR apre-

senta duas coleções para realização de experimentos: OHSUMED e TREC(TD2003 e TD2004), além de apresentar ferramentas para avaliações e dois métodos de aprendizagem automática de funções usados para comparação. As bases de dados contém informações relativas a consultas presentes em um conjunto de consultas. Para cada consulta, o LETOR apresenta listas ordenadas de respostas para diferentes funções de ordenação de documentos contendo os respectivos valores de similaridade.

Dentre as funções de ordenação que são parâmetros para experimentos sobre o LETOR, temos 25 aplicadas na coleção OHSUMED e 44 funções aplicadas sobre a coleção da TREC. Na coleção OHSUMED foram realizadas avaliações de relevância das listas de resposta seguindo um critério ternário de relevância. Já nas consultas executadas sobre as coleções da TREC, apenas uma avaliação binária foi executada em que cada documento pode ser relevante ou não-relevante. Em nossos experimentos, utilizamos apenas as coleções da TREC(TD2003 e TD2004) para analisar os resultados da abordagem de combinação de fontes de evidência usando PG. Dado que a coleção OHSUMED não apresenta as mesmas características de conectividade entre os documentos de uma base de dados extraída da Web e as fontes de evidência presentes não permitem estabelecer uma comparação com as utilizadas na WBR03, preferimos não executar experimentos sobre a OHSUMED. Entretanto, a abordagem de combinação apresentada aqui pode ser facilmente executada sobre a coleção OHSUMED provavelmente obtendo bons resultados. As buscas avaliadas para a TD2003 e TD2004 usam a coleção .GOV, que foi baseada em uma coleta realizada em Janeiro de 2002 em sítios da Web do domínio `.gov`. Existem nesta coleção 1,053,110 documentos com 11,164,829 apontadores.

Nossa abordagem pode ser aplicada sobre bases de dados genéricas e utilizando quaisquer fontes de evidência de relevância que possam ser extraídas de documentos destas coleções. No entanto, para manter um comparativo em relação às análises feitas sobre a coleção WBR03, nós utilizamos apenas as 3 funções de ordenação que são equivalentes às estudadas no presente trabalho. A Tabela 4.2 apresenta as três fontes de evidência disponibilizadas pelo LETOR estudadas, contendo informações como o identificador da fonte de evidência e os trabalhos usados como referência para extrair as fontes dos documentos da coleção.

Fonte	ID LETOR	Referência
tfidf do texto	32	[1]
tfidf do âncora	33	[1]
PageRank	14	[26]

Tabela 4.2: Fontes de evidência utilizadas extraída das coleções TREC no LETOR.

4.1.5 Configuração dos experimentos

Os experimentos do presente trabalho foram divididos em duas partes, uma para cada coleção de referência utilizada. A primeira parte corresponde aos experimentos na WBR03, coleção utilizada por uma máquina de busca real da Web, contendo um log de consultas submetidas por usuários reais do sistema. A segunda parte corresponde aos experimentos sobre o LETOR, necessários para validar a aplicação da técnica de programação genética em uma base de dados pública.

Em ambos os experimentos os parâmetros de PG genericamente descritos no Capítulo 2 foram escolhidos através de pré-experimentos e, com o intuito de permitir a reprodução de nossos resultados, estão dispostos na Tabela 4.3. Para mais detalhes a respeito da escolha de valores dos parâmetros de PG, veja [2, 22]. A seguir, uma descrição resumida sobre cada parâmetro da tabela 4.3:

- *Número máximo de gerações*: o critério de parada utilizado.
- *Tamanho da população*: o número máximo de indivíduos que compõem uma população em cada geração.
- *Número de melhores pais*: tamanho do subconjunto de indivíduos selecionados para fazer parte da próxima geração sem competir com os indivíduos gerados no cruzamento e mutação.
- *Probabilidade de mutação*: porcentagem dos indivíduos que irão sofrer mutação.
- *Altura máxima de árvore inicial aleatória*: altura máxima que uma árvore da população inicial pode obter.
- *Altura máxima da sub-árvore mutante*: altura máxima que uma sub-árvore mutante pode ter..
- *Método de seleção*: define o critério de seleção a ser usado no processo evolutivo.

Parâmetros
Número máximo de gerações **40**
Tamanho da população **400**
Numero de melhores pais **120**
Probabilidade de mutação (%) **2**
Altura máxima de árvore inicial aleatória **3**
Altura máxima da sub-árvore mutante **3**
Método de seleção **roleta-russa-ponderada**
Política de escolha de pares **aleatório**
Método de inicialização **grow**
Critério de parada **número de gerações**

Tabela 4.3: Parâmetros adotados nos experimentos parciais

- *Política de escolha de pares*: define o critério para escolha dos pares de indivíduos na operação de cruzamento.
- *Método de inicialização*: método para criação da população inicial.
- *Critério de parada*: define o critério de parada usado no processo evolutivo.

Os elementos do conjunto de funções foram: $+, -, *, /$ e *log* (soma, subtração, multiplicação, divisão e logaritmo respectivamente), mesmos elementos dos experimentos realizados por Fan em [14, 15]. A fase de treinamento do processo evolutivo foi repetida 20 vezes, cada rodada com um número limitado de gerações, conforme Tabela 4.3. Para cada rodada de treinamento a melhor resposta da última geração foi escolhida para passar a segunda fase da evolução. Na fase de validação, cada um dos 20 melhores indivíduos foram avaliados utilizando um conjunto de dados diferente e a melhor função foi escolhida como solução para o problema em questão.

Uma terceira fase, chamada de fase de teste, que não faz parte das fases da evolução é necessária para verificar a qualidade de todo o processo evolutivo. Nesta fase, um novo conjunto de instâncias do problema em questão, disjunto dos conjuntos de treino e validação, é utilizado para testar o desempenho atual da solução obtida ao final do processo evolutivo.

WBR03

Para prover os dados de entrada do método de PG, foram extraídas, para cada uma das quatro classes de consultas analisadas(informacional popular, informacional não-popular, navegacional popular e navegacional não-popular), 92 consultas do log do *TodoBr*, que é composto por 11,246,351 consultas. O conjunto de consultas foi dividido aleatoriamente em 37 consultas para

treinamento, 25 para validação e 30 consultas para teste. Todas as avaliações de relevância realizadas para cada consulta foram feitas por usuários externos, familiarizados com as páginas da Web brasileira.

O processo de PG foi executado 20 vezes com diferentes sementes de geração de números aleatórios. Em cada execução foi escolhida a melhor função gerada no treinamento, aquela com maior pontuação na função de aptidão (para as informacionais o bpref-10 e para navegacionais o MRR) ao final da última geração, selecionando 20 funções. Em seguida, estas funções foram avaliadas com o conjunto de consultas da fase de validação e a que obteve melhor resultado com a função de aptidão é passada para a fase de teste, necessária para prover os resultados finais de nossos experimentos. Na fase de teste, foram avaliadas as 10 primeiras respostas do topo de cada lista ordenada de documentos para cada função de combinação de fontes de evidência utilizadas[3]. O número de 10 consultas foi escolhido devido a característica dos usuários de máquinas de busca de procurarem as respostas relevantes apenas nas dez primeiras posições da lista ordenada de resposta.

LETOR

As coleções de referência TD2003 e TD2004 utilizadas pelo LETOR e adotadas na execução de nossos experimentos não separam o conjunto de consultas em diferentes classes, impedindo a análise do comportamento do método de combinação em diferentes tipos de consulta. Dessa forma, só foi possível realizar experimentos com consultas do tipo informacional. A coleção TD2003 apresenta 5 conjuntos de consultas, contendo 10 tópicos de informação (50 consultas ao todo). Para esta coleção foram utilizados os 3 primeiros conjuntos para treinamento, o quarto conjunto para a fase de validação e o último para a fase de testes. Já a coleção TD2004 apresenta também 5 conjuntos de consultas contendo 15 consultas em cada um (75 ao todo). A partição entre as fases de treinamento, validação e teste ocorrem da mesma maneira que na coleção TD2003, onde os 3 primeiros conjuntos foram usados para treinamento o quarto para validação e o último para teste.

O processo de PG também foi executado 20 vezes com diferentes sementes de geração de números aleatórios. Em cada execução foi escolhida a melhor função gerada no treinamento, aquela com maior valor de bpref-10, separando 20 funções para a fase de validação. Em seguida,

[3]Inclui-se aí as listas geradas pelas funções para comparação de resultados

estas funções foram avaliadas com um conjunto diferente de consultas na fase de validação e a melhor foi passada para a fase de teste.

4.2 Resultados

Antes de apresentar e discutir os resultados, nós mostramos a Tabela 4.4 contendo estatísticas a respeito do número de palavras distintas no conteúdo textual e na concatenação dos textos de âncora dos documentos relevantes para cada classe de consultas estudada na base de dados WBR03. Estas estatísticas são úteis para mostrar algumas características particulares de cada classe de consultas. Por exemplo, o conteúdo textual apresenta mais palavras distintas em documentos relevantes para as consultas informacionais do que para as consultas navegacionais. Pode-se ver também que a informação no texto de âncora aumenta com a popularidade e é mais freqüente em documentos relevantes de consultas navegacionais.

Foi avaliado também o comportamento do Pagerank, que é afetado pela popularidade das consultas, obtendo valores altos para documentos relevantes de consultas populares quando comparado aos valores obtidos por consultas não-populares. Os valores de Pagerank tendem também a ser mais altos para documentos relevantes de consultas navegacionais se comparados às consultas informacionais. Este comportamento é esperado, uma vez que consultas navegacionais normalmente estão a procura das páginas principais de sítios da Web. Uma análise final das estatísticas de Pagerank mostra que os valores em média para documentos relevantes de consultas navegacionais populares são mais altos, em contrapartida, os documentos relevantes de consultas informacionais não-populares apresentam em média os valores mais baixos.

Classe da Consulta	Tamanho Médio do Vocabulário	
	Texto	Âncora
Navegacional Popular	151	131
Navegacional Não-Popular	116	42
Informacional Popular	356	10
Informacional Não-Popular	476	5

Tabela 4.4: Número médio de palavras distintas no conteúdo textual e concatenação de texto de âncora dos documentos relevantes das quatro classes de consulta estudadas.

Foram adotados como métodos de referência em nossos experimentos três diferentes estratégias discutidas no Capítulo 1: a melhor combinação linear (BLC[4]), que foi adotada em [35],

[4]Best Linear Combination

o esquema de combinação proposto em [8], que apresenta uma estratégia para combinar fontes de evidência independentes com fontes de evidência dependentes de consulta (referenciado como SIGMOID) e o arcabouço de combinação proposto em [4], que aplica Redes Bayesianas(BN[5]) como uma abordagem de combinação. Estes métodos de referência foram escolhidos para serem modelos de referência do desempenho obtido por nossa abordagem.

O BLC realiza uma combinação linear dos valores obtidos por cada fonte de evidência, assinalando um peso para cada uma delas na combinação. O peso de cada fonte é ajustado através de experimentos. O método SIGMOID é uma função que ajusta o valor obtido pelas fontes de evidência independentes de consultas em relação aos valores das fontes dependentes de consulta (em nosso caso o conteúdo textual e os textos de âncora). O método SIGMOID obtém a combinação através de um processo de ajuste de parâmetros que necessita de uma fase de treinamento. Os métodos BLC e SIGMOID utilizam o mesmo conjunto de treinamento do método de PG. Já o método BN não necessita de fases de treinamento e utiliza uma única função de combinação para os quatro tipos de consulta estudados, com isso o método BN tende a obter desempenhos muito baixos. BN foi escolhido como método de referência para comparar seu desempenho com os métodos baseados em treinamento. Finalmente, os resultados mostrados pelo método de PG nesta seção são as medianas dos valores obtidos quando realizamos os experimentos de estabilidade em diferentes rodadas, presentes na Seção 4.2.1.1.

Os experimentos apresentados nesta seção são divididos em dois grupos para as duas coleções estudadas: WBR03 e LETOR.

4.2.1 WBR03

4.2.1.1 Estabilidade em diferentes rodadas

O processo de PG descrito aqui necessita de uma geração aleatória da população inicial adotada no processo evolutivo. Dessa forma, o resultado final obtido varia de acordo com a semente do gerador de números aleatórios adotado no processo para produzir a população inicial. Uma questão que nasce a partir deste cenário é: qual o impacto sobre o resultado final da aleatoriedade da população inicial? Para responder esta pergunta, foram executados experimentos variando as sementes e estudando a estabilidade do resultado final em diferentes execuções de todo o processo evolutivo. Para cada classe de consulta, todo o processo evolutivo, incluindo as fases

[5]Bayesian Network

de treinamento e validação, foi rodado 20 vezes e os resultados são apresentados pelas Figuras 4.1 a 4.4.

Estas figuras mostram os resultados de 20 execuções do processo de PG e compara tais resultados aos obtidos pelos 3 métodos de referência adotados (BLC, SIGM e BN). Note que os valores de referência são constantes, dado que estão incluídos apenas para comparação. Os resultados mostram que a PG se mantém bem estável a medida que varia a população inicial. A rodada executada cujo resultado aparece no ponto médio dos resultados foi a utilizada para comparação com os métodos de referência em nossos experimentos. Note que a comparação relativa entre PG e os métodos de referência não muda se nós tomarmos qualquer rodada específica para comparação, com PG obtendo melhores resultados que os métodos de referência para consultas navegacionais e obtendo resultados similares quando processamos consultas informacionais. Uma comparação detalhada entre PG e os métodos de referência é feita na Seção 4.2.1.3.

A Figura 4.1 apresenta os resultados obtidos para o conjunto de consultas navegacionais populares. Neste caso, o pior resultado obtido por PG obteve um valor de MRR de 0.75, enquanto o melhor método de referência foi o BLC que obteve apenas 0.57. A variação dos resultados de PG para esta classe de consultas foi de 0.75 a 0.84. A Figura 4.2 mostra os resultados obtidos para as consultas navegacionais não-populares. Novamente, a pior rodada do processo evolutivo foi 0.64 enquanto que o melhor método de referência foi de 0.53. A variação de PG no conjunto de consultas navegacionais não-populares foi de 0.64 a 0.75.

A Figura 4.3 mostra os resultados obtidos para o conjunto de consultas informacionais populares. Neste caso, o pior resultado obtido por PG atingiu um valor de bpref-10 de 0.488 e o melhor resultado obteve 0.498. Quando aplicamos o *teste t* de significância para comparar os resultados de BLC e SIGM, concluímos que a diferença entre os métodos em todas as rodadas não são significativas ao nível 95% de confiança. A Figura 4.4 mostra os resultados obtidos para consultas informacionais não-populares. Neste caso, o pior resultado obtido por PG foi 0.492 e o melhor 0.508. Novamente, o ganho obtido por PG, BLC e SIGM em todas as rodadas de acordo com o teste t não foi significativo.

A conclusão mais importante a que chegamos ao analisar os resultados produzidos por diversas rodadas de todo o processo de PG é que a variação das sementes para geração da população inicial aleatória não afeta de maneira significativa os resultados. Estas conclusões permitem que o método possa ser aplicado na prática.

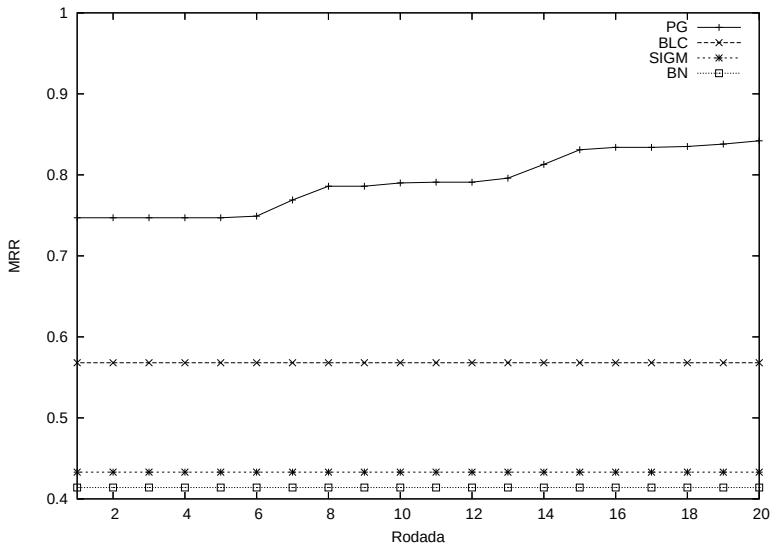

Figura 4.1: Estabilidade do processo de PG em 20 diferentes rodadas para consultas Navegacionais Populares.

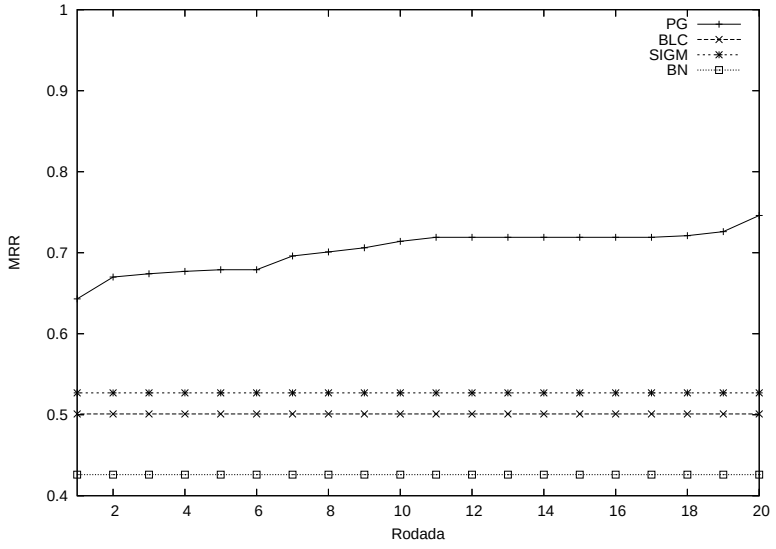

Figura 4.2: Estabilidade do processo de PG em 20 diferentes rodadas para consultas Navegacionais Não-Populares.

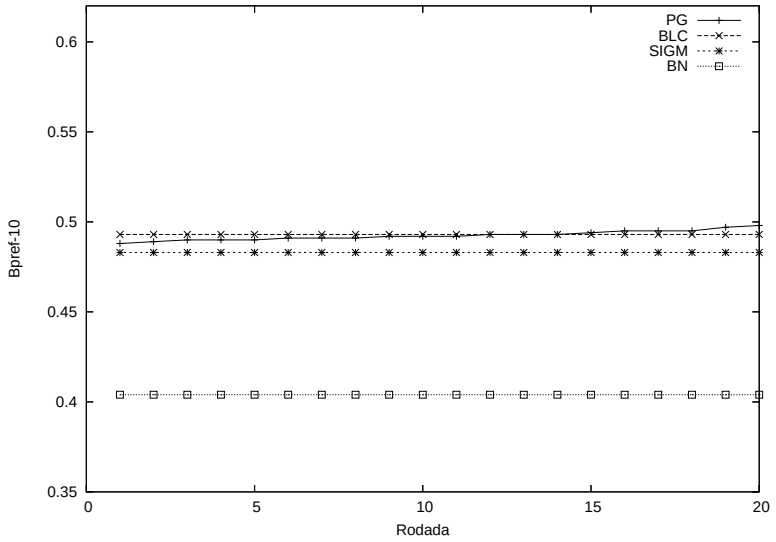

Figura 4.3: Estabilidade do processo de PG em 20 diferentes rodadas para consultas Informacionais Populares.

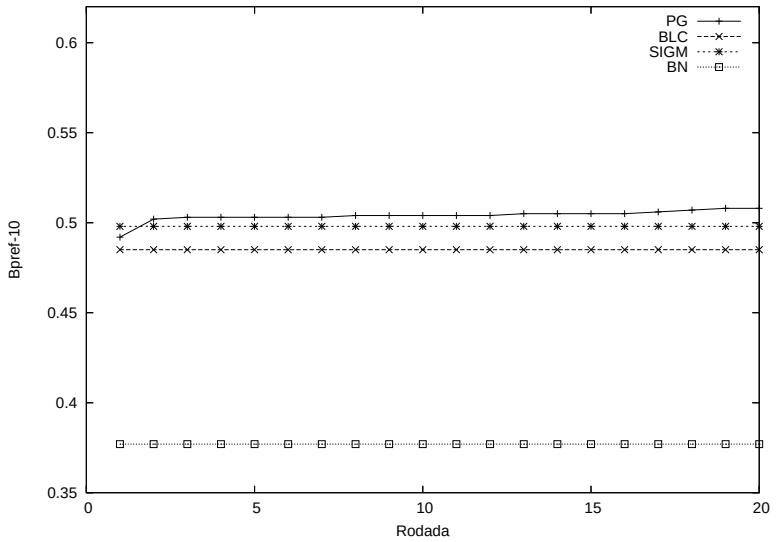

Figura 4.4: Estabilidade do processo de PG em 20 diferentes rodadas para consultas Informacionais Não-Populares.

4.2.1.2 Estabilidade em diferentes conjuntos de treinamento

No intuito de validar os métodos estudados, nós realizamos uma validação cruzada de 4 blocos usando todo o conjunto de consultas avaliadas por usuários adotadas nos experimentos de PG para geração da função de combinação. Os resultados para cada classe de consulta estudada estão presentes nas Tabelas 4.5, 4.6, 4.7 e 4.8. Como pode ser visto, os resultados relativos obtidos pelos métodos não se alteram nos 4 blocos de cada cenário, sendo o método de PG o melhor em todos os blocos analisados, exceto apenas no quarto bloco da Tabela 4.8 que mostra os resultados para consultas informacionais não-populares. As conclusões deste experimento mostram que os resultados obtidos são estáveis na coleção utilizada.

Note que a diferença entre os valores de resultados de cada bloco se altera devido às mudanças ocorridas para o conjunto de treinamento e validação de cada bloco. O uso de apenas quatro blocos pode ser considerado pequeno, porém é uma configuração adequada para esta coleção devido ao alto custo de avaliação das consultas para criação dos conjuntos de treinamento e validação. Entretanto, note que o desvio-padrão obtido é baixo o suficiente para indicar que as comparações relativas entre os métodos não muda quando variamos os conjuntos de validação e treino.

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,933	0,800	1,000	0,861	0,899	0,087
BLC	0,541	0,518	0,593	0,429	0,520	0,068
SIGM	0,382	0,454	0,503	0,449	0,447	0,050

Tabela 4.5: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas navegacionais populares. Todos os valores estão expressos em MRR.

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,797	0,870	0,657	0,634	0,739	0,113
BLC	0,398	0,348	0,433	0,134	0,328	0,134
SIGM	0,344	0,382	0,285	0,229	0,310	0,067

Tabela 4.6: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas navegacionais não-populares. Todos os valores estão expressos em MRR.

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,505	0,507	0,461	0,547	0,505	0,035
BLC	0,490	0,495	0,438	0,541	0,491	0,042
SIGM	0,495	0,488	0,437	0,535	0,489	0,040

Tabela 4.7: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas informacionais populares. Todos os valores estão expressos em bpref-10

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,534	0,620	0,710	0,516	0,595	0,089
BLC	0,532	0,615	0,703	0,533	0,596	0,081
SIGM	0,527	0,598	0,700	0,525	0,588	0,082

Tabela 4.8: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas informacionais não-populares. Todos os valores estão expressos em bpref-10.

4.2.1.3 Comparação com métodos de referência

A Tabela 4.9 apresenta os resultados para consultas navegacionais usando o método de PG e as outras métricas de combinação usadas como referência: BN, BLC e SIGM. Nós aplicamos o teste t de significância sobre os resultados e todas as diferenças entre PG e os outros métodos são significativas. PG atingiu valores de MRR próximos a 1, o que poderia ser o resultado perfeito, ou seja, colocar todos os documentos corretos no topo da lista ordenada de respostas. Para consultas populares, houve um ganho de 38% em MRR em relação ao BLC, que atingiu o segundo melhor resultado. Para consultas não-populares, o ganho é praticamente o mesmo, cerca de 37% sobre o segundo melhor método, que foi o SIGM.

Uma das possíveis causas desta diferença é a flexibilidade da abordagem de combinação por PG, que permite modelar de não linear as possíveis relações de dependência existentes entre as diferentes fontes de evidência, enquanto que as abordagens SIGM, BLC e BN só permitem combinações lineares. Um exemplo claro de dependência é a relação existente entre o Pagerank e as outras fontes de evidência. Como o Pagerank é uma fonte independente da consulta, um documento com Pagerank alto não é um indicativo claro de que ele seja um documento relevante para uma consulta. Evidências complementares como o texto do documento e os textos de âncora são necessárias neste caso. Se um documento não apresentar uma forte evidência de relevância no texto ou textos de âncora, por exemplo, então o Pagerank não pode ser levado em

consideração como fonte de evidência segura para atribuir uma estimativa de relevância para o documento. Este é apenas um exemplo de quão complexo as dependências entre as fontes de evidência podem ser, portanto a liberdade no formato que a função de combinação final pode ter permite que PG atinja combinações de fontes mais ricas que métodos como BLC e SIGM.

Consultas Navegacionais(MRR)		
Método	Populares	Não-populares
PG	0,786	0,723
BN	0,414	0,426
BLC	0,568	0,501
SIGM	0,433	0,527

Tabela 4.9: Resultados de MRR para consultas navegacionais quando combinadas as diferentes fontes de evidência com PG, BN, BLC and SIGM.

A Tabela 4.10 mostra os resultados obtidos quando processadas consultas informacionais. Neste cenário, a abordagem de PG apresenta ganhos significativos somente sobre BN, que não utiliza treinamento. Quando comparados todos os métodos baseados em treinamento, pudemos verificar que o ganho obtido por PG é pequeno e não significativo de acordo com o teste t. Uma explicação para este resultado é que a evidência contida no conteúdo textual das páginas para consultas informacionais é amplamente mais importante que as outras fontes de evidência estudadas, fazendo com que cada método de combinação apresentasse funções com o texto dominando todas as outras. A fim de entender estes resultados, nós realizamos experimentos para estudar o impacto de cada fonte isolada na combinação e observamos que o texto foi justamente a fonte de evidência mais importante para consultas informacionais. Além disso, o resultado não significativo sugere que o número de consultas informacionais utilizados nos experimentos é pequeno. Devido ao custo associado à avaliação de consultas, não foi possível realizar experimentos com um conjunto de consultas maior.

Método	Populares		Não-Populares	
	Bpref-10	MAP	Bpref-10	MAP
PG	0,493	0,185	0,504	0,331
BN	0,404	0,169	0,377	0,250
BLC	0,493	0,209	0,485	0,321
SIGM	0,483	0,199	0,498	0,334

Tabela 4.10: Resultados da combinação das diferentes fontes de evidência usando PG, BN, BLC and SIGM de acordo com as métricas Bpref-10 e MAP em consultas informacionais.

Consultas Navegacionais (MRR)		
Evidência	Populares	Não-populares
Texto	0,227	0,204
Âncora	0,141	0,387
Pagerank	0,338	0,137
Texto + Pagerank	0,512	0,280
Texto + Âncora	0,453	0,543
Âncora + Pagerank	**0,786**	0,707
Todas	**0,786**	**0,723**

Tabela 4.11: Resultados atingidos por Programação Genética em MRR de consultas navegacionais usando diferentes combinações de fontes de evidência. Todas significa a combinação usando as 3 fontes de evidência.

4.2.1.4 Impacto de cada fonte de evidência na combinação

Nós usamos a abordagem de PG para avaliar o impacto de cada fonte de evidência de relevância na qualidade final dos resultados. Com este objetivo, aplicamos o processo evolutivo separadamente para cada combinação possível das três fontes de evidência estudadas.

A Tabela 4.11 apresenta os resultados obtidos quando processamos consultas navegacionais. A tabela mostra os resultados quando usamos cada uma das evidências individualmente, para ilustrar o impacto de cada uma delas quando usadas isoladamente, e todas as possíveis combinações delas. Nos resultados isolados, consideramos cada consulta como uma consulta booleana e então ordenamos os documentos a partir das fontes de evidência individualmente. Note que as melhores fontes de evidência isoladas são o Pagerank e a Concatenação do Texto de Âncora, para consultas populares e não-populares respectivamente. Em qualquer classe de popularidade, os resultados das fontes isoladas são baixos.

Ao analisar a importância da informação textual em consultas navegacionais podemos notar que ela apresenta baixo impacto na qualidade final das respostas. Embora os valores finais de MRR atingidos para consultas não-populares usando as três fontes de evidência são ligeiramente maiores que as outras combinações, a diferença entre resultados desta combinação de fontes e a apresentada pela combinação de Pagerank e informação de textos de âncora (Âncora+Pagerank) não é estatisticamente significativo.

Na Tabela 4.12 estão apresentados os resultados obtidos por todas as possíveis combinações das três fontes de evidência estudadas para consultas informacionais. Como pode ser visto, o texto foi a melhor fonte de evidência individual no cenário de consultas informacionais. Para ambas as classes de popularidade, não houve diferença estatisticamente significativa entre a

fonte de evidência de texto isoladamente e o resultado combinando todas as fontes de evidência (Todas).

Consultas Informacionais				
Evidência	Populares		Não-Populares	
	Bpref-10	MAP	Bpref-10	MAP
Texto	0,472	0,171	0,480	0,318
Âncora	0,434	0,161	0,318	0,151
Pagerank	0,329	0,073	0,213	0,109
Texto+Âncora	0,496	0,185	0,502	0,332
Texto+Pagerank	0,477	0,174	0,482	0,320
Âncora+Pagerank	0,463	0,157	0,335	0,150
Todas	**0,493**	**0,185**	**0,504**	**0,331**

Tabela 4.12: Resultados atingidos por Programação Genética de acordo com as métricas Bpref-10 e MAP em consultas informacionais usando diferentes combinações de fontes de evidência.

Outra informação importante extraída dos experimentos é que o Pagerank apresenta um baixo impacto na qualidade dos resultados. Todas as combinações que incluem o Pagerank apresentam resultados próximos aos obtidos quando o Pagerank é retirado da combinação. Isto pode ser observado, por exemplo, quando comparamos a combinação de texto de âncora com o conteúdo textual (Âncora+Pagerank) com o resultado do texto de âncora isolado. Outro exemplo é a combinação das três fontes de evidência (Todas) com os resultados obtidos pela combinação de texto de âncora e conteúdo textual (Texto+Âncora). Entretanto, os melhores resultados são obtidos quando adicionamos texto às combinações de evidência, sendo que ele isoladamente já apresenta resultados próximos ao da melhor combinação.

Estes resultados fazem sentido, uma vez que documentos relevantes para consultas do tipo informacional tendem a apresentar uma grande quantidade de texto (para mais detalhes veja Tabela 4.4) e o objetivo dos usuários neste caso está relacionado com a busca de um conteúdo específico que satisfaça sua necessidade de informação em vez de uma página ou sítios específico na Web.

4.2.2 LETOR

4.2.2.1 Estabilidade em diferentes rodadas

Os experimentos de estabilidade que serão apresentados aqui seguem o mesmo método de execução utilizado nos experimentos executados sobre a WBR03. Eles também são necessários

para verificar se o método de PG para geração de funções de combinação de fontes de evidência não é afetado pela aleatoriedade do processo. Novamente, os experimentos de todo o processo evolutivo para combinação de fontes de evidência foram repetidos 20 vezes, variando as sementes para geração da população inicial aleatória. Os resultados são apresentados nas Figuras 4.5 e 4.6 para as coleções TD2003 e TD2004, respectivamente.

Estas figuras mostram o resultado de 20 execuções do processo de PG e compara os resultados aos três métodos de referência. É importante lembrar que os métodos estão presentes apenas para comparação, portanto são valores constantes no gráfico. O resultado escolhido para comparação direta com os métodos de referência é aquele presente no ponto médio dos resultados. Note que as curvas do método de PG não são tão estáveis quanto as curvas obtidas nos experimentos com a WBR03. Uma possível razão para este comportamento é o fato de terem sido usadas um número maior de consultas para a fase teste na WBR03 (30 consultas na WBR contra 10 da coleção TD2003 e 15 da coleção TD2004).

A Figura 4.5 mostra os resultados obtidos para o conjunto de consultas processadas sobre a coleção TD2003. Neste experimento, os resultados obtidos por PG foram equivalentes ao método BLC e melhores que os outros métodos de referência, usando o bpref-10 como métrica de avaliação de respostas. É importante observar nesta figura que a PG apresenta uma estabilidade durante as diferentes rodadas, o que pôde ser comprovado com a execução do teste t de significância entre o pior e o melhor resultado, mostrando que a diferença entre as funções geradas não são significativas. A diferença entre as rodadas de PG e o resultado do método BLC não foram significativas, fato que pode ser atribuído também ao baixo número de consultas usados na fase de teste.

Os resultados obtidos para o coleção TD2004 mostra uma estabilidade menor das funções geradas pelo método de PG. Esta instabilidade pôde ser comprovada executando o teste t de significância sobre a pior e a melhor função de combinação geradas no processo, que mostrou uma diferença singificativa entre os resultados. Novamente, a instabilidade do processo pode ser atribuída ao número reduzido de consultas usadas na fase de teste. No entanto, apesar da menor estabilidade, o método de PG continua apresentando resultados comparáveis aos métodos de referência nesta coleção em qualquer rodada do processo.

Como os experimentos de estabilidade apresentados sobre a coleção de referência LETOR apresentam resultados similares aos métodos de combinação utilizados, podemos concluir que o

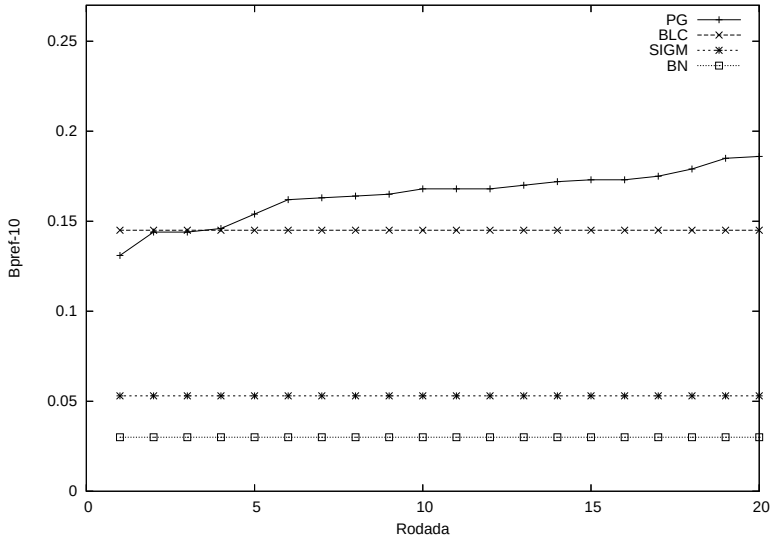

Figura 4.5: Estabilidade do processo de PG em 20 diferentes rodadas para consultas sobre a TD2003 do LETOR.

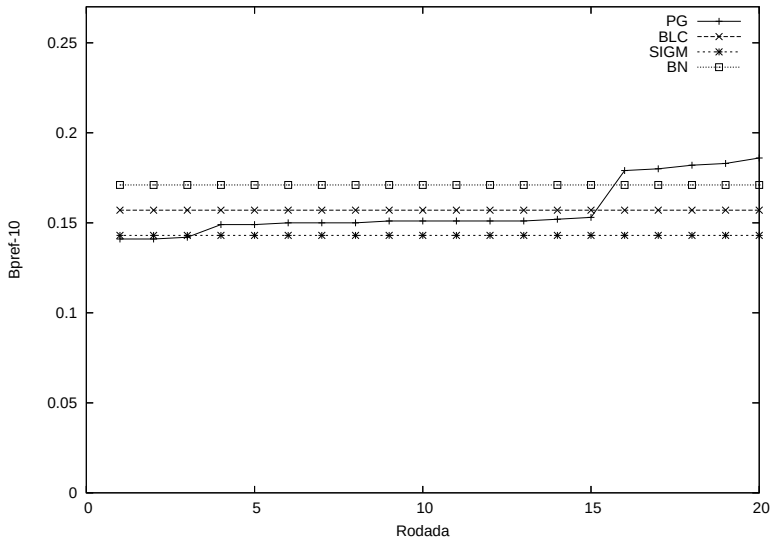

Figura 4.6: Estabilidade do processo de PG em 20 diferentes rodadas para consultas sobre a TD2004 do LETOR.

método de combinação por PG pode ser usado na prática, confirmando os resultados obtidos na Seção 4.2.1.1.

4.2.2.2 Estabilidade em diferentes conjuntos de treinamento

A fim de validar os métodos de combinação estudados sobre a base de dados LETOR, foram executados experimentos de validação cruzada de 4-blocos utilizando os conjuntos de treinamento e validação usado no processo de PG para comparação com os métodos de referência.

Os resultados para cada coleção (TD2003 e TD2004) são apresentados nas Tabelas 4.13 e 4.14. Note que na coleção TD2003, as variações dos valores de bpref-10 da PG para os diferentes blocos são estáveis. Nesta coleção, PG obteve ganhos em dois blocos. É importante observar que no bloco 2, o método BLC obteve um ganho muito elevado, fato que pode ser atribuído ao pequeno número de consultas usadas para treinamento e validação em cada bloco (30 e 10 respectivamente). Neste bloco, ao contrário do que ocorre com os outros métodos, o método de PG apresenta um valor muito baixo, que pode indicar ter havido superadaptação nesta configuração do conjunto de dados. Apesar disso, os resultados do método de PG na coleção TD2003 são comparáveis aos métodos de referência.

Na coleção TD2004, Tabela 4.14, as variações dos valores de bpref-10 de todos os métodos de combinação se mantiveram estáveis, onde PG obteve ganho sobre o método SIGM e resultados comparáveis ao método BLC. Estes resultados em conjunto com os resultados obtidos para a WBR03 sugerem que o método de PG para combinação pode ser aplicado a outras coleções de documentos que apresentem as mesmas características das coleções estudadas. No entanto, é importante observar que um conjunto pequeno de consultas pode levar a resultados mais imprevisíveis. Observando os valores de desvio-padrão obtidos notamos que as comparações relativas entre os métodos se mantém.

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,240	0,170	0,118	0,147	0,169	0,052
BLC	0,202	0,354	0,108	0,149	0,203	0,108
SIGM	0,177	0,216	0,056	0,153	0,150	0,068

Tabela 4.13: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas sobre a coleção TD2003. Todos os valores estão expressos em bpref-10.

Método	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Média	DP
PG	0,201	0,269	0,118	0,157	0,186	0,065
BLC	0,179	0,242	0,130	0,169	0,180	0,046
SIGM	0,144	0,147	0,027	0,109	0,107	0,056

Tabela 4.14: Resultados da validação cruzada sobre métodos de combinação baseados em Programação Genética (PG), melhor combinação linear (BLC), e SIGM quando processadas consultas sobre a coleção TD2004. Todos os valores estão expressos em bpref-10.

4.2.2.3 Comparação com métodos de referência

A Tabela 4.15 mostra os resultados comparativos entre o método de PG e os métodos de combinação de fontes de evidência estudados usados como referência. Foi aplicado o teste t para calcular o valor de significância das diferenças obtidas nos resultados. O teste t mostrou que o ganho relativo do método de PG sobre o segundo melhor método de referência para consultas processadas na base TD2003 foi de $0,15\%$, apesar disso, este ganho não foi significativo. Para consultas processadas na base TD2004, o melhor método de combinação foi o BN, com um valor de bpref-10 de $0,171$, com um ganho relativo de 13% em relação ao método de PG. Entretanto, também não houve ganho significativo. PG apresenta resultados similares aos métodos de referência e novamente, o reduzido número de consultas pode ter contribuído para estes resultados.

Método	TD2003		TD2004	
	Bpref-10	MAP	Bpref-10	MAP
PG	0,168	0,091	0,151	0,130
BN	0,030	0,043	0,171	0,146
BLC	0,145	0,081	0,157	0,134
SIGM	0,053	0,053	0,143	0,123

Tabela 4.15: Resultados da combinação das diferentes fontes de evidência usando PG, BN, BLC and SIGM de acordo com as métricas Bpref-10 e MAP em consultas informacionais.

4.2.2.4 Impacto de cada fonte de evidência na combinação

A fim de estudar o impacto de cada fonte de evidência, novamente aplicamos PG para testar as diversas possibilidades de combinação, como foi realizado na Seção 4.2.1.4. Aplicamos o processo evolutivo separadamente para cada combinação possível das três fontes de evidência estudadas.

A Tabela 4.16 mostra os resultados obtidos por cada fonte de evidência isoladamente, a fim de mostrar o impacto de cada uma sobre a coleção, todas as possíveis combinações de fontes de evidência dois a dois e por último a combinação com as três fontes. Nos resultados isolados, consideramos o valor de similaridade original fornecido pelo LETOR. Note que tanto na coleção TD2003 quanto na coleção TD2004 o texto de âncora é uma fonte de evidência dominante, fato que pode ser observado pela a queda na qualidade das respostas geradas com a combinação sem o texto de âncora(Texto+Pagerank). Essa característica foi identificada pelo método de PG que apresentou resultados muito próximos aos obtidos pelo texto de âncora isoladamente.

Quando uma fonte de evidência apresenta uma importância grande na coleção, pode ocorrer uma redução na complexidade das dependências entre as diferentes fontes de evidência, permitindo que funções de combinação mais simples, como as geradas pelo BLC, obtenham resultados similares a métodos de combinação mais complexos como a PG.

A tabela mostra também que a combinação entre texto de âncora e Pagerank é a melhor forma de realizar a combinação, obtendo resultados praticamente iguais à combinação com todas as fontes de evidência na coleção TD2003. No entanto, na coleção TD2004, a inserção do Pagerank na reduziu a qualidade das listas ordenadas de resposta produzidas. O texto praticamente não interfere na qualidade das respostas.

É importante destacar que a coleção do LETOR estudada aqui apresenta um conjunto reduzido de consultas disponíveis para teste em relação ao conjunto utilizado nos experimentos da WBR, gerando instabilidade nos resultados. Por esta razão os experimentos apresentam resultados inconclusivos. Porém, como o método de PG apresentou resultados similares aos demais métodos de referência, podemos verificar que a abordagem pode ser aplicada para combinação de fontes de evidência de relevância em diferentes coleções de documentos.

Evidência	TD2003		TD2004	
	Bpref-10	MAP	Bpref-10	MAP
Texto	0,019	0,024	0,071	0,058
Âncora	0,168	0,088	**0,151**	0,126
Pagerank	0,058	0,053	0,144	0,119
Texto+Âncora	0,150	0,078	**0,151**	**0,130**
Texto+Pagerank	0,058	0,057	0,071	0,057
Âncora+Pagerank	**0,170**	**0,095**	0,144	0,128
Todas	0,168	0,091	**0,151**	**0,130**

Tabela 4.16: Resultados atingidos por Programação Genética de acordo com as métricas Bpref-10 e MAP nas coleções TD2003 e TD2004 usando diferentes combinações de fontes de evidência.

Capítulo 5

Conclusões e Trabalhos Futuros

Apresentamos neste trabalho um novo estudo sobre a aplicação de Programação Genética para combinação de fontes de evidência de relevância em máquinas de busca. As fontes de evidência utilizadas neste trabalho foram o conteúdo textual, a concatenação dos textos de âncora e as estruturas de ligação entre os documentos. Trabalhos anteriores já haviam estudado PG com o objetivo de melhorar a qualidade das respostas de uma máquina de busca através da combinação de fontes de evidência [11], porém os resultados obtidos indicavam que PG não era uma boa estratégia para combinação. Analisando profundamente o trabalho proposto por Carvalho identificamos problemas na metodologia para realização dos experimentos. Dessa forma, neste trabalho realizamos experimentos mais confiáveis e mostramos que PG é uma estratégia eficiente para combinação de fontes de evidência em máquinas de busca.

Realizamos experimentos em uma máquina de busca real da Web contendo um log de consultas submetidas por usuários reais. Nós dividimos as consultas em 4 classes distintas: navegacionais populares, navegacionais não-populares, informacionais populares e informacionais não-populares. Realizamos experimentos de combinação aplicados a diferentes classes de consulta, o que se mostrou um critério útil, dado que diferentes resultados foram obtidos para cada classe. A fim de testar a qualidade da abordagem de combinação por PG, realizamos experimentos usando a coleção de referência LETOR. Outros métodos também foram testados para comparação.

Na base real, a abordagem de PG apresentou ganho de 38% sobre o segundo melhor método de combinação para consultas navegacionais não-populares. Nosso método também obteve um ganho de 37% para consultas navegacionais populares sobre o segundo melhor método de com-

binação. Em consultas informacionais, os resultados foram similares aos métodos de referência, pois o ganho não foi significante sobre os outros métodos baseados em treinamento.

Na coleção de referência LETOR, que se sub-divide em duas coleções distintas (TD2003 e TD2004), nosso método apresentou resultados similares aos demais métodos de referência, pois não foram observadas diferenças significativas entre seus resultados.

O arcabouço de PG reavaliado aqui apresenta as seguintes características:(1) gera funções de combinação apropriadas para diferentes fontes de evidência e (2) identifica a importância relativa a cada fonte de evidência permitindo modelar de maneira mais adequada possíveis dependências existentes entre as fontes em diferentes cenários de consultas ou coleções de documentos.

Em nosso trabalho, a separação das consultas em classes de acordo com a popularidade não afetou profundamente o desempenho final das funções geradas por PG. Tanto para consultas navegacionais quanto para as consultas informacionais o resultado para as diferentes popularidades se mostrou compatível. Ainda é necessário mais estudos buscando comprovar a eficiência da separação das consultas de acordo com a popularidade.

Um problema em aberto é identificar os tipos de consulta em ambientes de máquina de busca de maneira eficiente. Estudos aplicados a classificação de consultas no momento da submissão já vêm sendo realizados, porém estudos investigando as relações existentes entre as evidências das páginas na coleção e as consultas de diferentes classes são necessários.

Outro experimento necessário é a comparação da abordagem de PG com métodos de combinação mais complexos já publicados, como por exemplo os métodos baseados em *Support Vector Machine* (SVM). Estes métodos trazem duas vantagens sobre os métodos de referência utilizados neste trabalho: primeiramente há a possibilidade de investigar modelos de combinação de fontes de evidência não lineares, o que permite identificar de uma maneira mais eficiente as interdependências existentes entre as diferentes fontes de evidência. A segunda vantagem é a possibilidade de inserir uma fase de validação para a correta metodologia de experimentos de combinação, o que tornaria a comparação de PG com estes métodos mais justa.

Experimentos para avaliação do nosso método de combinação com outros conjuntos de fontes de evidência devem ser realizados como um trabalho futuro. Para isso, uma direção seria a aplicação dos experimentos apresentados neste trabalho sobre a base de dados OHSUMED que contém diferentes conjuntos de fontes de evidência e uma base de dados com uma característica diferente das duas apresentadas aqui. Assim seria possível comprovar a generalidade

da abordagem apresentada aqui.

Outra direção importante é investigar com maior profundidade a utilização de cada fonte de evidência na função de combinação gerada para entender a importância de cada uma no processo de combinação. As funções de combinação geradas são complexas e a análise da importância de cada fonte de evidência pela sua utilização não é trivial. A equação 5.1 é um exemplo de uma função de combinação para consultas navegacionais populares:

$$Comb(a,p,t) = \left\{ \frac{(p)}{\frac{((((a)+(p))+((p)+((a)+(p))))+((p)*((((p)*(6))+(a))+((a)+(p)))))+((a)+((t)+(p)))}{(a)+(((t)*((p)+(a)))+((t)+(a)))} \right. \quad (5.1)$$

onde a é a fonte de evidência de texto de âncora, p é a fonte de evidência de Pagerank e t é a fonte de evidência de texto puro. É importante observar a complexidade da combinação gerada pela programação genética. Da equação 5.1 podemos inferir que a evidência de texto puro t não foi utilizada com freqüência, e que a evidência de Pagerank p é a fonte mais importante, pois está presente como a única fonte de evidência como principal denominador. Como esta análise não é precisa, estudar estratégias automáticas para o profundo entendimento da importância de cada fonte de evidência para as diferentes funções de combinação de diferentes classes de consulta é um importante trabalho futuro.

Referências Bibliográficas

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] W Banzhaf, Peter Nordin, R E Keller, and F D Francone. *Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers, 1998.

[3] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM Press.

[4] Pável Calado, Marco Cristo, Edleno Moura, Nivio Ziviani, Berthier Ribeiro-Neto, and Marcos André Gonçalves. Combining link-based and content-based methods for web document classification. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 394–401, New York, NY, USA, 2003. ACM Press.

[5] H. Chen. Machine learning for information retrieval: neural networks, symbolic learining, and genetic algorithms. *Journal of the American Society for Information Science*, 46(3):194–216, 1995.

[6] Abdur Chowdhury, Ophir Frieder, David Grossman, and Catherine McCabe. Analyses of multiple-evidence combinations for retrieval strategies. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 394–395, New York, NY, USA, 2001. ACM Press.

[7] Nick Craswell, David Hawking, and Stephen Robertson. Effective site finding using link anchor information. In *SIGIR '01: Proceedings of the 24th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 250–257, New York, NY, USA, 2001. ACM Press.

[8] Nick Craswell, Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Relevance weighting for query independent evidence. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 416–423, New York, NY, USA, 2005. ACM Press.

[9] Jason M. Daida. Towards identifying populations that increase the likelihood of success in genetic programming. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1627–1634, New York, NY, USA, 2005. ACM.

[10] C. Darwin. John Murray, 1859.

[11] Moisés Gomes de Carvalho. Fusão de evidências de relevância através de técnicas evolucionárias, 2004.

[12] Nadav Eiron and Kevin S. McCurley. Analysis of anchor text for web search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 459–460, Toronto, July 2003.

[13] W. Fan, M. D. Gordon, and P. Path. Genetic programming-based discovery of ranking functions for effective web search. *Journal of Management Information Systems*, 21(4):37–56, 2005.

[14] Weiguo Fan, M.D. Gordon, P. Pathak, Wensi Xi, and E.A Fox. Ranking function optimization for effective web search by genetic programming: An empirical study. In *System Sciences, Proceedings of the 37th Annual Hawaii International Conference on*. IEE CNF, 2004.

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

[16] Leif Grönqvist. Evaluating latent semantic vector models with synonym tests and document retrieval. In *ELECTRA Workshop on Methodologies and Evaluation of Lexical Cohesion Techniques in Real-world Applications (Beyond Bag of Words)*, 2005.

[17] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[18] Thorsten Joachims, Hang Li, Tie-Yan Liu, and ChengXiang Zhai. Learning to rank for information retrieval (lr4ir 2007). *SIGIR Forum*, 41(2):58–62, 2007.

[19] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, New York, NY, USA, 2000. ACM Press.

[20] In-Ho Kang and GilChang Kim. Integration of multiple evidences based on a query type for web search. *Inf. Process. Manage.*, 40(3):459–478, 2004.

[21] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[22] John R Koza. *Gentic Programming: on the programming of computers by means of natural selection*. MIT Press, 1992.

[23] Joon H Lee. Combining multiple evidence from different properties of weighting schemes. Technical report, Cornell University, Ithaca, NY, USA, 1995.

[24] Joon H Lee. Analysis of multimple evidence combination. *ACM SIGIR*, 1997.

[25] R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19(2):131–160, 2001.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford Digital Library Technologies Project*, 1998.

[27] Berthier Ribeiro and Richard Muntz. A belief network model for ir. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA, 1996. ACM Press.

[28] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-4. In *Fourth Text Retrieval Conf.*, pages 73–97, 1996.

[29] Gerard Salton. Associative document retrieval techniques using bibliographic information. *J. ACM*, 10(4):440–457, 1963.

[30] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1st edition, 1983.

[31] Ilmerio Silva, Berthier Ribeiro-Neto, Pável Calado, Edleno Moura, and Nivio Ziviani. Link-based and content-based evidential information in a belief network model. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103, New York, NY, USA, 2000. ACM Press.

[32] Andrew Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.

[33] Trystan Upstill, Nick Craswell, and David Hawking. Query-independent evidence in home page finding. *ACM Trans. Inf. Syst.*, 21(3):286–313, 2003.

[34] Ellen M. Voorhees. Evaluation by highly relevant documents. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, New York, NY, USA, 2001. ACM Press.

[35] T. Westerveld, W. Kraai, and D. Hiemstra. Retrieving web pages using content, links, urls and anchors. *TREC*, 2001.

[36] Jen-Yuan Yeh, Jung-Yi Lin, Hao-Ren Ke, and Wei-Pang Yang. Learning to rank for information retrieval using genetic programming. In *SIGIR 2007*, New York, NY, USA, 2007. ACM.