Fernando Hattori

Feedback de Relevância Orientado a Termos - Um Novo Método para Ordenação de Resultados de Motores de Busca

São Paulo
2016

Fernando Hattori

Feedback de Relevância Orientado a Termos - Um Novo Método para Ordenação de Resultados de Motores de Busca

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para a obtenção do título de Mestre em Ciências

Área de Concentração:
Engenharia de Computação

Orientador: Prof. Dr. Edson Satoshi Gomi

São Paulo
2016

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Hattori, Fernando

Feedback de Relevância Orientado a Termos - Um Novo Método para Ordenação de Resultados de Motores de Busca / F. Hattori, E. Gomi -- versão corr. -- São Paulo, 2016.

80 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.RECUPERAÇÃO DA INFORMAÇÃO 2.MOTORES DE BUSCA 3.BIBLIOTECA DIGITAL I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t. III.Gomi, Edson

Fernando Hattori

Feedback de Relevância Orientado a Termos - Um Novo Método para Ordenação de Resultados de Motores de Busca

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para a obtenção do título de Mestre em Ciências

São Paulo
2016

AGRADECIMENTOS

Ao Professor Doutor Edson Satoshi Gomi, pela orientação e pelo constante estímulo transmitido durante todo o trabalho.

Aos amigos Robero Fray da Silva, Eduardo Marcel Maçan, Ariana Souza de Santana e a todos que colaboraram direta ou indiretamente na execução deste trabalho

RESUMO

O modelo de recuperação de informação mais amplamente utilizado no contexto de acervos digitais é o *Vector Space Model*. Algoritmos implementados para este modelo que aproveitam informações sobre relevância obtidas dos usuários (chamados *feedbacks*) na tentativa de melhorar os resultados da busca. Porém, estes algoritmos de *feedback* de relevância não possuem uma estratégia global e permanente, as informações obtidas desses *feedbacks* são descartadas para cada nova sessão de usuário (são perenes) ou não modificam os documentos como um todo (são alterações locais). Este trabalho apresenta um método de *feedbacks* de relevância denominado orientado a termos, permitindo que as modificações realizadas por influência dos *feedbacks* dos usuários sejam globais e permanentes. Foram realizados experimentos utilizando o *dataset* ClueWeb09 que dão evidências de que este método melhora a qualidade dos resultados da busca em relação ao modelo tradicional *Vector Space Model*.

Palavras-chave: recuperação de informação, motores de busca, biblioteca digital.

ABSTRACT

The Vector Space Model is the most widely used information retrieval model within digital libraries' systems. Algorithms developed to be used with this model use relevance information obtained from users (called feedbacks) to improve the search results. However, the relevance feedback algorithms developed are not global nor permanent, the feedbacks are discarded in users new sessions and do not affect every document. This paper presents a method that uses of relevance feedback named terms oriented. In this method, users' feedbacks lead to modifications in the terms' vectors representations. These modifications are global and permanent, influencing further searches. An experiment was conducted using the ClueWeb09 dataset, giving evidence that this method improves the quality of search results when compared with Vector Space Model.

Keywords: information retrieval, search engines, digital library.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O acesso à informação e ao conhecimento é fundamental no desenvolvimento social e econômico. O uso do conhecimento torna possível o avanço da sociedade como um todo e melhora a qualidade de vida de cada indivíduo. As bibliotecas possuem um importante papel neste processo, como instituições responsáveis por concentrar, organizar e disponibilizar essas informações. Mas, diante das novas tecnologias da informação, o desenvolvimento e adoção dos acervos digitais expande as responsabilidades das bibliotecas (como instituições detentoras dos acervos) e amplia o acesso ao conhecimento através da remoção dos limites físicos do alcance das informações.

As tecnologias da informação e o advento da *World Wide Web* tornaram possível a criação e a disponibilização de grandes repositórios e acervos digitais na internet. De acordo com os autores Schwartz (2000) e Wan e Liu (2008), a quantidade de acervos digitais e o volume de informações ou documentos dentro desses acervos estão aumentando, impulsionados pelo avanço dessas tecnologias responsáveis pelo armazenamento, gerenciamento e oferecimento, e também pelo reconhecimento da necessidade de preservação e disponibilização desses acervos.

Diante do grande volume de informações disponíveis e da necessidade de permitir o acesso a documentos ou itens rapidamente, o motor de busca é considerado um importante mecanismo de recuperação da informação contida nos acervos digitais. A utilização dos motores de busca faz com que o usuário receba uma quantidade muito grande de documentos como resultado de uma consulta ao acervo. Entretanto, muitos dos itens retornados não são de seu interesse ou não estão ordenados na ordem de relevância efetiva, obrigando o usuário a avaliar vários itens para encontrar aqueles que são de seu interesse. Além disso, os usuários se concentram principalmente nos primeiros resultados e dificilmente analisam todos os documentos devolvidos pela consulta (PASS; CHOWDHURY; TORGESON, 2006). Uma forma de minimizar este problema é utilizar algoritmos que melhorem a ordenação dos resultados da busca.

Os mecanismos de recuperação da informação são projetos de software responsáveis, nos acervos digitais, pela recuperação dos objetos digitais que respondam à consulta feita pelo usuário e ordenação desses resultados da forma mais útil para o usuário.

1.1 Contextualização do Problema

O contexto desde projeto de pesquisa são os acervos digitais memoriais cujas obras tenham principalmente características textuais, sejam obras de caráter cultural,

artístico e arquivístico. O termo mais utilizado na literatura para definir o que são os acervos digitais é, em inglês, *digital library* (traduzindo literalmente "biblioteca digital"), porém neste projeto de pesquisa, é utilizado o termo "acervo digital" como uma tradução livre para *digital library* com o objetivo de melhor representar os acervos memoriais (que representam mais que somente bibliotecas). Porque este projeto de pesquisa pretende abranger acervos textuais mais amplos do que bibliotecas digitais (geralmente limitadas a somente coleções de livros) e abranger outros acervos que mantenham características textuais, como arquivos por exemplo, que incluem além de livros, correspondências, diários oficiais e outros documentos. Existem muitas definições para o termo "acervo digital" ou "biblioteca digital". Por exemplo, em Schwartz (2000), o autor identificou 64 diferentes definições para os termos e reuniu as principais características de um acervo digital presentes na maior parte dessas diferentes definições. Essas principais características também estão presentes nas definições dadas por diversos outros autores, como Martins, Nunes e Rodrigues (2008), Sayão (2008), Lynch (2003), Lesk (1995), e são listadas a seguir. Um acervo digital possui:

- Grandes coleções de objetos nascidos digitais ou digitalizados em variados formatos de mídia;
- Instituição ou instituições responsáveis por manter o acervo persistente, bem organizado e bem gerenciado;
- Uma comunidade alvo ou um conjunto de comunidades alvo, para as quais o acesso ao acervo é geralmente gratuito;
- Acesso ao conteúdo do acervo de forma eficaz e eficiente.

Neste trabalho o acervo digital é definido como uma coleção de itens ou objetos digitais (tanto nascidos digitais quanto digitalizados) persistentes, organizados e gerenciados por uma instituição responsável. O acervo digital não envolve somente a coleção em si, envolve também o sistema de gerenciamento, as pessoas e instituições responsáveis, além de uma comunidade alvo desse acervo.

No contexto dos acervos digitais usualmente não existem hiperlinks explícitos entre os objetos digitais e nem existem referências bibliográficas entre os documentos, ou seja, um documento não possui uma relação explícita com outros documentos do acervo. Mesmo quando essas relações existem explicitamente, no contexto desses acervos os hiperlinks e as referências não influenciam a relevância de um documento durante o processo de busca. Os objetos nesses acervos são basicamente textos extensos e possuem algumas de suas informações organizadas por especialistas no acervo em campos de metadados bem definidos.

Por causa dessa particularidade dos acervos digitais, algoritmos de ordenação e motores de busca que considerem os hiperlinks entre documentos como uma variável relevante para determinar a ordenação dos resultados, como o *PageRank*, que é utilizado no contexto de página web (BRIN; PAGE, 1998), não são boas opções como algoritmos de ordenação. O algoritmo *PageRank* leva em consideração que quanto maior for a quantidade de hiperlinks que apontam para uma página web, maior é a relevância desta página. De modo análogo, algoritmos e motores de busca que analisam as referências (geralmente referências bibliográficas, características de artigos e trabalhos científicos) entre os documentos de um acervo e utilizam essas referências na ordenação da busca (DRORI, 2002) também não são bons candidatos, porque neste contexto de acervos digitais as referências entre documentos não influenciam na relevância desses documentos diante de uma consulta. Já no contexto de trabalhos científicos, os algoritmos de ordenação consideram que a quantidade de referências que um artigo recebe é diretamente proporcional à relevância deste artigo em sua área do conhecimento.

1.2 Descrição do Problema

Os motores de busca são ferramentas responsáveis por devolver uma lista de itens ou documentos do acervo que respondem a uma consulta feita pelo usuário. A qualidade dessa lista é diretamente proporcional à relevância dos documentos que fazem parte dela, essa qualidade é medida pela precisão (proporção de documentos relevantes recuperados em relação a todos os documentos recuperados) e cobertura (proporção de documentos relevantes recuperados em relação a todos os documentos relevantes do acervo). A relevância de um documento depende dos interesses e necessidades do usuário no momento em que ele formula a consulta que será inserida no sistema. Então, a qualidade de um motor de busca está ligado à sua capacidade em responder aos usuários.

Os motores de busca utilizados dentro de sistemas de acervos digitais não apresentam uma ótima qualidade, principalmente em relação à ordenação dos resultados da busca. Esses motores utilizam como base o VSM (*Vector Space Model*, descrito na Seção 2.1), um modelo de recuperação de informação simples que possui algumas deficiências (discutidas mais detalhadamente nas seções 3 e 4).

Uma das principais deficiências do VSM é o fato de o modelo não aproveitar o *feedback* dos usuários na melhoria dos resultados da busca. Como solução para contornar este problema, existe um grande conjunto de pesquisas e implementações de diferentes mecanismos que se focam nos *feedbacks* de relevância, tanto na recuperação dos *feedbacks* dos usuários quanto na utilização desses *feedbacks* recuperados de modo a melhorar os resultados da busca (discutidos na Seção 4). Um dos problemas é que a grande maioria das pesquisas realizadas na área se foca no uso de *feedbacks* orientado a consulta, ou seja,

os *feedbacks* de um usuário são utilizados para alterar a consulta feita pelo próprio usuário e, portanto, nas próximas consultas os *feedbacks* obtidos são descartados.

Existem métodos de *feedback* de relevância que mantém permanentemente as informações obtidas dos *feedbacks*, chamados métodos orientados a documentos pois alteram a representação dos documentos indexados pelo sistema. Como os *feedbacks* refletem na representação dos documentos e não das consultas, essas alterações terão efeito sobre as próximas consultas feitas no sistema. Mas a abordagem orientada a documentos possui as suas próprias deficiências. Alguns documentos, por serem adicionados no acervo posteriormente ou por não estarem nas primeiras posições da lista de resultados, podem ser relevantes mas nunca são julgados pelos usuários e, por isso, são "negligenciados" pelo algoritmo de *feedback* de relevância orientado a documentos. Esses documentos "negligenciados" recebem valores de similaridade com a consulta cada vez menores e continuam a ser preteridos nas próximas consultas.

Uma alternativa sugerida por este trabalho é a utilização da abordagem de *feedback* de relevância chamada orientada a termos, na qual as informações obtidas dos *feedbacks* são utilizadas para melhorar a compreensão do modelo de recuperação de informação sobre a relação entre os termos do vocabulário. Como esta nova abordagem altera a representação dos termos no modelo, mesmos aqueles documentos não julgados têm seus cálculos de similaridade afetados pelos *feedbacks* dos usuários

1.3 Objetivo

O objetivo deste trabalho de pesquisa é investigar se a abordagem de *feedback* de relevância orientada a termos pode melhorar a qualidade dos resultados obtidos pelo motor de busca diante de uma consulta feita pelo usuário. A qualidade dos resultados é medida utilizando uma métrica que combina as métricas de precisão e cobertura, chamada precisão média por nível de cobertura, descrita na Seção 6.2. Esta métrica é calculada para cada algoritmo que fará parte dos experimentos e as curvas resultantes são comparadas.

Para avaliar experimentalmente esta hipótese foi desenvolvido um algoritmo de *feedback* de relevância orientado a termos. Este algoritmo desenvolvido foi testado e comparado com o modelo VSM padrão por meio da realização de experimentos a serem realizados com o uso do *dataset ClueWeb09* (CALLAN et al., 2009).

Os objetivos específicos são:

- Apresentar as diferentes abordagens de *feedback* de relevância (tanto orientadas a documentos quanto orientadas a consultas);
- Desenvolver uma abordagem de *feedback* de relevância permanente (orientada a termos) e um algoritmo que utilize esta abordagem;

- Realizar experimentos utilizando esta abordagem orientada a termos comparando com o modelo VSM, utilizando a métrica de precisão média por nível de cobertura.

A abordagem proposta de *feedback* de relevância orientada a termos pode ser utilizada na contexto de acervos digitais, porque respeita as particularidades desses acervos, ou seja, não leva em consideração a existência de hiperlinks ou referências entre os documentos do acervo.

1.4 Estrutura

Esta dissertação está estruturada em seções, onde cada seção apresenta conceitos ou etapas distintas deste projeto de pesquisa. Na Seção 2 são apresentados os principais modelos de recuperação de informação, incluindo o VSM que é o modelo mais amplamente utilizado e serve de base de comparação para os experimentos realizados durante a pesquisa. Na Seção 3 o problema de ortogonalidade do VSM é descrito e discutido mais profundamente, e são apresentadas algumas soluções já desenvolvidas. Na Seção 4 são descritas as diferentes estratégias de uso dos *feedback* de relevância e é discutido o estado da arte das pesquisas realizadas nessa área.

Na Seção 5 é apresentado o algoritmo de *feedback* de relevância desenvolvido para esta pesquisa, descrevendo o seu funcionamento. Na Seção 6 são descritos os experimentos realizados que comparam o desempenho do algoritmo desenvolvido e o modelo VSM. Nas Seções 7 e 8 são apresentados os resultados obtidos pelos experimentos e as conclusões a partir dos resultados.

2 MODELOS DE RECUPERAÇÃO DE INFORMAÇÃO

A recuperação de informação (IR, *Information Retrieval*) está relacionada à representação, armazenamento, organização e acesso a informações em itens ou documentos. O principal objetivo da recuperação de informação é permitir ao usuário fácil acesso às informações de seu interesse. Desse modo, o objetivo é recuperar todos os documentos relevantes do acervo diante de uma consulta feita pelo usuário, enquanto recupera a menor quantidade possível de documentos não relevantes (BAEZA-YATES; RIBEIRO-NETO, 2011).

Os motores de busca são sistemas de recuperação de informação. Esses sistemas são divididos, simplificadamente, em três partes: (i) obtenção dos dados, na qual são obtidos os documentos que serão consultados e analisados diante das consultas dos usuários; (ii) indexação, na qual é gerado o índice invertido que será utilizado para facilitar o processo de busca; (iii) busca e ordenação dos resultados, na qual a consulta do usuário é comparada com o conjunto de documentos e os documentos relevantes para esta consulta são devolvidos em ordem decrescente de relevância.

(i) Obtenção dos dados

O processo de obtenção dos itens (ou documentos) que farão parte da busca depende muito do contexto no qual este motor de busca será utilizado. No contexto da busca na web, onde o motor de busca é responsável por recuperar páginas web, a obtenção dos itens é realizada utilizando *web crawlers*.

Web crawlers são agentes de software capazes de identificar, visitar e coletar informações de páginas web. Além de obter as informações necessárias para inserir essas páginas como itens no motor de busca, esses agentes analisam os hiperlinks que se encontram nessa página e navegam através desses hiperlinks na procura de novas páginas ou de maiores informações sobre essas páginas.

No contexto de acervo digitais, existem sistemas responsáveis pela inclusão de documentos e gerenciamento do acervo. Esses sistemas provêm aos administradores do acervo ferramentas que tornam possível a adição de novos itens ou a modificação de qualquer informações sobre os itens já existentes. As informações desses itens são repassadas pelo sistema gerenciador de acervo ao motor de busca que é responsável por organizar nessas informações de maneira a facilitar as buscas feitas pelos usuários do acervo posteriormente. Assim, neste contexto, o sistema gerenciador é o responsável por fornecer as informações

sobre os itens do acervo que serão consultas pelos usuários e o motor de busca não é responsável por ativamente encontrar os novos itens.

(ii) Indexação

A indexação é o processo de análise e pré-processamento dos documentos obtidos. Este processo é realizado pelo mecanismo de recuperação de informação e o principal objetivo é a construção do índice invertido. Durante a fase de indexação, os documentos ou itens são analisados, extraindo os termos que serão usados na representação de cada item dentro do modelo de recuperação de informação utilizado no motor de busca.

A escolha desses termos é definida pelo modelo de recuperação de informação utilizado. O termo de um item deve ocorrer pelo menos uma vez neste item e é uma palavra do item, mas pode ser formado por mais de uma palavra (unigramas, bigramas ou trigramas de palavras). Em alguns acervos específicos, a escolha desses termos é limitada por um dicionário controlado da área de conhecimento específica e todos os termos utilizados devem estar neste dicionário. Além disso, os modelos de recuperação de informação excluem os termos presentes na lista de "palavras vazias" (ou, em inglês *stopwords*) (MANNING; RAGHAVAN; SCHüTZE, 2009; BAEZA-YATES; RIBEIRO-NETO, 2011). A lista de *stopwords* depende da linguagem do acervo e é formada pelos termos dessa linguagem que não ajudam na descrição dos documentos, por exemplo, no português algumas *stopwords* comuns são os artigos ("a", "o", "uma", "um") e as conjunções ("e", "mas", "ou", "portanto").

Para cada termo extraído dos documentos é calculado o peso desse termo neste documento. Este peso mede numericamente o quanto este termo é relevante no documento específico ou o quanto este termo representa este documento. O conjunto de todos os termos de todos os documentos forma o vocabulário do acervo, que é utilizado na geração do índice invertido.

O índice invertido é uma estrutura de dados que relaciona cada termo do vocabulário com o conjunto de documentos que possuem pelo menos uma ocorrência desse termo (MANNING; RAGHAVAN; SCHüTZE, 2009; BAEZA-YATES; RIBEIRO-NETO, 2011). Então, este índice é utilizado para localizar rapidamente todos os documentos que possuem ocorrências dos termos buscados pelo usuário, agilizando o processo de busca.

(iii) Busca e ordenação dos resultados

O processo de busca se inicia normalmente na consulta do índice invertido, no qual são recuperados todos os documentos que possuem os termos buscados pelo usuário. Após isso, é calculada a similaridade de cada um desses documentos com a consulta realizada pelo usuário. Esta similaridade da consulta com documento é entendida como a relevância

do documento para a consulta realizada. Desse modo, os resultados da busca que serão devolvidos ao usuário podem ser ordenados em ordem decrescente de relevância.

A identificação da relevância de um documento diante de uma consulta é a parte mais crítica desses sistemas. Focados neste aspectos são criados os modelos de recuperação de informação. Um modelo de IR pode ser formalmente descrito como uma quadrupla: [D, Q, F, $r(d_i, q_j)$] (BAEZA-YATES; RIBEIRO-NETO, 2011), na qual:

- D é o conjunto de documentos ($\{d_1, d_2, d_3 ... d_l\}$);
- Q é o conjunto de consultas ($\{q_1, q_2, q_3 ... q_j\}$);
- F é o arcabouço (ou *framework*) lógico que permita representar os documentos e as consultas;
- $r(d_i, q_j)$ é uma função de relevância $r(d_i, q_j) \rightarrow \mathbb{R}$. Onde o valor retornado por esta função representa a relevância do documento d_i diante de uma consulta q_j e permite que os documentos seja ordenados em ordem decrescente de relevância.

Entre os quatro elementos da quadrupla, o conjunto de documentos (D) e o conjunto de consultas (Q) não dependem estritamente do modelo de IR escolhido. Portanto, os modelos de recuperação de informação podem ser caracterizados e diferenciados somente pelo arcabouço lógico de representação dos documentos e das consultas e pela função de ordenação dos resultados [F, $r(d_i, q_j)$].

As principais métricas utilizadas para a avaliação do desempenho de diferentes algoritmos ou modelos de recuperação de informação são a precisão e a cobertura. Onde precisão é a proporção de documentos retornados que são relevantes (ou seja, a quantidade de documentos relevantes retornados sobre a quantidade de documentos retornados) e cobertura é a proporção de documentos relevantes que são retornados (ou seja, a quantidade de documentos relevantes retornados sobre a quantidade de documentos relevantes) (JäRVELIN; KEKäLäINEN, 2000; SARACEVIC et al., 1988; BLAIR; MARON, 1985).

Existem três modelos de recuperação de informação clássicos: modelo de espaço vetorial (VSM, do inglês, *Vector Space Model*), modelo booleano e modelo probabilístico. No contexto de acervo digitais, o VSM é o modelo de recuperação de informação mais amplamente utilizado nos motores de busca no contexto de acervos digitais (MAO; CHU, 2002; GUO, 2008; TURNEY; PANTEL, 2010).

2.1 VSM - *Vector Space Model*

O modelo chamado VSM define um espaço vetorial para representar os documentos e as consultas dentro do modelo. Os documentos e consultas são representados por vetores multidimensionais, normalmente utilizando valores reais. Já a função de ordenação é caracterizada como uma das possíveis operações sobre vetores (produto vetorial ou cosseno do ângulo), utilizando esta simplificação de independência entre os termos.

O VSM é frequentemente atribuído a Salton, Wong e Yang (1975). Mas no seu artigo, Salton, Wong e Yang (1975) utilizam o espaço vetorial somente como ilustração da estrutura de dados por trás da representação de documentos no sistema e do processamento realizado durante a indexação automática. Este modelo apresentado por Salton, Wong e Yang (1975) será tratado como "VSM simplificado" e é melhor descrito na Seção 2.1.1. Como o espaço vetorial não é formalmente descrito, outros autores (WONG; ZIARKO; WONG, 1985; WONG; RAGHAVAN, 1984; KOLL, 1979) identificaram simplificações, principalmente relacionadas à ortogonalidade da base vetorial do espaço vetorial utilizado que influencia diretamente no cálculo de similaridade. A Seção 3 descreve mais profundamente essas simplificações e as implicações destas sobre o modelo VSM simplificado.

Somente em Salton (1989), o VSM é apropriadamente apresentado como um modelo de recuperação de informação e seu espaço vetorial é formalmente descrito, levando em consideração as contribuições de Wong, Ziarko e Wong (1985), Wong e Raghavan (1984), Koll (1979). Este será tratado como o "VSM completo".

2.1.1 Descrição do modelo VSM

Considerando um vocabulário $V = \{t_1, t_2, t_3, ...t_n\}$ com n termos, cada documento d_i indexado pelo VSM é representado por um vetor de n dimensões.

$$d_i = \{d_{i1}, d_{i2}, d_{i3}, ...d_{in}\} \tag{2.1}$$

Cada elemento d_{ik} do vetor d_i representa o valor do peso do termo t_k dentro do documento d_i. O peso de um termo dentro de um documento é um valor real (como a quantidade de ocorrências do termo no documento) ou um valor binário (onde 0 representa a ausência do termo no documento e 1 representa a presença). Da mesma maneira que os documentos, uma consulta q_j também é representada por um vetor multidimensional neste mesmo espaço vetorial.

$$q_j = \{q_{j1}, q_{j2}, q_{j3}, ...q_{jn}\} \tag{2.2}$$

Esta representação dos documentos e consultas como vetores também é interpretada como uma combinação linear de vetores, onde os documentos são os termos da combinação linear e os pesos dos termos são as constantes que multiplicam cada um desses documentos.

$$d_i = \sum_{k=1}^{n} d_{ik}.v_k \tag{2.3}$$

$$q_j = \sum_{k=1}^{n} q_{jk}.v_k \tag{2.4}$$

tal que,

$$v_1 = (1, 0, 0, ..., 0)$$
$$v_2 = (0, 1, 0, ..., 0)$$
$$\vdots$$
$$v_n = (0, 0, 0, ..., 1)$$

Desse modo, o conjunto dos vetores v_k forma a base desse espaço vetorial. Cada um desses vetores v_k é interpretado como a representação de cada um dos termos do vocabulário do modelo.

A similaridade entre um documento e uma consulta ($s(d_i, q_j)$) é calculada como o produto escalar entre os dois vetores, onde θ é o ângulo entre os dois vetores (SALTON; WONG; YANG, 1975; SALTON, 1989) (Equação 2.5). Conforme Baeza-Yates e Ribeiro-Neto (2011), Manning, Raghavan e Schütze (2009), uma forma alternativa de calcular a similaridade entre documentos e consultas é a utilização do valor do cosseno do ângulo formado entre os dois vetores (Equação 2.6).

$$s(d_i, q_j) = d_i \cdot q_j = |d_i||q_j| \cos(\theta) \tag{2.5}$$

$$s(d_i, q_j) \cos(\theta) = \frac{d_i \cdot q_j}{|d_i||q_j|} \tag{2.6}$$

O VSM simplificado assume que os vetores que representam os termos são ortogonais entre si. Esta ortogonalidade implica que $v_i \cdot v_j = 0$ (se $i \neq j$) e $v_i \cdot v_i = 1$ (se $i = j$). Nesta situação, os documentos e consultas são descritos como combinações lineares de vetores ortogonais entre si e o produto escalar entre dois vetores é calculado de modo mais simples.

$$d_i \cdot q_j = \sum_{k=1}^{n} d_{ik} q_{jk} \tag{2.7}$$

Ou

$$\cos(\theta) = \frac{\sum_{k=1}^{n} d_{ik} q_{jk}}{\sqrt{\sum_{k=1}^{n} d_{ik}^2} \sqrt{\sum_{k=1}^{n} q_{jk}^2}} \tag{2.8}$$

Esta simplificação afeta diretamente o modo como os termos do vocabulário são modelados dentro do VSM e implica que esses termos são independentes entre si, ou seja, não existe nenhuma relação semântica entre termos ou, ao menos, esta relação não é apropriadamente representada no modelo. Somente em 1989, Salton (1989) apresenta formalmente o VSM como um modelo de recuperação de informação e endereça esta questão relacionada à ortogonalidade dos termos.

Este modelo apresentado em 1989 é considerado o VSM completo e sugere que o cálculo da similaridade (ilustrado na equação 2.7) é realizado sem a simplificação aceita anteriormente, levando em consideração os vetores de termos (como na equação 2.9).

$$d_i \cdot q_j = \sum_{k,m=1}^{n} d_{ik} q_{jm} (v_k \cdot v_m) \tag{2.9}$$

Por mais que Salton (1989) tenha formalizado o VSM levando em consideração a relação entre termos e que alguns autores afirmem obter bons resultados na recuperação e ordenação dos documentos de uma busca levando em consideração a relação entre termos (KOLL, 1979; WONG; RAGHAVAN, 1984; WONG; ZIARKO; WONG, 1985; TSATSARONIS; PANAGIOTOPOULOU, 2009), a grande maioria dos trabalhos realizados na área de recuperação de informação descreve o VSM como sua versão simplificada (GUPTA; SAINI; SAXENA, 2014; BAEZA-YATES; RIBEIRO-NETO, 2011; MANNING; RAGHAVAN; SCHüTZE, 2009; SINGHAL, 2001) e a grande maioria das implementações do modelo em motores de busca utiliza o VSM simplificado (HATCHER; GOSPODNETIC, 2004; Index Data, 2014). Então, o VSM ainda se refere principalmente ao modelo simplificado de Salton, Wong e Yang (1975) e assume a ortogonalidade entre os vetores que representam os termos do vocabulário.

Na Seção 3 é realizada uma discussão mais aprofundada sobre a ortogonalidade entre os termos, incluindo um exemplo de cálculo de similaridade com relações entre termos e a descrição de modelos capazes de representar a relação entre os termos do vocabulário e utilizá-la no cálculo de similaridade entre documentos e consultas.

Além da limitação relacionada à ortogonalidade entre os termos, o VSM possui uma limitação relacionada à omissão dos interesses dos usuários. Esta limitação está relacionada à falta de mecanismos capazes de receber e entender os interesses e as necessidades dos usuários e aplicá-los para a melhoria do sistema. Assim, uma das abordagens utilizadas para contornar esta limitação é a implementação de mecanismos de "*feedback* de relevância".

Sistemas que implementam mecanismos de *feedback* de relevância são capazes de receber *feedbacks* dos usuários sobre a relevância dos itens apresentados diante de uma consulta. A partir desses *feedbacks*, as representações dos documentos ou das consultas podem ser alteradas para melhor refletirem os interesses desses usuários. O uso de *feedbacks* de relevância é abordado posteriormente na Seção 4.

2.2 Modelo Booleano

O modelo booleano é um modelo mais simples que o VSM e baseado na teoria de conjuntos e álgebra booleana (LASHKARI; MAHDAVI; GHOMI, 2009; WARTIK, 1992). Os documentos não são representados como vetores, mas sim como subconjuntos do conjunto de termos (vocabulário). Já as consultas são representadas como expressões booleanas, nas quais os elementos são termos do vocabulário e as operações sobre os elementos podem ser "AND", "OR" ou "NOT".

Desse modo, o processo de recuperação dos documentos relevantes para a consulta é dividido em dois passos:

- No primeiro passo, cada elemento da expressão booleana é consultado no índice invertido, formando um conjunto de todos os documentos que possuem o termo deste elemento. Este passo é repetido para cada elemento da consulta, com as devidas alterações caso o elemento possuir um operador "NOT";

- No segundo passo, as operações booleanas ("AND", "OR" e "NOT") são transformadas em operações sobre conjuntos (intersecção, união e diferença, respectivamente) e aplicadas sobre os conjuntos obtidos no passo anterior. Assim, o resultado deste segundo passo é a resposta devolvida pelo motor de busca.

Uma das principais deficiências desse modelo é a falta de um algoritmo de ordenação dos resultados por ordem de relevância. Em acervos com uma quantidade muito grande de documentos, uma consulta frequentemente devolverá uma quantidade muito grande de resultados e eles não estarão ordenados por relevância. Isto obrigará o usuário a analisar os documentos um a um frustrando a experiência do usuário no acervo e tornando o mecanismo de busca quase inútil.

Boa parte dos mecanismos de busca utilizam este modelo em conjunto com algum outro, por causa da sua capacidade de recuperar quais documentos possuem os termos da consulta e de permitir aos usuários a utilização de operações booleanas nas consultas. Depois de recuperados todos os documentos que possuem os termos utilizando o modelo booleano, são aplicados algoritmos de um outro modelo (por exemplo, o VSM) para ordenar

os resultados por relevância. Assim, o modelo booleano fica encarregado de diminuir o tamanho do conjunto de documentos que serão analisados pelo segundo modelo.

2.3 Modelo Probabilístico

O modelo probabilístico proposto pelos autores Robertson e Jones (1976) leva em consideração a existência de um conjunto R de documentos relevantes para uma determinada consulta q_j. Assim, a relevância de um documento (ou a similaridade entre o mesmo e a consulta) é calculada utilizando como base a probabilidade deste documento ser ou não parte do conjunto R de documentos relevantes. Este modelo, como o VSM, também representa documentos e consultas como vetores multidimensionais com tamanho igual à quantidade n de termos. Mas no caso desse modelo, os valores aceitos pelo vetor são somente 0 ou 1, representando a ausência ou presença do termo no documento.

Inicialmente, para a primeira interação com a busca, é feita uma previsão simples da probabilidade e um conjunto de documentos é devolvido pelo mecanismo de busca. Então, o usuário pode indicar aqueles documentos que ele considerou relevantes para sua consulta. A partir desta realimentação do usuário, o algoritmo utiliza as informações obtidas para melhorar a sua previsão e procura por um conjunto mais relevante de documentos.

A similaridade entre documento e consulta, neste modelo, é calculada como a proporção entre a probabilidade de um documento d_i ser relevante para a busca ($P(R|d_i)$) e a probabilidade deste mesmo documento não ser relevante para a busca ($P(\overline{R}|d_i)$) (BAEZA-YATES; RIBEIRO-NETO, 2011).

$$s(d_i, q_j) = \frac{P(R|d_i)}{P(\overline{R}|d_i)} \tag{2.10}$$

Esta probabilidade é calculada alterando esta função através do teorema de Bayes.

$$s(d_i, q_j) = \frac{P(d_i|R)P(R)}{P(d_i|\overline{R})P(\overline{R})}$$

Onde $P(d_i|R)$ é a probabilidade de selecionar um documento com a mesma representação vetorial de d_i dentro do conjunto de R de documentos relevantes. $P(d_i|\overline{R})$ é análogo ao anterior, mas dentro do conjunto $\overline{R}$ de documentos não relevantes. Como $P(R)$ e $P(\overline{R})$ são o mesmo para todos os documentos, não são relevantes para a ordenação, portanto, também são desconsiderados.

$$s(d_i, q_j) = \frac{P(d_i|R)}{P(d_i|\overline{R})}$$

Utilizando uma simplificação parecida com aquela admitida no modelo VSM, podemos considerar os termos ortogonais e independentes entre si. Desse modo, cada d_i é representado por uma combinação de vetores ortogonais entre si.

$$s(d_i, q_j) = \frac{(\prod_{d_{ik}=1} P(t_k|R)).(\prod_{d_{ik}=0} P(\overline{t_k}|R))}{(\prod_{d_{ik}=1} P(t_k|\overline{R})).(\prod_{d_{ik}=0} P(\overline{t_k}|\overline{R}))}$$

Onde $(\prod_{d_{ik}=1} P(t_k|R))$ é o produto de todas as probabilidades do termo t_k estar presente em um documento aleatório do conjunto R, para cada termo $d_{ik} = 1$ presente no documento d_i. E a equação $(\prod_{d_{ik}=0} P(\overline{t_k}|R))$ tem um significado análogo, para cada termo $d_{ik} = 1$ não presente no documento d_i, esta equação é o produto de todas as probabilidades do termo t_k não estar presente em um documento aleatório do conjunto R. Os outros termos da equação de similaridade têm significados análogos, mas para o conjunto $\overline{R}$.

Considerando $p_k = P(t_k|R)$, então, $1 - p_k = P(\overline{t_k}|R)$. Da mesma maneira, $pr_k = P(d_{ik}|\overline{R})$ e $1 - pr_k = P(\overline{t_k}|\overline{R})$.

$$s(d_i, q_j) = \frac{(\prod_{d_{ik}=1} p_k).(\prod_{t_k=0}(1 - p_k))}{(\prod_{d_{ik}=1} pr_k).(\prod_{t_k=0}(1 - pr_k))}$$

Calculando o logaritmo de todos os termos da equação altera somente os valores absolutos calculados, mas não altera a ordem de similaridade entre os documentos.

$$\begin{aligned} s(d_i, q_j) = &\log(\prod_{d_{ik}=1} p_k) + \log(\prod_{d_{ik}=0} (1 - p_k)) \\ &- \log(\prod_{d_{ik}=1} pr_k) - \log(\prod_{d_{ik}=0} (1 - pr_k)) \end{aligned}$$

Adicionando termos que não alteram o valor da similaridade.

$$\begin{aligned} s(d_i, q_j) = &\log(\prod_{d_{ik}=1} p_k) + \log(\prod_{d_{ik}=0} (1 - p_k)) \\ &- \log(\prod_{d_{ik}=1} (1 - p_k)) + \log(\prod_{d_{ik}=1} (1 - p_k)) && \text{(estes termos se cancelam)} \\ &- \log(\prod_{d_{ik}=1} pr_k) - \log(\prod_{d_{ik}=0} (1 - pr_k)) \\ &+ \log(\prod_{d_{ik}=1} (1 - pr_k)) - \log(\prod_{d_{ik}=1} (1 - pr_k)) && \text{(estes termos também se cancelam)} \end{aligned}$$

Agrupando logaritmos.

$$s(d_i, q_j) = \log\left(\prod_{d_{ik}=1} \frac{p_k}{(1-p_k)}\right) + \log(\prod_{d_{ik}}(1-p_k)) + \log\left(\prod_{d_{ik}=1} \frac{(1-pr_k)}{pr_k}\right) - \log(\prod_{d_{ik}}(1-pr_k))$$

Os termos da equação $\log(\prod_{d_{ik}}(1-p_k))$ e $\log(\prod_{d_{ik}}(1-pr_k))$ são produtos calculados sobre todos os termos do índice, portanto, seu valor não depende do documento e nem da consulta. Ou seja, estes termos são constantes e não alteram a ordem de similaridade entre os documentos.

$$s(d_i, q_j) = \log\left(\prod_{d_{ik}=1} \frac{p_k}{(1-p_k)}\right) + \log\left(\prod_{d_{ik}=1} \frac{(1-pr_k)}{pr_k}\right)$$

Agora, ambos os termos da equação levam em consideração somente os termos do índice que ocorrem no documento ($d_{ik} = 1$). Para o cálculo da relevância de um documento ou a similaridade com uma consulta, podemos assumir que para todo termo que não ocorre na consulta a probabilidade do termo estar presente em um documento do conjunto R é a igual à probabilidade deste mesmo termo estar presente em um documento aleatório do conjunto $\overline{R}$. Ou seja, para os termos $d_{ik} = 1$ que não estão presentes na consulta ($q_{jk} = 0$), $p_k = pr_k$. Assim, podemos calcular a similaridade entre documento e consulta do seguinte modo.

$$s(d_i, q_j) = \sum_{d_{ik}=1 \wedge q_{jk}=1} \log\left(\frac{p_k}{(1-p_k)}\right) + \log\left(\frac{(1-pr_k)}{pr_k}\right) \tag{2.11}$$

A fórmula 2.11 é a principal função para o cálculo da similaridade entre documentos e consultas, permitindo a ordenação dos resultados de uma busca. Mas para realizar este cálculo é necessário algum conhecimento sobre o o conjunto R. Como nas primeiras interações ainda não existe nenhuma informação sobre este conjunto, o modelo utiliza estimativas de probabilidade (BAEZA-YATES; RIBEIRO-NETO, 2011).

$$p_k = 0.5 \text{ e } pr_k = \frac{n_k}{N}$$

Onde n_k é o número total de documentos que possuem o termo $d_{ik} = 1$ e N é a quantidade total de documentos. Assim, utilizando estes valores sobre a função 2.11, temos o seguinte cálculo de similaridade.

$$s(d_i, q_j) = \sum_{d_{ik}=1 \wedge q_{jk}=1} \log\left(\frac{(N - n_k)}{n_k}\right) \tag{2.12}$$

Esta função 2.12 só é utilizada nas primeiras interações, pois nas próximas, com a realimentação do usuário em relação à relevância dos documentos é possível melhorar as estimativas de relevância dos documentos. Então, os documentos considerados relevantes para os usuários são inseridos no conjunto R e as informações sobre eles são utilizadas durante o cálculo da similaridade entre documentos e consultas 2.11.

Uma das deficiências desse modelo é utilização de pesos binários para os termos de um documento. Desse modo, o peso de dois termos presentes em um documento é sempre igual, levando a uma limitação na representação desses documentos. Esta limitação provoca duas situações indesejadas:

- Um termo que ocorre por todo um documento ou que está presente no título e é claramente muito relevante para este documento terá o mesmo peso no cálculo de similaridade que um outro termo menos relevante que ocorre uma única vez, pois o peso é binário e somente representa a presença ou ausência do termo no documento;
- Um termo terá sempre o mesmo peso em todos os documentos, tanto diante daqueles em que este termo é relevante quanto diante daqueles onde o termo é irrelevante.

Resumidamente, o modelo probabilístico também representa os documentos e as consultas como vetores (mas, diferentemente do modelo VSM, os vetores possuem valores binários). E a função de ordenação é baseada na teoria da probabilidade e a relevância de um documento diante de uma consulta é representada por uma função que relaciona as probabilidades de este documento pertencer ou não ao conjunto R de documentos relevantes.

3 ORTOGONALIDADE DOS VETORES DE TERMOS

Os modelos de recuperação de informação VSM simplificado e probabilístico utilizam um espaço vetorial para representar os documentos e consultas, apesar de o modelo VSM utilizar valores reais e o modelo probabilístico utilizar valores binários. Além disso, ambos representam os documentos como uma combinação linear utilizando como base um espaço de vetores ortogonais. Estes vetores ortogonais representam cada um dos termos do índice e esta simplificação facilita os cálculos de similaridade e da função de ordenação.

A ortogonalidade dos vetores que representam os termos implica que os termos são independentes entre si. Esta independência entre os termos não é uma suposição válida para a grande maioria dos textos e acervos digitais. Além disso, esta independência pode ser considerada uma deficiência desses modelos, implicando em uma caracterização do espaço de vetores incompleta (WONG; ZIARKO; WONG, 1985; WONG; RAGHAVAN, 1984). Alguns autores, como Doyle (1961), Koll (1979), sugerem que perceber a relação entre os termos é similar a entender a relação entre a semântica dos termos ou a relação entre os conceitos por trás desses termos. Portanto, a inclusão da relação entre os termos nos modelos de recuperação da informação aproximam esses modelos da raciocínio humano.

O autor Koll (1979) criou um exemplo muito simples comparando três sistemas de recuperação da informação diferentes, incluindo o modelo booleano, o VSM e um ser humano. No acervo deste exemplo existem somente dois livros e são indexados apenas pelos termos presentes no título: `(1)` “revisão sobre IA” e `(2)` “história do xadrez”. Nesta situação o índice possuirá cinco termos relevantes para o exemplo “computação”, “história”, “IA”, “revisão”, “xadrez”, sendo que o primeiro termo foi adicionado para facilitar a visualização do exemplo e por causa da presença dele na consulta que será feita pelo usuário. O usuário realiza uma busca pelos termos “computação de xadrez”. Considerando o modelo VSM, estas são as representações dos documentos e da consulta.

$$d_1 = \{0, 0, 1, 1, 0\}$$
$$d_2 = \{0, 1, 0, 0, 1\}$$
$$q_0 = \{1, 0, 0, 0, 1\}$$

Considerando a função de similaridade 2.7.

$$s(d_1, q_0) = 0$$
$$s(d_2, q_0) = 1$$

Ou seja, neste exemplo, a busca por "computação xadrez" utilizando tanto o modelo booleano quanto o VSM retornaria somente o livro `(2)`, por causa da ocorrência do termo "xadrez", e o livro `(1)` seria removido da resposta por não possuir nenhuma ocorrência dos termos da consulta. Mas, por outro lado, um ser humano, que conheça o acervo e entenda os conceitos implícitos nos títulos dos livros e na consulta, seria capaz de fazer uma melhor escolha e indicar o livro `(1)` como resposta a esta consulta. Este exemplo não prova as deficiências do VSM, mas ilustra como o conhecimento semântico sobre os termos, os documentos do acervo e as consultas podem melhorar os resultados do motor de busca comparado com o conhecimento simples da ocorrência de termos dentro dos documentos.

Diversos autores, como Wong, Ziarko e Wong (1985), Wong e Raghavan (1984), Koll (1979) incluindo o próprio Salton (1989), identificaram esta deficiência nos modelos de recuperação da informação, principalmente em relação ao VSM simplificado. E esses autores demonstraram que novos modelos que levam em consideração as relações entre termos melhoram a qualidade dos resultados devolvidos pelo motor de busca.

Duas abordagens são amplamente discutidas na literatura para contornar esta deficiência do modeo VSM são: o uso de vetores de termos não ortogonais e o GVSM (*Generalized Vector Space Model*).

3.1 Vetores de termos não ortogonais

Depois que os autores Wong, Ziarko e Wong (1985) identificaram e criticaram esta deficiência no modelo VSM, o próprio autor Salton (1989) sugere uma maneira de superar essas deficiência de uma maneira simples transformando os termos em vetores de n dimensões e cujos valores sejam a relação do termo representado por cada vetor com os outros termos.

Assim, cada vetor de termo v_i, que representa o vetor t_i, possui a informação da relação deste termo com cada um dos outros termos do vocabulário. Ou seja, o valor de v_{ij} é uma quantificação direta da relação entre os termos t_i e t_j.

$$v_i = \{v_{i1}, v_{i2}, v_{i3}, ...v_{in}\} \tag{3.1}$$

Alinhando todos os vetores de termos, é possível construir a chamada matriz de relação termo-termo (BAEZA-YATES; RIBEIRO-NETO, 2011). Utilizando esta matriz de

relação, a função de ordenação baseada em produto escalar (Equação 2.5) não pode ser simplificada e transformada na Equação 2.7. Então, a Equação 2.5 leva em consideração as relações entre os termos apresentadas na matriz e é calculada através da Equação 2.9.

Onde v_j e v_k representam os vetores dos termos t_j e t_k. Portanto, $v_j \cdot v_k$ representa a relação entre os termos t_j e t_k.

De acordo com Salton (1989), os valores da relação entre os termos são calculados a partir do próprio acervo, partindo da própria indexação já realizada. Nesta situação, o valor da relação entre dois termos representa em quantos documentos do acervo existem a coocorrência desses dois termos e os pesos desses termos nestes documentos. Existem outras abordagens para o cálculo da relação entre os termos diferentes do cálculo sobre o próprio acervo. Algumas dessas abordagens calculam as relações semânticas entre os termos, utilizando fontes externas ao acervo como dicionários de sinônimos (WIBOWO; HANDOJO; HALIM, 2011; TSATSARONIS; PANAGIOTOPOULOU, 2009) ou ontologias (POLYVYANYY; KUROPKA, 2009).

A vantagem da abordagem do cálculo sobre o próprio acervo é que as relações entre os termos refletem muito bem a distribuição desses termos entre os documentos. E mesmo que essas relações não reflitam relações semânticas entre os termos, elas modelam mais fielmente o acervo e podem auxiliar na separação dos documentos relevantes e não-relevantes. Por outro lado, o cálculo das relações entre os termos sobre o próprio acervo é feito sobre o índice gerado pela indexação dos documentos e leva em consideração que esses documentos são ortogonais entre si (SALTON, 1989). Mas este pressuposto não é real em praticamente nenhum acervo digital (WONG; RAGHAVAN, 1984; SALTON, 1989).

3.2 *Generalized Vector Space Model*

Então, como uma alternativa ao modelo VSM proposto por Salton (1979), o modelo GVSM foi proposto por Wong, Ziarko e Wong (1985). Sendo que a principal melhoria desse modelo em relação ao modelo anterior é a introdução de um algoritmo bem definido para o cálculo das relações entre termos.

Este modelo também representa um documento d_i e as consultas q_j como vetores de n dimensões. Além disso, o GVSM também considera um vocabulário $V = \{t_1, t_2, t_3, ...t_n\}$ com n termos. A partir do vocabulário, é possível gerar o conjunto de todas as conjunções geradas entre todos os elementos do vocabulário, de modo que, em cada conjunção, cada termo aparecerá exatamente 1 vez, seja positivamente ou negativamente. Por exemplo, para um vocabulário de 3 termos ($V = \{t_1, t_2, t_3\}$) serão geradas as seguintes conjunções: $t_1\overline{t_2}\overline{t_3}$, $t_1t_2\overline{t_3}$, $t_1\overline{t_2}t_3$, $t_1t_2t_3$, $\overline{t_1}\overline{t_2}\overline{t_3}$, $\overline{t_1}t_2\overline{t_3}$, $\overline{t_1}t_2t_3$, $\overline{t_1}\overline{t_2}t_3$, onde t_i indica a presença do termo t_1 e $\overline{t_1}$, a ausência.

Cada conjunção é chamada de um mintermo m_r, como ilustrado abaixo:

$$
\begin{aligned}
& (t_1, t_2, t_3, ...t_n) \\
m_1 =& (0, 0, 0, ...0) \\
m_2 =& (1, 0, 0, ...0) \\
m_3 =& (0, 1, 0, ...0) \\
m_4 =& (1, 1, 0, ...0) \\
& \vdots \\
m_{2^m} =& (1, 1, 1, ...1)
\end{aligned}
$$

Onde m_4, por exemplo, representa a conjunção em que os termos t_1 e t_2 estão presentes e todos os outros termos, ausentes. Para todo documento d_i existe uma conjunção m_r que o representa exatamente, ou seja, para todo documento existe uma conjunção que respeita as seguintes regras para todo k:

$$
\begin{aligned}
&\text{se } d_{ik} > 0 \text{ então } m_{rk} = 1 \\
&\text{se } d_{ik} = 0 \text{ então } m_{rk} = 0
\end{aligned}
$$

Desse modo, é definida a função $c(d_i) = m_r$, que permite definir o mintermo correspondente a um documento d_i. Outra função definida neste modelo é a função $on(k, m_r)$.

$$
on(k, m_r) = \begin{cases} 1, & \text{se o termo } t_k \text{ está presente em } m_r \\ 0, & \text{se o termo } t_k \text{ está ausente} \end{cases}
$$

Esta função permite definir se um termo t_k está presente no mintermo m_r.

A partir dos mintermos, é possível definir um conjunto de vetores de mintermos M_r, onde cada vetor M_r representa um mintermo m_r. Esses vetores M_r possuem 2^n dimensões e os valores para cada uma das dimensões desses vetores são definidos do seguinte modo.

$$
\begin{aligned}
& 1, 2, 3, ...2^n \\
M_1 &= (1, 0, 0, ...0) \\
M_2 &= (0, 1, 0, ...0) \\
M_3 &= (0, 0, 1, ...0) \\
&\vdots \\
M_{2^n} &= (0, 0, 0, ...1)
\end{aligned}
$$

Ou seja, a única dimensão cujo valor é maior que zero representa qual exatamente é o mintermo a que este vetor se refere. Os vetores de mintermos são todos ortogonais entre si e formam o espaço vetorial sobre o qual é construído o GVSM. O fato de os vetores de mintermos serem ortogonais não implica na ortogonalidade (ou independência) entre os termos do vocabulário, pelo contrário, alguns desses vetores de mintermos representam a relação entre diversos termos. Por exemplo, o vetor M_4 representa o mintermo m_4, que representa a relação entre os termos t_1 e t_2.

Então, a partir desses conjuntos de vetores e de mintermos, o modelo GVSM define um vetor v_i para o termo t_i, que leva em consideração a relação entre os termos do índice. O vetor do termo é descrito do seguinte modo.

$$
v_i = \frac{\sum_{r=1}^{2^n} on(i, m_r) c_{ir} M_r}{\sqrt{\sum_{r=1}^{2^n} on(i, m_r) c_{ir}^2}} \tag{3.2}
$$

$$
\text{onde } c_{ir} = \sum_{d_k | c(d_i) = m_r} d_{ik}
$$

Assim, os documentos e as consultas podem ser representados como uma combinação linear sobre estes novos vetores de termos, que não são mais ortogonais. Então, estes vetores são utilizados na representação dos documentos na equação 2.3. Ou, estes vetores de termo podem ser utilizados na equação 2.9.

Uma deficiência deste modelo é a quantidade de vetores de mintermos e o tamanho em número de dimensões desses vetores. No pior caso, existem 2^n diferentes mintermos e os vetores de mintermos possuem 2^n dimensões. Os autores afirmam que durante a execução desses algoritmos, são considerados somente a quantidade de mintermos ativos, ou seja, somente aqueles mintermos que representam a ocorrência de termos em algum dos documentos do acervo, portanto, a quantidade máxima de mintermos é a quantidade de documentos no acervo (no pior caso, onde cada documento é representado por um

mintermo diferente). Ainda assim, este é um valor bastante grande para os atuais acervos digitais e os experimentos realizados pelos autores (WONG; ZIARKO; WONG, 1985) lidavam com acervos pequenos (um deles com 82 documentos e outro com 424), não testando o modelo em situações mais extremas.

Essas alternativas aos modelos clássicos que consideram a relação entre os termos durante a ordenação dos resultados (tanto a matriz de relação dos termos e GVSM) apresentam algumas deficiências na construção dessa relação entre termos. Ambos modelos calculam ou inferem as relações entre os termos do vocabulário utilizando os próprios documentos do acervo, ou seja, ambos consideram que as relações entre termos é proporcional à coocorrência desses termos nos documentos e assumem que esses documentos em questão são representativos o suficiente para permitir o cálculo da relação entre termos a partir da simples coocorrência destes. Então, por exemplo, a relação calculada entre dois termos é diretamente proporcional à quantidade de documentos que possuam ambos termos e nunca pondera as preferências dos usuários do acervo nesse cálculo.

4 *FEEDBACK* DE RELEVÂNCIA

Além da deficiência relacionada como a ortogonalidade dos vetores que representam os termos, o VSM possui uma segunda deficiência está relacionada ao fato de o modelo não levar em consideração as interações dos usuários sobre o acervo para melhorar as repostas dadas pelo mecanismo de busca. Depois que um sistema de acervos digitais é disponibilizado permitindo acessos do público, os usuários são uma ótima fonte de informação sobre quais documentos são relevantes ou não diantes das consultas feitas por eles, mas o modelo VSM (SALTON; WONG; YANG, 1975) não se preocupa em aproveitar essas informações. A indexação do acervo é feita somente uma vez e seus resultados são fixos, somente alterados diante da inserção de novos itens, e os únicos critérios para a busca são o índice criado e consulta realizada. Endereçando esta deficiência, foram desenvolvidos os algoritmos de *feedback* de relevância, baseados principalmente no algoritmo de Rocchio (1971).

O princípio básico do funcionamento dos mecanismos de *feedback* de relevância é baseado no fato de que um usuário pode enfrentar dificuldades ao formular uma consulta ótima sem um conhecimento mais detalhado do acervo digital, mas ele consegue facilmente avaliar ou julgar, de acordo com os seus interesses, a relevância de um documento específico do acervo (MANNING; RAGHAVAN; SCHüTZE, 2009). Assim, os mecanismos que utilizam o *feedback* de relevância são capazes de recuperar esses *feedbacks* feitos pelos os usuários para tentar melhorar o funcionamento do motor de busca, retornando mais resultados relevantes e melhorando a ordenação dos resultados.

Os diferentes algoritmos de *feedback* de relevância são caracterizados pela sua estratégia utilizada para obter os *feedbacks* dos usuários e pela utilização das informações obtidas desses *feedbacks* pelo motor de busca. Existem três estratégias básicas de coleta dos *feedbacks*: *feedback* explícito, *feedback* implícito e *pseudo-feedback*.

Depois de receber os *feedbacks* de relevância, o algoritmo seleciona e prepara quais informações desses *feedbacks* são de interesse do algoritmo. Esta seleção é um pré-processamento muito simples realizado sobre os dados dos *feedbacks* e dependerá muito de como esses *feedbacks* serão utilizados pela busca. Depois de obtidas as informações dos *feedbacks*, o motor de busca as utiliza a partir de duas perspectivas diferentes: orientada a consultas ou orientada a documentos.

4.1 Coleta de *feedback*

Os algoritmos que utilizam os *feedbacks* de relevância possuem diferentes estratégias para obter esses *feedbacks* dos usuários. A escolha por uma estratégia ou outra influencia a implementação não só do motor de busca em si, mas também da interface gráfica da visualização dos resultados da consulta, principalmente diante da escolha pela estratégia mais tradicional de *feedback* explícito.

Feedback explícito

Os primeiros trabalhos desenvolvidos na área de *feedback* de relevância utilizavam a estratégia de *feedback* explícito para recuperar os *feedbacks* (ROCCHIO, 1971; SALTON, 1971) e alguns dos trabalhos mais recentes também utilizam esta mesma estratégia (SINGH; RAJPAL, 2015; ZHANG; DONG; LIU, 2015; MA; LIN, 2014). Uma das vantagens desta estratégia é a simplicidade e alta confiança nas informações dos *feedbacks* recebidos, já que as outras estratégias não dialogam diretamente com os usuários (*feedback* implícito) ou não dialogam de forma alguma (*pseudo-feedback*).

A estratégia de *feedback* explícito consiste em perguntar diretamente ao usuário quais documentos ele considera relevante ou não. Depois do usuário realizar uma consulta inicial no sistema, ele pode julgar os resultados devolvidos pelo motor de busca e retroalimentar o sistema informando explicitamente quais desses documentos são relevantes para a consulta feita. Assim, o sistema pode processar esses *feedbacks* e agir de acordo, a ação mais comum tomada é a realizar uma nova consulta considerando as novas informações recebidas.

O objetivo é que o usuário julgue alguns dos primeiros resultados, assim, uma nova e melhor consulta é processada. O usuário precisa explicitamente apontar os documentos relevantes ou não entre aqueles devolvidos pela consulta inicial. Portanto, as pesquisas de novos mecanismos de *feedback* de relevância (relacionadas ao uso dos *feedbacks* e não à coleta deles) se apoiam na confiabilidade das informações obtidas através do *feedback* explícito para realizar os experimentos e ilustrar o funcionamento do mecanismos proposto.

Feedback implícito

Poucos usuários utilizam efetivamente o sistema de *feedback* e julgam a relevância dos documentos diante da consulta inicial, apesar do valor deste mecanismo (LAGUN et al., 2013; JANSEN; SPINK; SARACEVIC, 2000). Desse modo, surgiu a necessidade de implementação de um mecanismo de *feedback* implícito, ou seja, surgiu a necessidade do sistema obter informações sobre a relevância dos documentos percebida pelos usuários sem fazê-los a explicitamente julgar os documentos e enviar *feedbacks*. Isto é realizado tanto através de rastreamento dos olhos do usuário quanto através da análise dos cliques realizados pelo usuário nas páginas do sistema após a realização de consultas (JOACHIMS

et al., 2007; RADLINSKI; JOACHIMS, 2005).

A análise dos cliques realizados pelo usuário nas páginas do sistema é feita utilizando os logs das próprias aplicações de gerenciamento de acervos digitais, sem exigir ações extras dos usuários do acervo. Por outro lado, de acordo com alguns autores como Baeza-Yates e Ribeiro-Neto (2011), Joachims et al. (2007), Radlinski e Joachims (2005), os cliques dos usuários em documentos resultantes da busca não devem ser interpretados como indicação de relevância desses documentos diante desta busca. Isso acontece por causa do fator de "alta confiança no motor de busca", ou seja, a confiança de um usuário no motor de busca faz com que a posição de um documento dentro do ranking influencie a escolha do usuário.

Portanto, os cliques feitos por um usuário devem ser analisados comparativamente em relação aos outros documentos devolvidos pela busca, principalmente em relação aos documentos ordenados antes do documento que recebeu o clique. Então, se o usuário abriu um documento, isto é interpretado como uma preferência do usuário por este documento sobre os outros. Na seguinte situação, o usuário recebeu cinco documentos como resposta a uma consulta e ele abriu somente o terceiro (d_3).

$$d_1 \; - \; d_2 \; - \; \boxed{d_3} \; - \; d_4 \; - \; d_5$$

A partir dessa situação, é possível seguir duas estratégias: "ignorar tudo acima" ou "ignorar anterior". Utilizando a estratégia "ignorar tudo acima", é possível afirmar que o documento d_3 é preferido (ou mais relevante) em relação ao d_2 e d_1. Já utilizando a estratégia "ignorar anterior". afirma-se que o documento d_3 é preferido em relação ao documento d_2 somente. Desse modo, escolhendo uma das duas estratégias, os cliques do usuário são utilizados como uma medida de relevância relativa. O documento preferido é comparado com os preteridos e a "diferença" encontrada entre os termos presentes nos documentos comparados é utilizada como *feedback* (tanto positivo quanto negativo).

A principal vantagem do uso de *feedback* implícito é o fato de não ser necessário uma interação explícita dos usuários. Ou seja, os usuários devem utilizar o sistema de acervo digital normalmente e o sistema se responsabiliza de inferir a relevância dos documentos a partir do comportamento dos usuários. A principal deficiência desta estratégia é o baixo nível de confiança sobre as informações obtidas desses *feedbacks*.

Pseudo-feedback

A estratégia de *pseudo-feedback*, também chamada de *feedback* de relevância às cegas, permite o uso automático de algoritmos de *feedback* de relevância. Esta estratégia consiste em considerar que os k primeiros resultados da consulta são relevantes e, desse modo, aplicar os algoritmos de *feedback* de relevância utilizando este pressuposto (BAEZA-YATES; RIBEIRO-NETO, 2011; MANNING; RAGHAVAN; SCHüTZE, 2009).

Alguns trabalhos afirmam obter bons resultados com a aplicação desta estratégia em contextos específicos (CAO et al., 2008; LV; ZHAI, 2010; GROC; TANNIER, 2012; CARPINETO; ROMANO, 2012; BHATNAGAR; PAREEK, 2014; LIU et al., 2014). Esta estratégia geralmente melhora principalmente os valores de cobertura dos resultados, trazendo mais resultados mais parecidos com os primeiros resultados da busca. A principal vantagem deste abordagem é ser completamente automática, não dependendo de nenhuma interação anterior dos usuários.

Uma das desvantagens acontece quando a consulta possui termos ambíguos e os k primeiros resultados estão relacionados a somente um dos significados. Neste contexto, o *pseudo-feedback* devolverá como relevante documentos completamente enviesados em direção a este significado, piorando a qualidade da lista de resultados para um usuário que esteja interessado no outro significado da consulta (não explorado) (MANNING; RAGHAVAN; SCHüTZE, 2009).

4.2 *Feedback* orientado a consultas

A maior parte da pesquisa utilizando *feedback* de relevância foi realizada considerando a perspectiva orientada às consultas, na qual as consultas são adaptadas diante das avaliações feitas pelo usuário (BODOFF et al., 2001). O processo básico dos algoritmos de *feedback* de relevância sob esta perspectiva é dividido em quatro etapas: busca utilizando a consulta inicial, recuperação de *feedbacks*, expansão ou modificação da consulta e busca utilizando a consulta modificada.

Durante a primeira etapa dos algoritmos de *feedback* de relevância, o usuário formula uma consulta inicial e o sistema devolve como resposta um conjunto inicial de documentos ordenados de acordo com a relevância inicial percebida. Diante dos resultados dessa consulta inicial acontece a etapa de recuperação de *feedback* seguindo a estratégia de coleta de dados escolhida.

A partir da consulta inicial e do julgamento feito pelo usuário, o algoritmo passa para a próxima etapa (modificação da consulta), na qual o sistema formula uma nova consulta. Esta consulta modificada ($q_{modificada}$) é criada a partir da consulta inicial ($q_{inicial}$) e procura maximizar a distância entre a consulta modificada e os documentos julgados não-relevantes e minimizar a distância entre a consulta modificada e os documentos considerados relevantes. A consulta modificada é representada pela equação de Rocchio (SALTON, 1971).

$$q_{modificada} = \alpha q_{inicial} + \beta \frac{1}{\|D_r\|} \sum_{i=1}^{r} d_i - \gamma \frac{1}{\|D_{nr}\|} \sum_{j=1}^{nr} d_j \tag{4.1}$$
$$\text{tal que, } d_i \in D_r \text{ e } d_j \in D_{nr}$$

Onde D_r é o conjunto de documentos julgados relevantes e D_{nr} é o conjunto de documentos julgados não relevantes. Nesta equação, o termo $\beta \frac{1}{\|D_r\|} \sum_{i=1}^{r} d_i$ é o *feedback* positivo (ou seja, está relacionado ao documentos marcados como relevantes) e o termo $\gamma \frac{1}{\|D_{nr}\|} \sum_{j=1}^{nr} d_j$ é o *feedback* negativo (relacionado aos documentos marcados como não relevantes).

De acordo com Zhai (2008), o algoritmo de Rocchio ainda é uma forte base para tarefas genéricas de recuperação de informação utilizando *feedbacks* de relevância. Assim como afirmam Baeza-Yates e Ribeiro-Neto (2011), Salton e Buckley (1990), as informações obtidas dos documentos considerados relevantes (*feedback* positivo) são mais importantes do que as obtidas a partir dos documentos considerados não relevantes (*feedback* negativo), por isso o valor de γ é menor que o valor de β.

Manning, Raghavan e Schütze (2009) define que os valores ótimos para α, β e γ são: $\alpha = 1$, $\beta = 0.75$ e $\gamma = 0.15$. Por outro lado, Salton e Buckley (1990) sugerem que ótimos valores para β e γ são 0.75 e 0.25 respectivamente. Os valores de α, β e γ podem ser modificados de acordo com o sucesso em obter *feedback* de relevância do usuário, por exemplo, caso um usuário julgue muitos documentos, os valores de β e γ podem ser maiores, aumentando a importância do *feedback* dado pelo usuário em relação à consulta inicial (BAEZA-YATES; RIBEIRO-NETO, 2011).

Uma das deficiências do uso desta perspectiva é o fato dos *feedbacks* de relevância para uma determinada consulta serem utilizados apenas para aquelas consultas dentro daquela sessão do usuário. Ou seja, mesmo que um usuário em uma sessão realize todo o processo de envio de *feedbacks* diante de uma consulta, as informações dos *feedbacks* recebidos não são utilizadas para melhorar as próximas consultas em sessões posteriores (sejam do mesmo usuário ou não).

Por um lado, este fato permite que o mecanismo de *feedback* de relevância individualize as informações de cada usuário e, desse modo, as preferências e necessidades de um usuário não influenciam as consultas de outros. Por outro lado, os usuários são capazes de entender a semântica por trás da consulta realizada e dos documentos devolvidos, e melhor avaliar a relevância destes documentos. Então, os *feedbacks* dos usuários são uma importante fonte de informações sobre os documentos do acervo e podem ser aproveitados além da sessão do próprio usuário.

4.3 *Feedback* orientado a documentos

Uma segunda abordagem de utilização das informações obtidas a partir dos *feedbacks* de relevância é orientada a documentos, na qual os *feedbacks* são utilizados para modificar a representação dos documentos. Esta perspectiva é apresentada principalmente como uma solução para a deficiência da perspectiva orientada a consultas, pois os *feedbacks* modificam a representação dos documentos e essa modificação é permanente e influenciará consultas realizadas nas sessões dos próximos usuários (BRAUEN; HOLT; WILCOX, 1968a).

O processo básico desta perspectiva também é dividido em quatro etapas, mas neste caso as etapas não precisam acontecer durante uma única sessão de usuário: consulta inicial, recuperação de *feedbacks*, modificação da da representação dos documentos e busca sobre nova representação dos documentos.

Os dois primeiros passos são iguais aos passos da perspectiva orientada a consultas. Mas, no terceiro passo, a modificação é feita sobre a representação dos documentos e não sobre a consulta. Então, partindo da consulta inicial e dos julgamentos devolvidos pelo usuário, o sistema modifica as representações dos documentos, tentando minimizar a distância entre os documentos considerados relevantes e a consulta. Brauen, Holt e Wilcox (1968a) propôs o algoritmo básico para estas modificações, seguindo a equação 4.2.

$$d_i^{novo} = d_i + \alpha(q_j - d_i) \tag{4.2}$$

Onde d_i é um documento considerado relevante diante da consulta q_j e d_i^{novo} é a nova representação deste documento. De acordo com Bot e Wu (2004), o valor ótimo de α é $0,2$.

Por esta abordagem de utilização dos *feedbacks* de relevância modificar a representação dos documentos julgados pelo usuário, muitos dos documentos são ignorados, pois dificilmente um usuário irá julgar cada um dos documentos devolvidos pela busca. Esta deficiência pode levar a documentos relevantes nunca serem julgados e serem preteridos de consultas futuras.

4.4 Estado da arte

Os primeiros trabalhos relacionados à modificação ou expansão de consultas dos usuários (adicionando à consulta novos termos, relacionados aos termos da consulta original) são creditados a Maron e Kuhns (1960). A partir desse trabalho, Rocchio (1971) estudou a expansão de consultas a partir do *feedback* de usuários e incluiu na fórmula de expansão os pesos dos termos do modelo VSM , desenvolvendo assim o algoritmo de Rocchio. Este algoritmo ainda é amplamente utilizado na maioria dos mecanismos de

recuperação de informação (JOACHIMS, 1997; WANG et al., 2013) mesmo que diversas alterações sobre o algoritmo básico foram estudadas.

As contribuições realizadas a partir ou sobre o algoritmo de Rocchio podem ser divididas em 4 grupos, sendo eles:

- Modelo probabilísto;
- *Feedback* de relevância implícito;
- *Pseudo-feedback*;
- Alterações sobre o algoritmo de Rocchio padrão.

Modelo probabilístico

Baseado no algoritmo de Rocchio e nos conceitos de *feedbacks* de relevância, foi desenvolvido o modelo probabilístico de recuperação de informação (ROBERTSON; JONES, 1976), melhor descrito na Seção 2.3. As contribuições deste grupo não estão diretamente relacionadas ao algoritmo de Rocchio, pois somente basearam-se nele para o desenvolvimento deste novo modelo, portanto não serão profundamente exploradas.

O modelo probabilístico utiliza conceitos da teoria de probabilidade e os *feedbacks* obtidos dos usuários para modificar os pesos dos termos da consulta na tentativa de melhorar os resultados da busca de acordo com as necessidades dos usuários (ROBERTSON; JONES, 1976). Mas este modelo não inclui a expansão de consultas, pois no modelo probabilístico os *feedbacks* influenciam os pesos da consulta sem adicionar novos termos a essa consulta. Então, a expansão de consultas é feita separadamente, como estudaram HARPER e RIJSBERGEN (1978), WU e SALTON (1981).

Feedback de relevância implícito

Após o desenvolvimento do algoritmo de Rocchio, algumas dificuldades foram identificadas quando o usuário precisa utilizar esse mecanismo. Como já detalhado na seção 4.1, poucos usuários acabam realmente devolvendo *feedbacks* para o sistema. Desse modo, o uso de *feedback* implícito torna possível a melhoria dos resultados da busca sem que o usuário seja obrigado a realizar ações extras dentro do sistema.

As principais contribuições neste grupo foram realizadas por Joachims como parte de suas pesquisas sobre o uso de *feedback* implícito através da análise dos *clicks* dos usuários no sistema (RADLINSKI; JOACHIMS, 2005; JOACHIMS et al., 2007; RADLINSKI; KURUP; JOACHIMS, 2008; RAMAN; JOACHIMS, 2013). De acordo com Joachims et al. (2007), o uso de algoritmos de aprendizagem de máquina em mecanismos de *feedback* de relevância não é incomum (WHITE; RUTHVEN; JOSE, 2002; RADLINSKI; JOACHIMS, 2005;

BURGES et al., 2005; DESELAERS et al., 2008), mas a quantidade de dados disponíveis para treinamento desses algoritmos é muito pequena, dificultando a aplicação e teste desses algoritmos.

Desse modo, a análise dos *clicks* feitos pelos usuários permite ao sistema coletar muito mais dados e utilizá-los em algoritmos de aprendizagem de máquina. Joachims et al. (2007), além dos *clicks* dos usuários, utilizam um mecanismos de rastreamento dos olhos que permite observar a movimentação dos olhos dos usuários durante os experimentos. A partir da análise da movimentação dos olhos e dos *clicks* foi possível entender melhor o comportamento dos usuários em uma página de busca, incluindo, quais documentos receberão atenção e quais receberam *clicks*, em qual ordem os documentos receberão atenção, quais documentos receberam atenção dos usuários antes de escolherem *clickar* em um documento específico. A partir da análise do comportamento dos usuários, Joachims et al. (2007) concluíram que os *feedbacks* implícitos não devem ser interpretados como relevância absoluta de um documento, mas sim como uma relevância comparativa, ou seja, o documento *clickado* deve ser considerado relevante em relação aos outros documentos que aparecem antes dele na página de busca.

Diante desses resultados obtidos por Joachims et al. (2007), foram desenvolvidos uma série de novos mecanismos de busca utilizando aprendizado de máquina e *feedback* implícito (BRANDT et al., 2011; RAMAN; JOACHIMS; SHIVASWAMY, 2011; RAMAN; SHIVASWAMY; JOACHIMS, 2012; RAMAN; JOACHIMS, 2013).

Pseudo-feedback

Diferente do *feedback* de relevância implícito, o *pseudo-feedback* não precisa de interações com o usuário e se retro-alimenta dos resultados da própria busca realizada (BAEZA-YATES; RIBEIRO-NETO, 2011; MANNING; RAGHAVAN; SCHüTZE, 2009).

As principais contribuições neste grupo estão relacionadas principalmente à seleção de quais documentos ou quais termos serão automaticamente selecionados e aplicados como *feedback* na geração de uma nova consulta (BHATNAGAR; PAREEK, 2014; CAO et al., 2008; LIU et al., 2014; PARAPAR; PRESEDO-QUINDIMIL; BARREIRO, 2014; YE; HUANG, 2014).

Bhatnagar e Pareek (2014), Cao et al. (2008), Ye e Huang (2014) sugerem o uso de algoritmos de aprendizagem de máquina para realizar a seleção de quais termos devem entrar na expansão da consulta. Já Parapar, Presedo-Quindimil e Barreiro (2014) sugerem alterar a seleção de quais documentos devem ser utilizados como *pseudo-feedback*.

Alterações sobre o algoritmo de Rocchio padrão

Diversas variações foram estudadas e propostas sobre o algoritmo de Rocchio padrão, as principais contribuições deste grupo podem ser divididas em dois grupos: (i) aquelas diretamente relacionadas à equação de Rocchio (Equação 4.1) e (ii) aquelas variações no modo como os *feedbacks* alteram o modelo. Essas contribuições levam em consideração o uso de *feedback* explícito, mas como relacionadas à base do algoritmo de Rocchio padrão podem ser adaptadas ou aplicadas para o uso de *feedback* implícito ou *pseudo-feedback*.

As principais contribuições que alteram o funcionamento da equação de Rocchio foram feitas por Ide (1971), Buckley e Salton (1995), Lv e Zhai (2009).

Ide (1971) propôs e testou algumas variações sobre a equação de Rocchio, relacionadas os termos α, β e γ presentes na Equação 4.1. O autor testou por exemplo, uma dessas variações define as constantes da equação de Rocchio como $\alpha = \beta = \gamma = 1$. Nesta variação, os *feedbacks* influenciam a nova consulta com o mesmo fator dos termos da consulta original, ou seja, os *feedbacks* influenciam a consulta final na mesma intensidade que a própria consulta feita inicialmente. Uma segunda variação proposta por Ide (1971) sugere a utilização de somente um documento como feedback negativo, deste modo há um único ponto bem definido do qual a nova consulta deve ser afastada.

Depois de Ide (1971), novas variações relacionadas diretamente à equação do algoritmo de Rocchio foram propostas e se preocupam em tornar a escolha dos valores das constantes adaptativa, ou seja, as constantes passam a ser otimizadas de acordo com a consulta realizada e os *feedbacks* recebidos (BUCKLEY; SALTON, 1995; LV; ZHAI, 2009).

Buckley e Salton (1995) utilizam inicialmente a equação padrão de Rocchio e introduzem uma pequena variação sobre o peso de um dos termos dos *feedbacks* e testam os resultados dessa variação sobre um conjunto de documentos de treinamento. Se esta pequena variação melhorar os resultados diante deste conjunto de treinamento, a alteração é mantida e será aplicada sobre a consulta original, caso contrário, ela é revertida. Então, pequenas variações são testadas consecutivamente do mesmo modo para cada termo dos documentos recebidos como *feedback*.

Lv e Zhai (2009) calculam dinamicamente os valores das constantes α e β de acordo com a consulta e os *feedbacks* recebidos. Este cálculo é realizado levando em consideração três simples heurísticas: nível de especifidade da consulta, nível de especifidade dos documentos dos *feedbacks* e divergência entre consulta e *feedbacks*. Sendo que quanto maior os valores de cada uma dessas heurísticas, maior é a confiança sobre os *feedbacks* recebidos e maior é o valor atribuído a β na equação de Rocchio. A especifidade de uma consulta é calculada com base no seu tamanho e clareza. A especifidade dos *feedbacks* é calculada com base na quantidade de documentos presentes no *feedback* e na distância euclidiana

entre esses documentos (quanto menor a distância, mais focado em um tópico específico esses *feedbacks* são considerados). A divergência entre consulta e *feedbacks* é calculada como a distância euclidiana dos vetores que representam a consulta e os documentos dos *feedbacks*.

Uma segunda linha de pesquisa utiliza modos alternativos no uso dos *feedbacks*, ou seja, no modo como esses *feedbacks* influenciam as consultas realizadas (BRAUEN; HOLT; WILCOX, 1968b; FUHR; BUCKLEY, 1991; FUHR; BUCKLEY, 1990; BOT; WU, 2004). Estas contribuições desenvolveram a estratégia orientada a documentos descrita na Seção 4.3.

O algoritmo de *feedback* de relevância orientado a termos proposto nesta dissertação é uma contribuição dentro deste subgrupo. Este algoritmo introduz uma estratégia diferentes para o uso dos *feedbacks* recebidos dos usuários conforme descrito na Seção 5.

5 ALGORITMO DE *FEEDBACK* DE RELEVÂNCIA ORIENTADO A TERMOS

As abordagens orientadas a consultas possuem uma deficiência no fato dos *feedbacks* recebidos dos usuários serem utilizados somente durante as sessões do próprio usuário, descartando essas informações ao final dessas sessões. Já as abordagens orientadas a documentos possuem uma deficiência relacionada à negligência dos documentos que não foram visitados pelos usuários, pois esses documentos negligenciados serão preteridos nas próximas consultas semelhantes e, mesmo se forem relevantes, dificilmente os usuários visualizarão esses documentos.

Assim, o algoritmo de *feedback* de relevância orientado a termos tem como objetivo desenvolver uma abordagem para o aproveitamento dos *feedbacks* de relevância recebidos dos usuários (abordagem orientada a termos, na qual os *feedbacks* serão utilizados para modificar a representação dos termos no modelo) e uma nova abordagem para o significado semântico da relação entre os termos (na qual a relação entre termos representa no modelo as informações extraídas dos *feedbacks*). Nesta abordagem proposta, uma relação entre dois termos i e j no sistema significa que existe uma relação entre um termo de uma das consultas e um termo de um dos documentos considerados relevantes pelo usuário diante desta consulta.

5.1 Descrição

O algoritmo de *feedback* de relevância orientado a termos é baseado no modelo VSM e também representa documentos e consultas como vetores de n dimensões, onde n é o tamanho do vocabulário. Porém o algoritmo de *feedback* de relevância orientado a termos representa de maneira diferente os vetores de termos e esse vetores refletem as informações obtidas dos *feedbacks* recebidos dos usuários.

Enquanto no modelo VSM simplificado cada um dos termos é representado por um dos vetores unitários e ortogonais entre si que formam a base vetorial do espaço de representação dos documentos (WONG; ZIARKO; WONG, 1985; SALTON, 1989). No algoritmo de *feedback* de relevância orientado a termos, os vetores de termos não são necessariamente ortogonais entre si. Cada um desses vetores (v_i) de n dimensões representa cada um dos termos (t_i) do vocabulário. O valor (v_{ij}) do vetor v_i representa o fator de relação entre o termo t_i e o termo t_j, que deve ser o mesmo valor para os termos t_j e t_i (v_{ji}).

O funcionamento do algoritmo é dividido em 4 etapas:

- Indexação;
- Coleta de *feedback*;
- Transformação dos vetores de termos;
- Busca.

5.2 Indexação

A indexação funciona como o modelo VSM apresentado por Salton (1989), no qual o peso de cada termo sobre os documentos é medido utilizando o *tf-idf*. tf_{ik} se refere à frequência do termo k dentro do documento i (normalizado usando uma função logarítmica ou para a quantidade total de termos do documento) e idf_k se refere ao inverso da frequência de documentos que possuem o termo k (normalizado usando uma função logarítmica). Assim, cada w_{ik} é calculado da seguinte maneira:

$$tf_{ik} = 1 + \log_2 f_{ik} \tag{5.1}$$

$$idf_k = \log_2 \frac{|D|}{|\{d_i \in D : f_{ik} > 0\}|} \tag{5.2}$$

$$tfidf_{ik} = tf_{ik}.idf_k$$

$$tfidf_{ik} = (1 + \log_2 f_{ik}).\log_2 \frac{|D|}{|\{d_i \in D : f_{ik} > 0\}|} \tag{5.3}$$

Onde f_{ik} é a frequência (ou contagem de ocorrências) bruta do termo k no documento i, $|D|$ é cardinalidade do conjunto total de documentos do acervo (ou a quantidade total de documentos no acervo) e $|\{d_i \in D : f_{ik} > 0\}|$ é a quantidade de documentos do acervo que possuem pelo menos uma ocorrência do termo k.

Nesta etapa também, os vetores de termos já são criados e inicializados (seguindo a função 5.4) de modo que não exista nenhuma relação entre os termos, somente entre eles mesmos.

$$v_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{5.4}$$

5.3 Coleta de *feedback*

Esta etapa é responsável pela coleta ou recuperação dos *feedbacks* fornecidos pelos usuários. Os *feedbacks* dos usuários informam quais documentos foram considerados

relevantes diante de quais consultas. O usuário, após realizar uma consulta no sistema, pode indicar quais documentos ele considera relevante diante desta consulta. Assim, cada *feedback* de um usuário possui uma consulta (que este usuário realizou) e um documento (que foi considerado relevante para a própria consulta pelo usuário). Se o usuário selecionar mais de um documento relevante, são geradas várias entrada de *feedback*, cada uma com um documento selecionado.

A partir de cada um dos *feedbacks* são formados dois conjuntos de termos:

- o conjunto de termos que formam a consulta realizada;
- o conjunto dos 20 termos mais representativos do documento considerado relevante.

A escolha de quais termos são considerados os mais representativos em um documento é determinada pelo valor de *tf-idf* dos termos do documento. Os termos com maiores valores de *tf-idf* são considerados os mais representativos, porque equilibram uma alta ocorrência dentro do documento analisado e uma baixa ocorrência em outros documentos do acervo. O conjunto dos 20 termos representativos é suficientemente grande para permitir uma melhora na qualidade da busca (HARMAN, 1992).

No passo seguinte, é gerado o produto cartesiano entre os dois conjuntos, formando pares ordenados no formato (t_i, t_j) que relacionam um dos termos t_i da consulta e um dos termos t_j dos documentos. O conjunto de todos os pares ordenados gerados a partir de todos os *feedbacks* recebidos forma o conjunto chamado F e cada um dos pares ordenados deste conjunto F é analisado e utilizado nas transformações sobre os vetores de termos, descritas a seguir.

5.4 Transformação dos vetores de termos

Cada um dos pares ordenados do conjunto F gerado durante a etapa anterior afeta os vetores de termos, exatamente nos pontos que representam a relação entre os dois termos do par ordenado. Para cada par $(t_i, t_j) \in F$, são modificados os valores v_{ij} e v_{ji} nos vetores de termos, seguindo a Equação 5.5. Se o valor já presente na matriz é maior que zero, o novo valor é a multiplicação do valor antigo por $(1+\beta)$, caso contrário, o novo valor é igual .

$$v_{ij}^{novo} = v_{ji}^{novo} = \begin{cases} v_{ij} + \alpha \times v_{ij} & \text{se } v_{ij} >= 0 \\ \beta & \text{se } v_{ij} >= \beta \end{cases} \tag{5.5}$$

Foram realizados experimentos durante esta pesquisa com o objetivo de analisar diferentes valores de β. Os melhores resultados foram obtidos utilizando $\beta = 0,4$ e o

valor de α é definido a partir do valor de β e da quantidade de *feedbacks* disponíveis. Os resultados dos experimentos estão descritos na Seção 7.

5.5 Busca

A busca é realizada seguindo o modelo VSM completo e, portanto, utilizando a seguinte equação (igual à Equação 2.9) para o cálculo de similaridade entre um documento e uma consulta, sendo que essa similaridade representa a relevância deste documento diante desta consulta. Desse modo, a lista de documentos devolvidos pelo motor de busca é ordenada em ordem decrescente de relevância.

$$d_i \cdot q_j = \sum_{k,m=1}^{n} d_{ik} q_{jm} (T_k \cdot T_m) \tag{5.6}$$

5.6 Exemplo simplificado

Esta seção se propõe a ilustrar a diferença entre ambas abordagens de *feedbacks* de relevância permanentes: orientada a documentos e orientada a termos.

Considerando o seguinte vocabulário, os documentos indexados somente pelo título e consulta inicial ("computação xadrez"):

$$\text{vocabulário} = \{\text{IA}, \text{computação}, \text{xadrez}\}$$

$$d_1 = \{1, 0, 0\}$$
$$d_2 = \{0, 0, 1\}$$
$$d_3 = \{1, 1, 0\}$$

$$q = \{0, 1, 1\}$$

Assim, o resultado da consulta inicial devolve os documentos d_2 e d_3 e o usuário julgou como relevante o documento d_3. Desse modo, seguindo um algoritmo orientado a documentos, o documento d_3 terá seu vetor modificado.

$$d_{3modificado} = \{1, 1.5, 1.5\}$$

Uma nova consulta pelos mesmos termos "computação xadrez", devolve novamente os mesmo documentos d_3 e d_2, mas priorizando o documento d_3. Porém, neste exemplo, o documento d_1 também é relevante, mas não possui nenhum dos termos da consulta e, por este motivo, nunca será devolvido pela busca.

Mas considerando a abordagem orientada a termos para os mesmos vocabulário, documentos e consulta. Os vetores iniciais que representam os termos estão descritos a seguir:

$$T_1 = \{1, 0, 0\}$$
$$T_2 = \{0, 1, 0\}$$
$$T_3 = \{0, 0, 1\}$$

Como esses são vetores ortonormais iguais àqueles considerados pelo VSM simplificado, a consulta inicial devolverá o mesmo resultado (os documentos d_2 e d_3). Se o usuário fizer o mesmo julgamento (indicando o documento d_3 como relevante). Na abordagem orientada a termos, as seguintes relações entre termos terão seus valores aumentados:

$$(\text{IA}, \text{computação})$$
$$(\text{IA}, \text{xadrez})$$
$$(\text{computação}, \text{xadrez})$$

Neste exemplo todos os termos acabaram relacionados. Desse modo, considerando um aumento na relação entre termos de 0.15, os novos vetores de termos terão as seguintes representações:

$$T_{1modificado} = \{1, 0.15, 0.15\}$$
$$T_{2modificado} = \{0.15, 1, 0.15\}$$
$$T_{3modificado} = \{0.15, 0.15, 1\}$$

Então, neste exemplo considerando o *feedback* de relevância orientado a termos, o documento d_1 passará a aparecer como resultado da consulta. Porque mesmo que o termo para o qual ele foi indexado não esteja presente na consulta, este termo possui alguma relação com os termos que estão.

6 EXPERIMENTOS

Como prova de conceito para demonstrar o funcionamento do algoritmo de *feedback* de relevância orientado a termos sugerido, o desempenho do algoritmo proposto é comparado com o desempenho do modelo VSM simplificado diante de experimentos sobre um *dataset* (conjunto de dados) desenvolvido para pesquisas em recuperação de informação.

O experimento realizado compara o desempenho do algoritmo proposto com o modelo VSM simplificado, porque tem como objetivo demonstrar que o uso de *feedbacks* de relevância orientados a termos podem melhorar a qualidade dos resultados. Como esta abordagem é orientada a termos, este algoritmo proposto possui características permanentes e globais, diferente das abordagens anteriores (orientada a documentos e orientada a consultas), como explicado na Seção 5. A comparação dos resultados obtidos pela busca utilizando as abordagens anteriores é infrutífera, porque a abordagem proposta é inspirada no algoritmo de Rocchio. Portanto, se forem utilizados os mesmos *feedbacks*, os resultados obtidos em todas as abordagens serão similares. A diferença reside no modo como esses *feedbacks* serão armazenados e reaproveitados pelo sistema.

O *dataset* utilizado no experimento é formado por um subconjunto dos documentos do *dataset ClueWeb09* (CALLAN et al., 2009) e pelas consultas e julgamentos disponibilizados pelo TREC NIST (NIST, 2015) no eixo temático de *feedback* de relevância (*Relevance Feedback Track*) (BUCKLEY et al., 2010).

Também foi realizado um outro experimento utilizando o mesmo *dataset* com o objetivo de analisar diferentes valores de β para a Equação 5.5 de transformação dos vetores de termos no algoritmo proposto e como os diferentes valores de β afetam os resultados da busca devolvidos pelo algoritmo. Este experimento permitiu avaliar os melhores valores de β, que foram utilizados no experimento comparativo entre o algoritmo de *feedback* de relevância orientado a termos e o VSM.

Além desses experimentos, foi realizado um experimento de comparação do desempenho do algoritmo proposto e do VSM simplificado em relação ao tempo de computação. Estes experimentos incluem o tempo de computação necessário para pré-processar todos os *feedbacks* obtidos no algoritmo de relevância orientado a termos e os tempos de computação necessários para processar as consultas no algoritmo do VSM simplificado e do algoritmo proposto. O algoritmo VSM simplificado não realizada nenhum passo de pré-processamento dos *feedbacks*.

6.1 *Dataset*

Datasets na área de recuperação de informação são formados por um conjunto de documentos (ou acervo), um conjunto de consultas e um conjunto de julgamentos de relevância. Os julgamentos de relevância são indicações de quais documentos são relevantes ou não diante de cada uma das consultas. Na maioria das situações, estes julgamentos são criados manualmente por especialistas da área, que julgam um a um os documentos diante de cada consulta. Como o processo de criação desses julgamentos é muito caro e trabalhoso, este conjunto de julgamentos é geralmente muito pequeno e parcial (nem todos os documentos do acervo são analisados diante de cada consulta) (ZHANG; KAMPS, 2010).

Documentos

O conjunto de documentos *ClueWeb09* possui 1.040.809.705 de documentos em 10 diferentes línguas e esta quantidade torna o processo de indexação muito custoso trazendo poucas vantagens para a análise comparativa dos algoritmos de recuperação de informação. As poucas consultas e julgamentos disponibilizados pelo TREC NIST cobrem somente uma parte muito pequena desse conjunto e são todas feitas em inglês, por isso, os experimentos foram realizados somente sobre um subconjunto de documentos em inglês do *dataset*. Este subconjunto, que foi utilizado nos experimentos, é formado pelos primeiros 1.000 documentos devolvidos por um mecanismo de busca simples (utilizando um algoritmo de VSM simplificado, descrito mais detalhadamente na Seção 2.1) diante de cada uma das 100 consultas disponibilizadas pelo TREC NIST mais os documentos julgados pelos especialistas, totalizando 101.479 documentos.

Os documentos do *dataset ClueWeb09* são páginas coletadas em 2009 e incluem páginas da Wikipédia e páginas informativas aleatórias. O contexto deste trabalho são os acervos digitais, por este motivo, as *tags html* dessas páginas foram removidas durante o pré-processamento utilizando uma biblioteca Java de manipulação de HTML chamda *jsoup* (HEDLEY, 2015). Assim, essas páginas *web* adquirem características mais semelhantes com documentos de acervos digitais, tornando-se puramente textuais e sem a presença de hiperlinks.

Consultas e julgamentos

As consultas e julgamentos foram disponibilizadas pelo TREC (Text REtrieval Conference) NIST (National Institute of Standards and Technology) como parte de um projeto de incentivo à pesquisa em recuperação de informação, provendo a infraestrutura necessária para avaliação de novos mecanismos de recuperação de informação em larga escala.

Nos experimentos são utilizadas as consultas e julgamentos de cada uma dessas consultas desenvolvidas para o eixo temático de *feedback* de relevância. Alguns exemplos de consultas realizadas são: "*wedding budget calculator*", "*elvish language*" e "*when did the civil war end*".

Os julgamentos de relevância são indicações de quais documentos são relevantes ou não diante de cada uma das consultas. Esses julgamentos são criados manualmente por especialistas da área, que julgam um a um os documentos diante de cada consulta. Como o processo de criação desses julgamentos é muito caro e trabalhoso, este conjunto de julgamentos é geralmente parcial (ou seja, nem todos os documentos do acervo são analisados diante de cada consulta) (ZHANG; KAMPS, 2010).

Os *feedbacks* de usuários são utilizados para retroalimentar o algoritmo de *feedback* de relevância orientado a termos (alterando a representação dos termos dentro do modelo e, portanto, alterando os resultados das consultas) e os julgamentos de teste são utilizados para comparar os resultados obtidos pelo algoritmo com os julgamentos humanos.

6.2 Avaliação

A avaliação de sistema de recuperação de informações é, tradicionalmente, realizada através da comparação dos resultados produzidos pelo sistema e os resultados sugeridos por usuários ou especialistas diante das mesmas consultas (BAEZA-YATES; RIBEIRO-NETO, 2011; MANNING; RAGHAVAN; SCHüTZE, 2009). Os julgamentos realizados pelos especialistas são considerados verdadeiros e, desse modo, é possível calcular quantos documentos relevantes foram recuperados. Assim, o desempenho de vários sistemas é testado diante dos mesmos acervos e das mesmas consultas, e o desempenho deles é comparado estatisticamente.

As principais métricas utilizadas para a avaliação são a precisão e a cobertura. Onde precisão (p) é a proporção de documentos retornados que são relevantes (Equação 6.1) e cobertura (r) é a proporção de documentos relevantes que são retornados (Equação 6.2). Nesta situação, os julgamentos humanos são considerados verdadeiros e completos, desse modo, os documentos relevantes de um acervo são todos aqueles que foram julgados relevantes por um especialista (BAEZA-YATES; RIBEIRO-NETO, 2011).

$$p = \frac{|D \cap R|}{|D|} \tag{6.1}$$

$$r = \frac{|D \cap R|}{|R|} \tag{6.2}$$

Sendo que D é o conjunto de documentos devolvidos pelo algoritmos e R é o conjunto de documentos julgados relevantes pelo especialista humano.

Como os julgamentos para este *dataset* são parciais, a avaliação comparativa entre diferentes algoritmos não deve considerar somente os valores de cada métrica (precisão e cobertura), pois a métrica de cobertura facilmente atinge 100% (ou seja, todos os documentos determinados como relevantes pelo especialista são recuperados pelo algoritmo) já que poucos documentos são analisados para cada consulta e a própria seleção dos documentos que os especialistas analisaram é feita utilizando algum mecanismo de busca simples.

Os algoritmos de recuperação de informação devolvem, diante de uma consulta, uma lista de documentos ordenada por ordem de relevância, na qual essa relevância é a estimativa de relevância calculada pelo algoritmo e cada documento ocupa uma posição do *ranking* nesta lista. As métricas de precisão e cobertura são métricas calculadas sobre conjuntos, onde a ordenação dos elementos desses conjuntos é irrelevante. No contexto da recuperação de informação, a ordenação dos documentos é um dos fatores mais relevantes para a comparação do desempenho dos algoritmos, portanto é utilizada uma métrica mais sensível à ordenação dos resultados da busca: curva de precisão por cobertura (BAEZA-YATES; RIBEIRO-NETO, 2011; RUTHVEN; LALMAS, 2003).

A curva de precisão por cobertura demonstra a variação da precisão em relação à variação da cobertura. Os valores de cobertura são sempre crescentes percorrendo a lista de documentos devolvida pelo algoritmo partindo da primeira posição do *ranking* até a última. Ou seja, a curva de precisão por cobertura também consegue ilustrar o comportamento de maneira geral da precisão em relação à lista ordenada de documentos devolvida pelo algoritmo. Assim, o desempenho de diferentes algoritmos diante de uma consulta podem ser comparados através da análise de suas respectivas curvas de precisão por cobertura.

O cálculo da precisão por cobertura é realizado calculando-se os valores de precisão e cobertura para cada posição do *ranking* da lista de documentos. Sendo que os valores de precisão e cobertura para a posição x do *ranking* são calculados levando em consideração o conjunto de todos os documentos devolvidos cujas posições no *ranking* sejam menores ou iguais a x. O Algoritmo 1 é um pseudo-código que ilustra este cálculo para uma das consultas em um algoritmo. A entrada do algoritmo são a lista de documentos devolvidos pelo algoritmo de busca diante desta consulta em ordem decrescente de relevância (ou lista de resultados) e a lista de documentos considerados relevantes diante desta consulta (ou lista de relevantes). A saída é a curva de precisão por cobertura.

A análise comparativa da curva de precisão por cobertura de diferentes algoritmos em relação a uma única consulta é pouco representativa, para tanto, a análise deve ser realizada sobre o comportamento desses algoritmos diante de um conjunto de consultas. Para definir um comportamento médio de um algoritmo diante de uma grande quantidade de consultas é necessário realizar o cálculo dos valores médios de precisão por cobertura. Como a quantidade de documentos considerados relevantes não são necessariamente a

Algoritmo 1: Curva de precisão por cobertura de uma consulta

```
1  início
2      recuperados ← 0;
3      relevantes_recuperados ← 0;
4      relevantes ← tamanho da lista de relevantes;
5      enquanto não é o fim da lista de resultados faça
6          leia documento_atual;
7          recuperados ← recuperados + 1;
8          se documento_atual ∈ lista de relevantes então
9              relevantes_recuperados ← relevantes_recuperados + 1;
10         fim se
11         calcular precisão (P ← relevantes_recuperados / recuperados);
12         calcular cobertura (R ← relevantes_recuperados / relevantes);
13         armazenar o par ordenado (P,R);
14         seguir para próximo documento da lista de resultados;
15     fim enquanto
16 fim
```

mesma para todas as consultas, os valores obtidos de cobertura são diferentes para cada consulta e, por isso, são criados os níveis de cobertura.

Os níveis de cobertura são valores de cobertura estabelecidos para os quais são determinados os valores de precisão em cada consulta. Desse modo, utilizando os valores de precisão por nível de cobertura de cada consulta é trivial realizar o cálculos das médias de precisão em cada nível de cobertura, determinando a curva média de precisão por nível de cobertura do algoritmo. Os níveis de cobertura utilizados foram estabelecidos para os experimentos realizados como os valores do conjunto {1%, 2%, 3%, 4%, ... 100%}. Intervalos menores entre os níveis de cobertura permitem visualizar melhor cada mudança de precisão, mas não melhoram a visualização do comportamento geral do algoritmo e adicionam muito ruído à curva; Já intervalos maiores podem esconder pequenas alterações na precisão encontradas entre cada nível de cobertura.

O Algoritmo 2 ilustra, através de pseudo-código, o cálculo da curva de precisão média por nível de cobertura de um algoritmo testado nos experimentos. Inicialmente, o Algoritmo 2 executa o Algoritmo 1 para cada consulta que foi testada durante os experimentos e, posteriormente, são calculados os valores médios de precisão por nível de cobertura.

As médias de precisão calculadas para cada nível de cobertura formam a curva de precisão média por nível de cobertura e podem ser utilizadas na análise comparativa entre diferentes algoritmos de recuperação de informação.

Algoritmo 2: Curva de precisão média por nível de cobertura

```
1  início
2      para cada consulta faça
3          execute Algoritmo 1;
4      fim para cada
5
6      porcentagem ← 1;
7      enquanto porcentagem ≤ 100% faça
8          soma_precisao ← 0;
9          para cada consulta faça
10             encontrar um par ordenado (P',R'), onde
11             R' = máximo(R), tal que R ≤ porcentagem;
12             soma_precisao = soma_precisao + P'
13         fim para cada
14         média = soma_precisao / quantidadedeconsultas;
15         armazenar (média, porcentagem);
16         porcentagem ← porcentagem + 1;
17     fim enquanto
18 fim
```

Exemplo dos cálculos de avaliação

Esta seção ilustra a execução dos pseudo-códigos responsáveis pelo cálculo das curvas de precisão média por nível de cobertura utilizando duas consultas feitas durante o experimento utilizando os resultados devolvidos pelo algoritmo de feedback de relevância orientado a termos desenvolvido. As consultas exemplificada são "jimmy earl carter"e "orlando sentinel", cujos identificadores são 20102 e 20832 respectivamente.

Diante da consulta 20102, o algoritmo devolveu nas dez primeiras posições do *ranking* os seguintes documentos (se diante do identificador do documento existe um asterisco, o documento em questão foi considerado relevante por um especialista). Para esta consulta 15 dos documentos analisados por um especialista foram considerados relevantes.

```
Consulta 20102 ("jimmy earl carter"), resultados:
1 clueweb09-en0004-34-24405*
2 clueweb09-enwp00-19-05525
3 clueweb09-enwp01-10-03030*
4 clueweb09-enwp01-07-07433*
5 clueweb09-enwp01-13-03747
6 clueweb09-enwp01-13-03748
7 clueweb09-enwp01-04-03865
8 clueweb09-enwp02-24-03452*
```

```
9 clueweb09-enwp01-05-03452
10 clueweb09-enwp02-20-07508*
```

A seguir estão os dez primeiros resultados obtidos pelo algoritmo para a consulta 20832. Nesta consulta, 18 dos documentos analisados foram considerados relevantes.

```
Consulta 20832 ("orlando sentinel"), resultados:
1 clueweb09-en0000-83-23420*
2 clueweb09-en0000-83-22212*
3 clueweb09-en0002-97-23838
4 clueweb09-en0004-34-19830
5 clueweb09-en0011-55-06333*
6 clueweb09-en0003-67-03716
7 clueweb09-en0011-55-06341*
8 clueweb09-en0000-71-14573
9 clueweb09-en0011-55-06339
10 clueweb09-en0002-02-25487
```

As duas tabelas a seguir mostram os estados das variáveis do Algoritmo 1 no final de cada iteração do laço que se inicia na linha 5 durante a execução do Algoritmo 1 para a consulta 20102 (Tabela 1) e consulta 20832 (Tabela 2).

Tabela 1: Estados das variáveis no Algoritmo 1 para a consulta 20102

Iterações	recuperados	relevantes_recuperados	relevantes	Precisão	Cobertura
1	1	1	15	1	0,06
2	2	1	15	0,5	0,06
3	3	2	15	0,66	0,13
4	4	3	15	0,75	0,2
5	5	3	15	0,6	0,2
6	6	3	15	0,5	0,2
7	7	3	15	0,42	0,2
8	8	4	15	0,5	0,26
9	9	4	15	0,44	0,26
10	10	5	15	0,5	0,33

6.3 Descrição dos experimentos

Depois de formar o subconjunto de documentos a partir do *ClueWeb09*, este subconjunto de documentos é indexado, seguindo a indexação padrão do modelo VSM simplificado, formando o índice inicial. Durante esta etapa de indexação, as *stopwords*

Tabela 2: Estados das variáveis no Algoritmo 1 para a consulta 20832

Iterações	recuperados	relevantes_recuperados	relevantes	Precisão	Cobertura
1	1	1	18	1	0,05
2	2	2	18	1	0,11
3	3	2	18	0,66	0,11
4	4	2	18	0,5	0,11
5	5	3	18	0,6	0,16
6	6	3	18	0,5	0,16
7	7	4	18	0,57	0,22
8	8	4	18	0,5	0,22
9	9	4	18	0,44	0,22
10	10	4	18	0,4	0,22

foram removidas dos documentos. A lista de *stopwords* utilizada é a mesma lista utilizada na ferramenta Weka de mineração de dados (HALL et al., 2009).

A partir deste índice inicial, o algoritmo de *feedback* de relevância orientado a termos (alimentado pelos *feedbacks* dos usuários) gera um novo índice modificado, no qual os vetores que representam os termos são modificados seguindo o algoritmo descrito na Seção 5.

O índice inicial é utilizado para realizar os experimentos utilizando o algoritmo VSM simplificado. Sobre este índice, o modelo VSM simplificado realiza as consultas e devolve os documentos que o algoritmo considera relevantes. Esta lista de documentos devolvida pelo algoritmo é comparada com os julgamentos dos especialistas, desse modo é possível determinar quais dos documentos devolvidos pelo algoritmo são relevantes e calcular os valores de precisão e cobertura do algoritmo.

O índice modificado é utilizado pelo algoritmo de *feedback* de relevância orientado a termos. Este índice modificado já possui as alterações baseadas nos *feedbacks* dos usuários. As consultas sobre este índice modificado são realizadas como descrito na Seção 5.5. As mesmas consultas realizadas pelo VSM simplificado são realizadas pelo algoritmo proposto, desse modo, é possível comparar os valores de precisão e cobertura obtidos pelos diferentes algoritmos.

Para a realização do experimento realizado sobre a Equação 5.5 em relação aos possíveis valores β da Equação 5.5 (equação de transformação dos vetores de termos) foi utilizado basicamente o mesmo índice modificado. Porém, para cada valor de β testado (0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9), os vetores de termos foram modificados de forma diferente de acordo com o valor de β e as consultas foram feitas sobre o índice modificado. O valor de α pode ser selecionado respeitando $\alpha < \beta$. Nos experimentos, o valor de α utilizado foi selecionado dependendo do valor de β e da quantidade de documentos disponíveis nos *feedbacks* dos usuários. Como nos experimentos existem 10 documentos

disponíveis na forma de *feedback* para cada consulta, foi utilizado o valor $\alpha = \frac{\beta}{10}$. Desse modo, a somatória máxima de α naturalmente será igual a β.

Os experimentos relacionados aos tempos de computação dos dois algoritmos foram realizados em computadores utilizando um processador Intel Core i5-4670 3.4Ghz com 6MB de memória cache, uma memória RAM DDR3 de 8GB (1600MHz de velocidade de barramento) e um disco rígido com 64MB de memória cache e 7200 RPM. Os resultados obtidos por estes experimentos estão na Seção 7 e mostram tempos médios de processamento para cada etapa dos algoritmos testados.

7 RESULTADOS

O experimento realizado sobre os possíveis valores de β da Equação 5.5 mostrou que um valor ótimo para ser utilizado é $\beta = 0,4$. O gráfico da Figura 1 mostra os valores de β testados e os valores de precisão média por nível de cobertura obtidos.

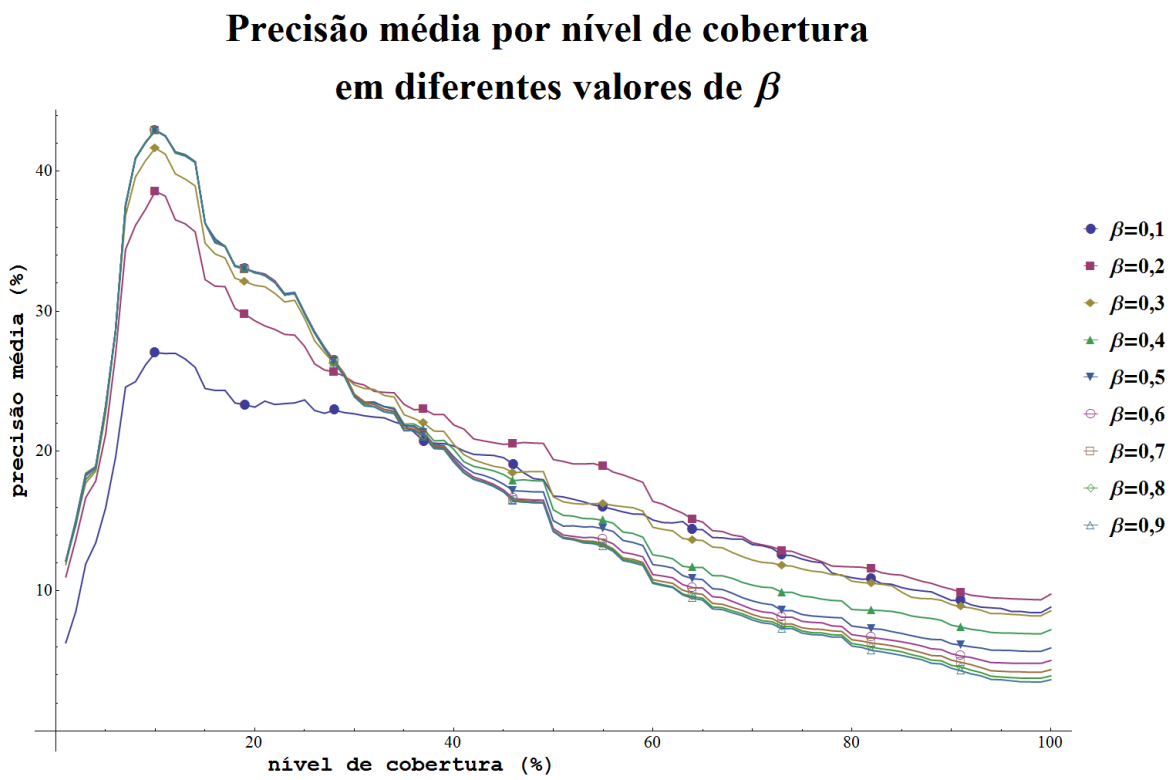

Figura 1: Gráfico comparativo da precisão média por nível de cobertura utilizando diferentes valores de β na transformação dos vetores de termos no algoritmo de *feedback* orientado a termos. Fonte: elaborado pelos autores.

Os valores de precisão por nível de cobertura são muito semelhantes para todos os valores de $\beta \geq 0,4$ no níveis iniciais de cobertura (até 40%), mas piora com o aumento do valor de β para os níveis de cobertura maiores que 40%. Este comportamento ocorre, porque o conjunto de *feedbacks* de relevância é pequeno e, consequentemente, o volume de informações obtidas desses *feedbacks* utilizadas pelo algoritmo proposto também é pequeno e isto se torna um limitante na habilidade do algoritmo em melhorar os resultados da busca e o valor de β não tem grande influência diante deste limitante.

Levando em consideração o valor de $\beta = 0,4$, o algoritmo de *feedbacks* de relevância orientado a termos foi comparado com o VSM simplificado e os resultados do experimento realizado com ambos algoritmos são apresentados no gráfico da Figura 2, que mostra os valores de precisão média por nível de cobertura.

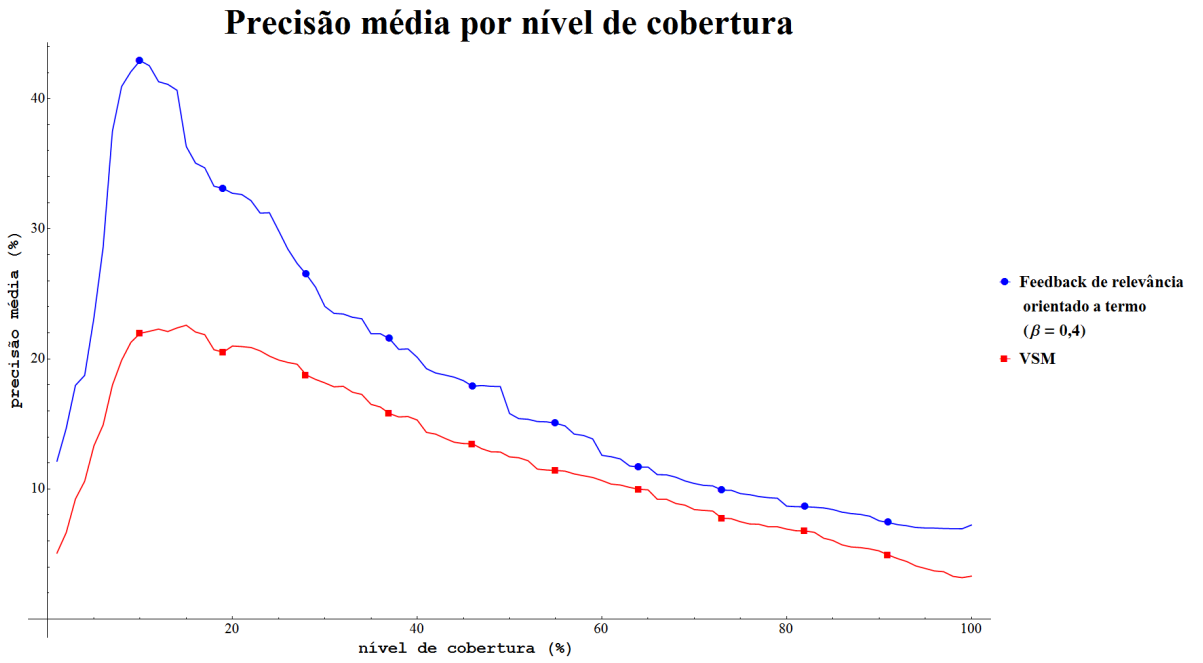

Figura 2: Gráfico comparativo entre o algoritmo de *feedback* orientado a termos e VSM em relação aos valores de precisão média por cobertura. Fonte: elaborado pelos autores.

O algoritmo que utiliza os *feedbacks* de relevância orientados aos termos apresenta uma melhora significativa para os valores de níveis iniciais de cobertura (até 40%). Esta porção inicial dos resultados da busca é considerada a mais relevante para a comparação de diferentes modelos de recuperação de informação, porque a maioria dos usuários normalmente se concentram nos primeiros resultados e ignoram o restante dos resultados (PASS; CHOWDHURY; TORGESON, 2006).

Mesmo utilizando um valor pequeno de $\beta = 0,1$ na Equação 5.5 (equação de transformação dos vetores de termos), o algoritmo melhora os resultados da busca em relação ao VSM simplificado, como é possível observar no gráfico da Figura 3. Demonstrando que o uso de *feedbacks* de relevância podem efetivamente melhorar os resultados de um mecanismo de busca em relação ao modelo mais amplamente utilizado (VSM), mesmo utilizando um valor pequeno no fator β de transformação dos vetores de termos.

O aumento do valor de β (considerando $\beta \leq 0,4$) gerou um aumento também na precisão da busca nos níveis iniciais de cobertura (até 40%), mas piorou nos níveis maiores. Mostrando que as informações obtidas dos *feedbacks* são relevantes e que o aumento de quanto estas informações influenciam a busca pode melhorar os resultados da busca nos níveis iniciais de cobertura até certo valor de β. Porém, por outro lado, essas informações acabam aumentando a quantidade total de documentos recuperados, piorando a precisão nos níveis de cobertura maiores, porque mais documentos que não foram julgados relevantes são recuperados.

A Tabela 3 permite observar mais precisamente alguns os valores de precisão média

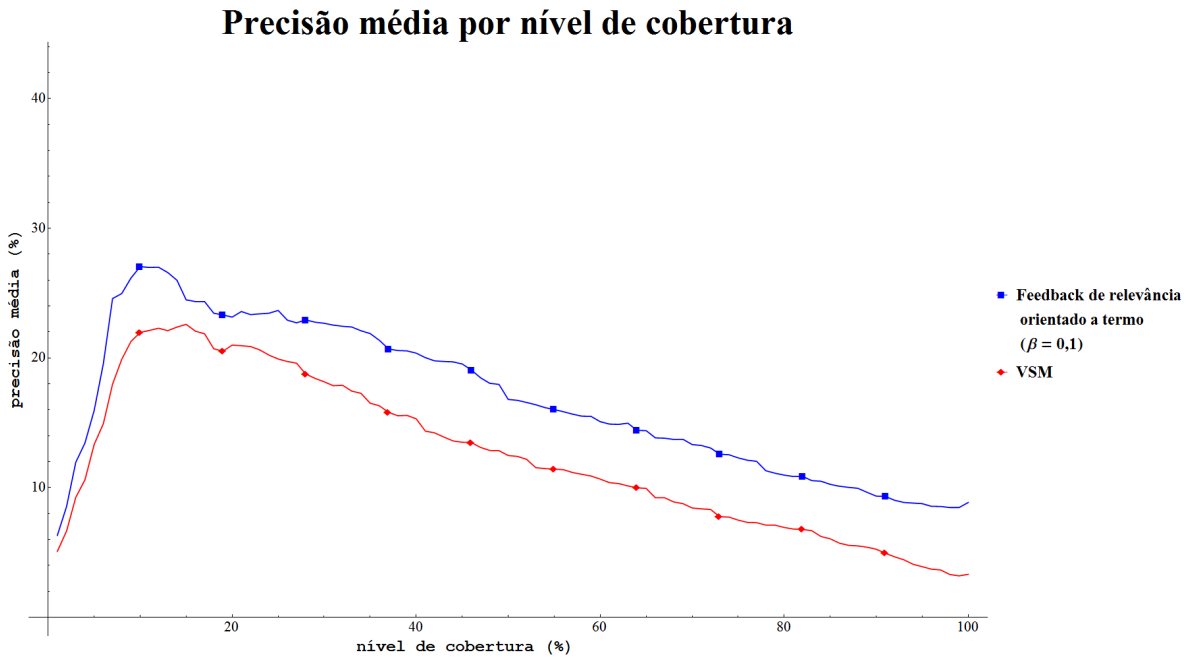

Figura 3: Gráfico comparativo entre o algoritmo de *feedback* orientado a termos e VSM em relação aos valores de precisão média por cobertura. Fonte: elaborado pelos autores.

Tabela 3: Valores de precisão média por nível cobertura (até 40%).

Algoritmos	Nível de cobertura			
	10%	20%	30%	40%
FROT ($\beta = 0,4$)	42,92%	32,72%	24,03%	20,12%
FROT ($\beta = 0,1$)	27,04%	23,15%	22,67%	20,36%
VSM	21,96%	20,99%	18,15%	15,30%

por nível de cobertura (até 40%) obtidos pelos algoritmos de *Feedback* de Relevância Orientado a Termos (FROT), utilizando $\beta = 0,4$ e $\beta = 0,1$, e VSM padrão.

Os resultados obtidos pelos experimentos de tempo de processamento comparando o algoritmo proposto e o algorito do VSM simplificado podem ser visualizados na Tabela 4.

Tabela 4: Tempos de processamento

Etapas do algoritmo		Algoritmo proposto	VSM
Pré-processamento	Média	0,98s/documento	-
	Desvio padrão	0,21s	-
Busca	Média	7,23s	7,15s
	Desvio padrão	5,37s	5,42

Os resultados obtidos em relação aos tempos de processamento dos algoritmos nos permite observar que após o pré-processamento, a etapa de busca utilizando o algoritmo proposto de *feedback* de relevância orientado a termos ou o algoritmo do VSM simplificado

possuem aproximadamente os mesmos tempos de processamento. Isto ocorre, porque após o pré-processamento a complexidade de ambos algoritmos é igual e estes algoritmos processarão a mesma quantidade de informações.

Porém, o algoritmo proposto precisa realizar um pré-processamento dos *feedbacks* recebidos de modo que seu desempenho em relação ao tempo seja igual ao algoritmo VSM. Este pré-processamento pode ser muito custoso e dependerá de quantos documentos são influenciados pelos *feedbacks* recebidos. Nos experimentos realizados, o pré-processamento dos *feedbacks* recebidos levou aproximadamente 3 horas e 37 minutos.

O pré-processamento em sistemas reais é executado durante a madrugada ou em horários com poucos acessos utilizando todos os *feedbacks* obtidos durante o dia. São esperados poucos *feedbacks* por dia, pois a maioria dos usuários não julga muitos documentos e não utiliza frequentemente o sistema de envio de *feedback* (LAGUN et al., 2013; JANSEN; SPINK; SARACEVIC, 2000). Este tempo do pré-processamento pode ser otimizado em sistemas reais utilizando melhores banco de dados e paralelização/distribuição de processos, mas não foram realizadas experimentos em diferentes condições por não ser este o escopo deste projeto de pesquisa.

8 CONCLUSÃO

A Figura 2 e a Tabela 3 evidenciam a capacidade do algoritmo de *Feedback* de Relevância Orientado a Termos em melhorar a qualidade dos resultados de uma busca em relação ao modelo mais amplamente utilizado (VSM), utilizando *feedbacks* de relevância. Quando o nível de cobertura atinge o valor de 10%, onde se concentram a maioria dos acessos e visualizações dos usuários de acordo com Pass, Chowdhury e Torgeson (PASS; CHOWDHURY; TORGESON, 2006), a melhoria é de 95% em relação ao VSM padrão.

A utilização de *feedbacks* de relevância na alteração dos vetores que representam os termos do vocabulário pode melhorar a qualidade dos resultados obtidos. A estratégia de uso dos *feedbacks* permite que as informações obtidas dos usuários sejam armazenadas permanentemente e possam ser utilizadas em sessões de outros usuários, além de serem globais, ou seja, os *feedbacks* influenciam o cálculo de similaridade para todo o acervo e não somente para os documentos julgados pelos usuários.

Os trabalhos futuros incluem a investigação de variações para a realização dos cálculos dos pesos dos termos sobre os documentos (incluindo variações do próprio *tf-idf*) (MANNING; RAGHAVAN; SCHüTZE, 2009) e variações sobre as transformações dos vetores de termos (incluindo o uso de uma equação inspirada na equação de Brauen (BRAUEN; HOLT; WILCOX, 1968a) no lugar do algoritmo de Rocchio (ROCCHIO, 1971)). Além da análise de como modelos de recuperação da informção alternativos capazes de representar da relação entre termos, como por exemplo o uso do GVSM (WONG; ZIARKO; WONG, 1985), se comportarão utilizando *feedbacks* como alimentação dessas relações.

REFERÊNCIAS

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 2nd. ed. USA: Addison-Wesley Publishing Company, 2011. ISBN 9780321416919.

BHATNAGAR, P.; PAREEK, N. Improving pseudo relevance feedback based query expansion using genetic fuzzy approach and semantic similarity notion. *JOURNAL OF INFORMATION SCIENCE*, SAGE PUBLICATIONS LTD, 1 OLIVERS YARD, 55 CITY ROAD, LONDON EC1Y 1SP, ENGLAND, 40, n. 4, p. 523–537, AUG 2014. ISSN 0165-5515.

BLAIR, D. C.; MARON, M. E. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Commun. ACM*, ACM, New York, NY, USA, v. 28, n. 3, p. 289–299, mar. 1985. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/3166-.3197>.

BODOFF, D.; ENACHE, D.; KAMBIL, A.; SIMON, G.; YUKHIMETS, A. A unified maximum likelihood approach to document retrieval. *Journal of the American Society for Information Science and Technology*, John Wiley & Sons, Inc., v. 52, n. 10, p. 785–796, 2001. ISSN 1532-2890. Disponível em: <http://dx.doi.org/10.1002/asi.1137>.

BOT, R. S.; WU, Y.-f. B. Improving document representations using relevance feedback: The rfa algorithm. In: *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2004. (CIKM '04), p. 270–278. ISBN 1-58113-874-1. Disponível em: <http://doi.acm.org/10.1145/1031171-.1031230>.

BRANDT, C.; JOACHIMS, T.; YUE, Y.; BANK, J. Dynamic ranked retrieval. In: *ACM International Conference on Web Search and Data Mining (WSDM)*. [S.l.: s.n.], 2011. p. 247–256.

BRAUEN, T. L.; HOLT, R. C.; WILCOX, T. R. Document indexing based on relevance feedback. *Report ISR-14 to the National Science Foundation, Section XI*, 1968. Department of Computer Science, Cornell University, Ithaca, NY.

BRAUEN, T. L.; HOLT, R. C.; WILCOX, T. R. *Document Indexing Based on Relevance Feedback*. [S.l.], 1968.

BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 30, n. 1-7, p. 107–117, abr. 1998. ISSN 0169-7552. Disponível em: <http://dx.doi.org/10.1016/S0169-7552(98)00110-X>.

BUCKLEY, C.; LEASE, M.; SMUCKER, M. D.; JUNG, H. J.; GRADY, C.; BUCKLEY, C.; LEASE, M.; SMUCKER, M. D.; GRADY, C.; LEASE, M. et al. Overview of the trec 2010 relevance feedback track (notebook). In: *The Nineteenth Text Retrieval Conference (TREC) Notebook*. [S.l.: s.n.], 2010.

BUCKLEY, C.; SALTON, G. Optimization of relevance feedback weights. In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1995. (SIGIR '95), p. 351–357. ISBN 0-89791-714-6. Disponível em: <http://doi.acm.org/10.1145/215206.215383>.

BURGES, C.; SHAKED, T.; RENSHAW, E.; LAZIER, A.; DEEDS, M.; HAMILTON, N.; HULLENDER, G. Learning to rank using gradient descent. In: *Proceedings of the 22Nd International Conference on Machine Learning*. New York, NY, USA: ACM, 2005. (ICML '05), p. 89–96. ISBN 1-59593-180-5. Disponível em: <http://doi.acm.org/10.1145/1102351.1102363>.

CALLAN, J.; HOY, M.; YOO, C.; ZHAO, L. *Clueweb09 data set*. 2009. Disponível em: <http://lemurproject.org/clueweb09/>.

CAO, G.; NIE, J.-Y.; GAO, J.; ROBERTSON, S. Selecting good expansion terms for pseudo-relevance feedback. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2008. (SIGIR '08), p. 243–250. ISBN 978-1-60558-164-4. Disponível em: <http://doi.acm.org/10.1145/1390334.1390377>.

CARPINETO, C.; ROMANO, G. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 44, n. 1, p. 1:1–1:50, jan. 2012. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2071389.2071390>.

DESELAERS, T.; PAREDES, R.; VIDAL, E.; NEY, H. Learning weighted distances for relevance feedback in image retrieval. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. [S.l.: s.n.], 2008. p. 1–4. ISSN 1051-4651.

DOYLE, L. B. Semantic road maps for literature searchers. *J. ACM*, ACM, New York, NY, USA, v. 8, n. 4, p. 553–578, out. 1961. ISSN 0004-5411. Disponível em: <http://doi.acm.org/10.1145/321088.321095>.

DRORI, O. Algorithm for documents ranking: Idea and simulation results. In: *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*. New York, NY, USA: ACM, 2002. (SEKE '02), p. 99–102. ISBN 1-58113-556-4. Disponível em: <http://doi.acm.org/10.1145/568760.568779>.

FUHR, N.; BUCKLEY, C. Probabilistic document indexing from relevance feedback data. In: *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1990. (SIGIR '90), p. 45–61. ISBN 0-89791-408-2. Disponível em: <http://doi.acm.org/10.1145/96749.98008>.

FUHR, N.; BUCKLEY, C. A probabilistic learning approach for document indexing. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 9, n. 3, p. 223–248, jul. 1991. ISSN 1046-8188. Disponível em: <http://doi.acm.org/10.1145/125187.125189>.

GROC, C. de; TANNIER, X. Experiments on pseudo relevance feedback using graph random walks. In: CALDERóN-BENAVIDES, L.; GONZáLEZ-CARO, C.; CHáVEZ, E.; ZIVIANI, N. (Ed.). *String Processing and Information Retrieval*. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7608). p. 193–198. ISBN 978-3-642-34108-3.

GUO, Q. The similarity computing of documents based on vsm. In: _____. *Network-Based Information Systems: 2nd International Conference, NBiS 2008, Turin, Italy, September 1-5, 2008. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 142–148. ISBN 978-3-540-85693-1. Disponível em: <http://dx.doi.org/10.1007/978-3-540-85693-1_16>.

GUPTA, Y.; SAINI, A.; SAXENA, A. K. Fuzzy logic-based approach to develop hybrid similarity measure for efficient information retrieval. *JOURNAL OF INFORMATION SCIENCE*, SAGE PUBLICATIONS LTD, 1 OLIVERS YARD, 55 CITY ROAD, LONDON EC1Y 1SP, ENGLAND, 40, n. 6, p. 846–857, DEC 2014. ISSN 0165-5515.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>.

HARMAN, D. Relevance feedback revisited. In: *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1992. (SIGIR '92), p. 1–10. ISBN 0-89791-523-2. Disponível em: <http://doi.acm.org/10.1145/133160.133167>.

HARPER, D.; RIJSBERGEN, C. V. An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation*, v. 34, n. 3, p. 189–216, 1978. Disponível em: <http://dx.doi.org/10.1108/eb026659>.

HATCHER, E.; GOSPODNETIC, O. *Lucene in Action (In Action series)*. Greenwich, CT, USA: Manning Publications Co., 2004. ISBN 1932394281.

HEDLEY, J. *jsoup: Java HTML Parser*. 2015. Website (`https://jsoup.org/`).

IDE, E. New experiments in relevance feedback. In: SALTON, G. (Ed.). *The SMART Retrieval System–Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice Hall, Englewood Cliffs, 1971. cap. 16.

Index Data. *Zebra search engine*. 2014. Disponível em: <https://www.indexdata.com-/zebra>.

JANSEN, B. J.; SPINK, A.; SARACEVIC, T. Real life, real users, and real needs: A study and analysis of user queries on the web. *Inf. Process. Manage.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 36, n. 2, p. 207–227, jan. 2000. ISSN 0306-4573. Disponível em: <http://dx.doi.org/10.1016/S0306-4573(99)00056-4>.

JäRVELIN, K.; KEKäLäINEN, J. Ir evaluation methods for retrieving highly relevant documents. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2000. (SIGIR '00), p. 41–48. ISBN 1-58113-226-3. Disponível em: <http://doi.acm.org/10.1145/345508.345545>.

JOACHIMS, T. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. (ICML '97), p. 143–151. ISBN 1-55860-486-3. Disponível em: <http://dl.acm.org/citation-.cfm?id=645526.657278>.

JOACHIMS, T.; GRANKA, L.; PAN, B.; HEMBROOKE, H.; RADLINSKI, F.; GAY, G. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 25, n. 2, abr. 2007. ISSN 1046-8188. Disponível em: <http://doi.acm.org/10.1145/1229179.1229181>.

KOLL, M. B. Weird: An approach to concept-based information retrieval. *SIGIR Forum*, ACM, New York, NY, USA, v. 13, n. 4, p. 32–50, abr. 1979. ISSN 0163-5840. Disponível em: <http://doi.acm.org/10.1145/1095366.1095368>.

LAGUN, D.; SUD, A.; WHITE, R. W.; BAILEY, P.; BUSCHER, G. Explicit feedback in local search tasks. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2013. (SIGIR '13), p. 1065–1068. ISBN 978-1-4503-2034-4. Disponível em: <http://doi.acm.org/10.1145/2484028.2484123>.

LASHKARI, A.; MAHDAVI, F.; GHOMI, V. A boolean model in information retrieval for search engines. In: *Information Management and Engineering, 2009. ICIME '09. International Conference on*. [S.l.: s.n.], 2009. p. 385–389.

LESK, M. Why digital libraries? *United Kingdom Office for Library and Information Networking, UKOLN*, 1995. Disponível em: <http://www.lesk.com/mlesk/follett/follett-.html>.

LIU, X.; YU, Y.; GUO, C.; SUN, Y. Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2014. (CIKM '14), p. 121–130. ISBN 978-1-4503-2598-1. Disponível em: <http://doi.acm.org/10.1145/2661829.2661965>.

LV, Y.; ZHAI, C. Adaptive relevance feedback in information retrieval. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 255–264. ISBN 978-1-60558-512-3. Disponível em: <http://doi.acm.org/10.1145/1645953.1645988>.

LV, Y.; ZHAI, C. Positional relevance model for pseudo-relevance feedback. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2010. (SIGIR '10), p. 579–586. ISBN 978-1-4503-0153-4. Disponível em: <http://doi.acm.org/10.1145/1835449.1835546>.

LYNCH, C. A. Institutional repositories: Essential infrastructure for scholarship in the digital age. *ARL: A Bimonthly Report*, n. 226, p. 1–7, 2003. Disponível em: <http://www.arl.org/resources/pubs/br/br226/br226ir.shtml>.

MA, Y.; LIN, H. A Multiple Relevance Feedback Strategy with Positive and Negative Models. *PLOS ONE*, PUBLIC LIBRARY SCIENCE, 1160 BATTERY STREET, STE 100, SAN FRANCISCO, CA 94111 USA, 9, n. 8, AUG 19 2014. ISSN 1932-6203.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2009. ISBN 0521865719, 9780521865715.

MAO, W.; CHU, W. W. Free-text medical document retrieval via phrase-based vector space model. In: *Proceedings of the AMIA Symposium (2002)*. [S.l.: s.n.], 2002. p. 489–493.

MARON, M. E.; KUHNS, J. L. On relevance, probabilistic indexing and information retrieval. *J. ACM*, ACM, New York, NY, USA, v. 7, n. 3, p. 216–244, jul. 1960. ISSN 0004-5411. Disponível em: <http://doi.acm.org/10.1145/321033.321035>.

MARTINS, A.; NUNES, M. B.; RODRIGUES, E. Repositórios de informação e ambientes de aprendizagem: Criação de espaços virtuais para a promoção da literacia e da responsabilidade social. *Rede de Bibliotecas Escolares Newsletter*, n. 3, 2008.

NIST. *TREC: Text REtrieval Conference*. 2015. Website (`http://trec.nist.gov/`). The TREC Conference series is co-sponsored by the National Institute of Standards and Technology (NIST) Information Technology Laboratory's (ITL) Retrieval Group of the Information Access Division (IAD).

PARAPAR, J.; PRESEDO-QUINDIMIL, M. A.; BARREIRO, A. Score distributions for Pseudo Relevance Feedback. *INFORMATION SCIENCES*, ELSEVIER SCIENCE INC, 360 PARK AVE SOUTH, NEW YORK, NY 10010-1710 USA, 273, p. 171–181, JUL 20 2014. ISSN 0020-0255.

PASS, G.; CHOWDHURY, A.; TORGESON, C. A picture of search. In: *Proceedings of the 1st International Conference on Scalable Information Systems*. New York, NY, USA: ACM, 2006. (InfoScale '06). ISBN 1-59593-428-6. Disponível em: <http://doi.acm.org/10.1145/1146847.1146848>.

POLYVYANYY, A.; KUROPKA, D. *A quantitative evaluation of the enhanced topic-based vector space model*. [S.l.]: Universität Potsdam, 2009.

RADLINSKI, F.; JOACHIMS, T. Query chains: Learning to rank from implicit feedback. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York, NY, USA: ACM, 2005. (KDD '05), p. 239–248. ISBN 1-59593-135-X. Disponível em: <http://doi.acm.org/10.1145/1081870.1081899>.

RADLINSKI, F.; KURUP, M.; JOACHIMS, T. How does clickthrough data reflect retrieval quality? In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2008. (CIKM '08), p. 43–52. ISBN 978-1-59593-991-3. Disponível em: <http://doi.acm.org/10.1145/1458082.1458092>.

RAMAN, K.; JOACHIMS, T. Machine learning and knowledge discovery in databases: European conference, ecml pkdd 2013, prague, czech republic, september 23-27, 2013, proceedings, part ii. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. cap. Learning Socially Optimal Information Systems from Egoistic Users, p. 128–144. ISBN 978-3-642-40991-2. Disponível em: <http://dx.doi.org/10.1007/978-3-642-40991-2_9>.

RAMAN, K.; JOACHIMS, T.; SHIVASWAMY, P. Structured learning of two-level dynamic rankings. In: *Conference on Information and Knowledge Management (CIKM)*. [S.l.: s.n.], 2011.

RAMAN, K.; SHIVASWAMY, P.; JOACHIMS, T. Learning to diversify from implicit feedback. In: *WSDM Workshop on Diversity in Document Retrieval*. [S.l.: s.n.], 2012.

ROBERTSON, S. E.; JONES, K. S. Relevance weighting of search terms. *Journal of the American Society for Information Science*, Wiley Subscription Services, Inc., A Wiley Company, v. 27, n. 3, p. 129–146, 1976. ISSN 1097-4571. Disponível em: <http://dx.doi.org/10.1002/asi.4630270302>.

ROCCHIO, J. J. Relevance feedback in information retrieval. In: SALTON, G. (Ed.). *The SMART Retrieval System–Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice Hall, Englewood Cliffs, 1971. cap. 14.

RUTHVEN, I.; LALMAS, M. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, Cambridge University Press, New York, NY, USA, v. 18, n. 2, p. 95–145, jun. 2003. ISSN 0269-8889. Disponível em: <http://dx.doi.org/10.1017/S0269888903000638>.

SALTON, G. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1971.

SALTON, G. Mathematics and information retrieval. *Journal of Documentation*, v. 35, n. 1, p. 1–29, 1979.

SALTON, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0-201-12227-8.

SALTON, G.; BUCKLEY, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, v. 41, p. 288–297, 1990.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, nov. 1975. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/361219.361220>.

SARACEVIC, T.; KANTOR, P.; CHAMIS, A. Y.; TRIVISON, D. A study in information seeking and retrieving. i. background and methodology. *Journal of the American Society for Information Science*, v. 39, n. 3, p. 161–176, 1988.

SAYÃO, L. Bibliotecas digitais e suas utopias. *PontodeAcesso*, v. 2, n. 2, 2008. ISSN ISSN 1981-6766.

SCHWARTZ, C. Digital libraries: an overview. *The Journal of Academic Librarianship*, v. 26, n. 6, p. 385–393, 2000. ISSN 0099-1333.

SINGH, J.; RAJPAL, N. Study on efficacy of relevance feedback for content based image retrieval. In: *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*. [S.l.: s.n.], 2015. p. 19–23.

SINGHAL, A. Modern information retrieval: a brief overview. *BULLETIN OF THE IEEE COMPUTER SOCIETY TECHNICAL COMMITTEE ON DATA ENGINEERING*, v. 24, p. 2001, 2001.

TSATSARONIS, G.; PANAGIOTOPOULOU, V. A generalized vector space model for text retrieval based on semantic relatedness. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (EACL '09), p. 70–78. Disponível em: <http://dl.acm.org/citation.cfm?id=1609179.1609188>.

TURNEY, P. D.; PANTEL, P. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 37, n. 1, p. 141–188, jan. 2010. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=1861751.1861756>.

WAN, G. G.; LIU, Z. Content-based information retrieval and digital libraries. *Information Technology and Libraries*, v. 27, n. 1, p. 41–47, 2008.

WANG, C.; SHEN, Y.; YANG, H.; GUO, M. Web information systems engineering – wise 2013: 14th international conference, nanjing, china, october 13-15, 2013, proceedings, part i. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. cap. Improving Rocchio Algorithm for Updating User Profile in Recommender Systems, p. 162–174. ISBN 978-3-642-41230-1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-41230-1_14>.

WARTIK, S. Information retrieval. In: FRAKES, W. B.; BAEZA-YATES, R. (Ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992. cap. Boolean Operations, p. 264–292. ISBN 0-13-463837-9. Disponível em: <http://dl.acm.org/citation.cfm?id=129687.129699>.

WHITE, R. W.; RUTHVEN, I.; JOSE, J. M. Advances in information retrieval: 24th bcs-irsg european colloquium on ir research glasgow, uk, march 25–27, 2002 proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. cap. The Use of Implicit Evidence for Relevance Feedback in Web Retrieval, p. 93–109. ISBN 978-3-540-45886-9. Disponível em: <http://dx.doi.org/10.1007/3-540-45886-7_7>.

WIBOWO, A.; HANDOJO, A.; HALIM, A. Application of topic based vector space model with wordnet. In: *Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on*. [S.l.: s.n.], 2011. v. 1, p. 133–136.

WONG, S. K. M.; RAGHAVAN, V. V. Vector space model of information retrieval: A reevaluation. In: *Proceedings of the 7th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Swinton, UK, UK: British Computer Society, 1984. (SIGIR '84), p. 167–185. ISBN 0-521-26865-6. Disponível em: <http://dl.acm.org/citation.cfm?id=636805.636816>.

WONG, S. K. M.; ZIARKO, W.; WONG, P. C. N. Generalized vector spaces model in information retrieval. In: *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1985. (SIGIR '85), p. 18–25. ISBN 0-89791-159-8. Disponível em: <http://doi.acm.org/10.1145/253495.253506>.

WU, H.; SALTON, G. The estimation of term relevance weights using relevance feedback. *Journal of Documentation*, v. 37, n. 4, p. 194–214, 1981. Disponível em: <http://dx.doi.org/10.1108/eb026717>.

YE, Z.; HUANG, J. X. A simple term frequency transformation model for effective pseudo relevance feedback. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. New York, NY, USA: ACM, 2014. (SIGIR '14), p. 323–332. ISBN 978-1-4503-2257-7. Disponível em: <http://doi.acm.org/10.1145/2600428.2609636>.

ZHAI, C. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, Now Publishers Inc., Hanover, MA, USA, v. 2, n. 3, p. 137–213, mar. 2008. ISSN 1554-0669. Disponível em: <http://dx.doi.org/10.1561/1500000008>.

ZHANG, J.; KAMPS, J. A search log-based approach to evaluation. In: LALMAS, M.; JOSE, J.; RAUBER, A.; SEBASTIANI, F.; FROMMHOLZ, I. (Ed.). *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, 2010, (Lecture

Notes in Computer Science, v. 6273). p. 248–260. ISBN 978-3-642-15463-8. Disponível em: <http://dx.doi.org/10.1007/978-3-642-15464-5_26>.

ZHANG, S.-L.; DONG, J.-T.; LIU, L.-L. A relevance feedback algorithm combining bayesian and fsrm. *The Open Cybernetics & Systemics Journal*, v. 9, p. 491–495, 2015.