Uma implementação de *feedback* da relevância utilizando o algoritmo Rocchio

Caroline Tomasini, André Prisco, Eduardo N. Borges

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brasil

{caroline.tomasini, andre.prisco, eduardoborges}@furg.br

Abstract. *This paper describes an information retrieval tool that implements the vector space model and the Rocchio algorithm. Once a user has viewed the result of a query, he or she can point out which of the returned items are relevant. The system will optimize the query according to the user relevance feedback.*

Resumo. *Este artigo descreve uma ferramenta de recuperação de informações que implementa o modelo espacial vetorial e o algoritmo Rocchio. Depois que um usuário tenha visualizado o resultado de uma consulta, ele pode apontar quais dos itens retornados são relevantes. O sistema otimizará a consulta de acordo com o feedback de relevância do usuário.*

1. Introdução

Sistemas de recuperação de informações têm como objetivo encontrar, em grandes coleções, documentos de natureza pouco ou não estruturada que satisfaçam uma necessidade de informação [Baeza-Yates e Ribeiro Neto 1999]. A necessidade de informação do usuário geralmente é mapeada para uma consulta em linguagem natural ou através de palavras-chave. Entretanto, essas palavras podem ter múltiplos significados. Por exemplo, a consulta pelo termo *graça* poderia recuperar documentos com os seguintes segmentos de texto: "fiéis agradecem a graça recebida" e "o produto saiu de graça". Essa propriedade, denominada polissemia, reduz a qualidade dos sistemas de recuperação de informação porque diminui a precisão do resultado, recuperando documentos que não fazem parte do interesse do usuário.

A sinonímia é outra propriedade que limita a qualidade dos sistemas de recuperação de informações porque diminui a abrangência do resultado. Por exemplo, a consulta pelo termo *carro* não seria capaz de recuperar documentos que contivessem apenas os termos *veículo* ou *automóvel*. O usuário teria que refinar a consulta diversas vezes até satisfazer sua necessidade de informação.

Para solucionar os problemas apresentados, foram propostos sistemas baseados no *feedback* da relevância do usuário [Manning et al. 2008]. A Figura 1 apresenta um exemplo do funcionamento de um sistema de recuperação de imagens deste tipo. O usuário realiza uma consulta pelo termo "*mouse*". O sistema retorna um conjunto inicial de documentos (a). O usuário marca alguns documentos retornados como relevantes (b). O sistema recalcula a representação da necessidade de informação com base no *feedback* dado pelo usuário. Por fim, é exibido um conjunto revisto de resultados recuperados (c).

(a) (b) (c)

Figura 1. Exemplo de recuperação de informações com base no *feedback* da relevância. (a) Resultado original. (b) Seleção de relevantes. (c) Resultado final.

Este artigo apresenta uma ferramenta de recuperação de informações que implementa o modelo espacial vetorial e o algoritmo Rocchio, descritos nas próximas seções. Ela foi desenvolvida para apoiar o aprendizado na disciplina de Sistemas de Informações Avançados oferecida pelo Centro de Ciências Computacionais da FURG.

2. Modelo espacial vetorial

No modelo espacial vetorial [Baeza-Yates e Ribeiro Neto 1999], cada documento d é representado por um vetor com t dimensões, uma para cada termo do vocabulário de toda a coleção. O peso w_i de cada dimensão i é calculado a partir da frequência dos termos e tem a função de quantificar a relevância de cada termo para as consultas e para os documentos. A relevância do documento em relação a uma consulta q é dada por uma função de similaridade baseada no cosseno do ângulo formado pelos dois vetores, conforme a Equação 1 [Salton e Bucckley 1988].

$$\mathrm{sim}(\mathrm{d},\mathrm{q}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{t}} \mathrm{w_{id}}\, \mathrm{w_{iq}}}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{t}} \mathrm{w_{id}^2}}\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{t}} \mathrm{w_{iq}^2}}} \tag{1}$$

Se uma coleção possui N documentos e df_i é a quantidade de documentos que possuem o termo t_i, então os pesos w_{id} são calculados através da métrica $tf\ x\ idf$ conforme a Equação 2, onde $freq_{id}$ é a frequência do termo i no documento d e $\max(freq_{td})$ é a máxima frequência de qualquer termo t presente no mesmo documento. Para as consultas, essa frequência normalizada ainda pode ser suavizada através de um fator de amortecimento. Perceba que quanto mais frequente é um termo num documento, maior o peso nessa dimensão. Entretanto, a frequência inversa idf reduz o peso de termos comuns na coleção, ou seja, daqueles que aparecem em muitos documentos.

$$w_{id} = tf_{id}\ idf_i = \frac{freq_{id}}{\max(freq_{td})} log\frac{N}{df_i} \tag{2}$$

3. Algoritmo Rocchio

Rocchio (1971) é um algoritmo que incorpora as informações de *feedback* da relevância no modelo espacial vetorial, melhorando o resultado final das consultas. O algoritmo tem como objetivo encontrar um vetor consulta modificado $\overrightarrow{q_m}$ que maximiza a similaridade com os documentos marcados como relevantes pelo o usuário e que minimiza a semelhança com documentos não relevantes. A Equação 3 define o vetor

consulta modificado, onde $\overrightarrow{q_0}$ é a consulta original, $\overrightarrow{d_j}$ é a representação vetorial de um documento que pode pertencer ao conjunto de relevantes para o usuário D_r ou ao de não relevantes D_{nr}. A equação soma à consulta original a representação centroide dos documentos relevantes e diminui o centroide dos não relevantes. Os pesos α, β e γ ponderam a importância do *feedback* em relação à consulta original.

$$\overrightarrow{q_m} = \alpha\, \overrightarrow{q_0} + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \overrightarrow{d_j} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \overrightarrow{d_j} \qquad (3)$$

4. Implementação da ferramenta

A Figura 2 apresenta a arquitetura da ferramenta desenvolvida. O usuário realiza uma consulta em linguagem natural que é entregue ao componente *Pré-processamento*. Este componente é responsável pelas etapas de decomposição e normalização do texto, remoção de palavras irrelevantes, composição do vocabulário de termos (dimensões do modelo espacial vetorial) e pela atribuição de pesos. Ele é utilizado para gerar a representação vetorial das consultas do usuário $\overrightarrow{q_0}$ e de todos os documentos da coleção. O componente *Busca* pesquisa na coleção pelos documentos que contenham os termos de $\overrightarrow{q_0}$. A seguir, o componente *Ranking* calcula a similaridade entre $\overrightarrow{q_0}$ e cada documento recuperado. Estes documentos são ordenados de acordo com a similaridade calculada e entregues ao usuário. Analisando o *ranking* de documentos, o usuário seleciona um conjunto de resultados relevantes para a consulta. Este *feedback* é usado para retroalimentar o sistema, permitindo ao componente *Rocchio* calcular uma nova representação da necessidade de informação do usuário $\overrightarrow{q_m}$. O novo vetor de consulta modificado é submetido ao componente *Busca* e o restante do processo é repetido.

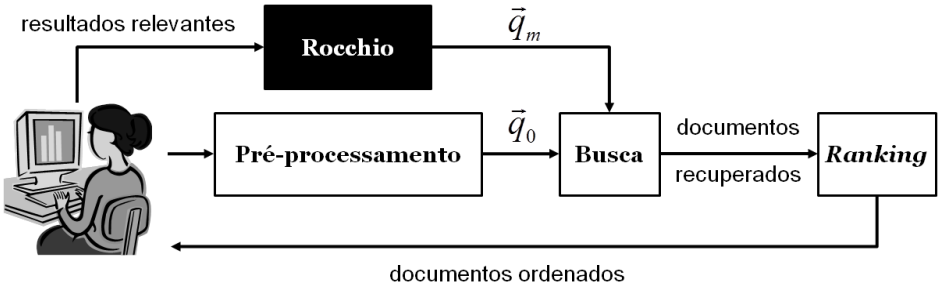

Figura 2. Arquitetura da ferramenta desenvolvida.

Foi utilizada a linguagem de programação PHP para implementar os componentes apresentados. Conforme sugeridos por Manning et al. (2008), foram utilizados os pesos $\alpha = 1$, $\beta = 0.75$ e $\gamma = 0.15$. Entretanto, a ferramenta permite a alteração desses e de outros parâmetros como o diretório contendo a coleção.

A Figura 3 apresenta uma consulta e uma coleção de documentos pré-processados. A Figura 4 mostra o resultado intermediário em que o *ranking* foi gerado apenas pela aplicação do modelo espacial vetorial e o *ranking* final, gerado após a aplicação do *feedback* da relevância. Perceba que o documento *jogo.txt*, que é do interesse do usuário, ficou na última posição com apenas 11% de similaridade em relação à consulta. Depois do *feedback*, o mesmo documento passou para a primeira posição do *ranking* junto de *xadrez.txt* que também é relevante, melhorando significativamente a qualidade do resultado. A precisão média [Manning et al. 2008] passou de 75 para 100%.

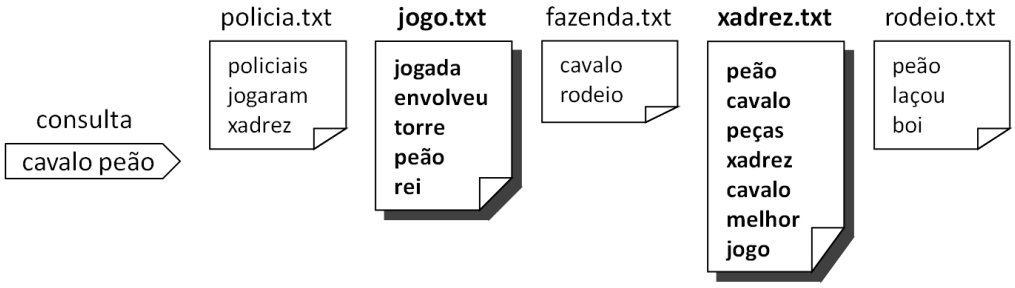

Figura 3. Exemplo destacando os documentos relevantes para a consulta do usuário.

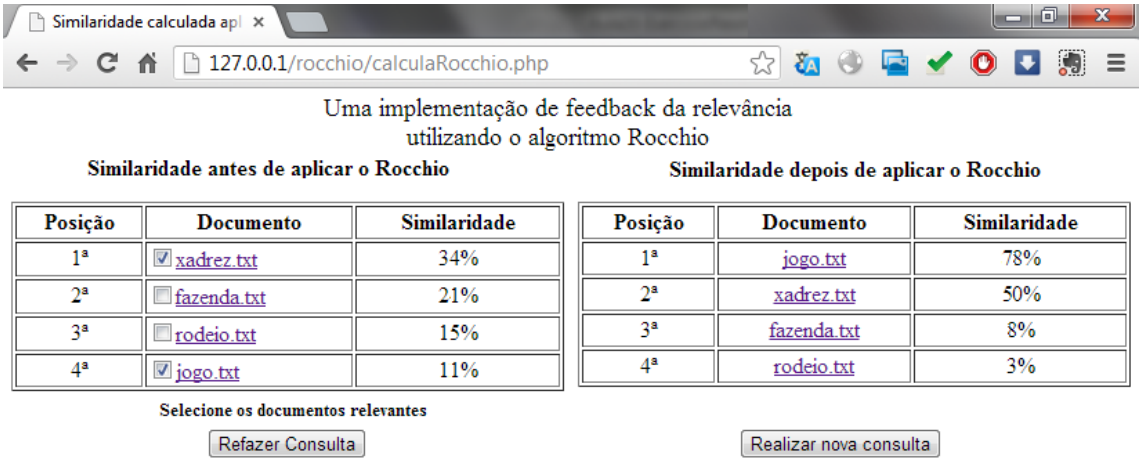

Figura 4. *Ranking* original (à esquerda) e recalculado a partir do *feedback* da relevância do usuário (á direita).

A ferramenta também exibe os passos intermediários da contagem de frequências e dos cálculos da métrica $tf\ x\ idf$, da similaridade e algoritmo Rocchio. É possível visualizar os vetores de cada documento e das consultas $\overrightarrow{q_0}$ e $\overrightarrow{q_m}$. Entretanto, por restrições de espaço, essas informações não puderam ser apresentadas neste artigo.

5. Conclusão

Para ser utilizada em sala de aula, ainda são necessárias melhorias na interface gráfica, as quais serão realizadas até o início do próximo semestre letivo. Com a ferramenta finalizada, espera-se que os estudantes possam aprender os conceitos apresentados neste artigo com mais facilidade.

Referências

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

Manning, C. D.; Raghavan, P. and Schutze H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G. (Ed.), The Smart Retrieval System – Experiments in Automatic Document Processing, Prentice Hall., p. 313 -323.

Salton, G. and Bucckley, C (1988). Term-weighting approaches in Automatic Retrieval. In *Information Processing & Management*, 24(5), p. 513–523.