|  |
| --- |
| brasao |
| Demetrius Costa Rapello  Sistema de recomendação para suporte a produção de matérias relacionadas no portal G1 |
| Dissertação de mestrado  Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio.  Orientador: Prof. Marco Antônio Casanova |
|  |
| Rio de Janeiro, 19 de novembro de 2011. |

|  |
| --- |
| brasao |
| Demetrius Costa Rapello  Sistema de recomendação para suporte a produção de matérias relacionadas no portal G1 |
|  |
| Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada. |
| Prof. Marco Antônio Casanova  Orientador e Presidente  Departamento de informática - PUC-Rio  Prof. Alessandro Fabricio Garcia  Departamento de informática - PUC-Rio  Prof. Gustavo Robichez de Carvalho  Departamento de informática - PUC-Rio  Prof.ª. Simone Diniz Junqueira Barbosa  Departamento de informática - PUC-Rio  Rio de Janeiro, 19 de novembro de 2011. |

|  |  |
| --- | --- |
|  | Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador. |
|  | Demetrius Costa Rapello  Graduou-se em Ciência da Computação pela Universidade de Augosto Motta (UNISUAM) em Dezembro de 2000. Tem experiência na área de Ciência da Computação, com ênfase em Desenvolvimento de Software. Tem trabalhado em analise de sistemas desde 1997. |
|  | Ficha Catalográfica |
|  | Demetrius Costa Rapello  Sistema de recomendação para suporte a produção de matérias relacionadas no portal G1 / Demetrius Costa Rapello; orientador: Marco Antônio Casanova. - Rio de Janeiro: PUC-Rio, Departamento de Informática, 2011.  v., 100 f.: il. ; 29,7 cm  Dissertação de Mestrado - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática  Referencias bibliográficas incluídas.  Sistemas de Recomendação; Recuperação da Informação; Extração de Entidades Nomeadas; |

A Deus, aos meus pais, a minha esposa e aos meus filhos.

|  |
| --- |
| Agradecimentos |
| A Deus pelo apoio incondicional, conforto de coração e paz de espírito que tanto foram importantes para alcançar este objetivo.  Aos meus pais, Tarciso Rapello e Maria da Conceição costa Rapello, pelo apoio, educação, carinho e dedicação.  A minha esposa Gabriela barbosa da silva Rapello, pelo apoio constante, carinho e compreensão.  Aos meus filhos Gabriel e Matheus que entenderam a ausência do pais com carinho e compreensão.  Ao meu orientador, prof. Marco Antônio Casanova, por sua dedicação, motivação, ensinamentos e orientação.  À Globo.com, pelo financiamento e auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.  Aos professores da Comissão examinadora.  A todos os amigos e familiares que de alguma forma contribuíram para a realização deste trabalho. |

|  |
| --- |
| Resumo |
| Demetrius Costa Rapello. **Sistema de recomendação para suporte a produção de artigos do portal G1.** Rio de Janeiro, 2011. 100p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.  bla |
| Palavras-chave  Sistemas de Recomendação; Recuperação da Informação; Extração de Entidades Nomeadas. |

|  |
| --- |
| Abstract |
| Demetrius Costa Rapello. **Recomended System to support News article production on G1.** Rio de Janeiro, 2011. 101p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.  BLa. |
| Keywords  Recommended Systems; Information Retriaval; Named Entity Extraction. |

1 Introdução 13

1.1 Objetivo da Dissertação 14

1.2 Organização da Dissertação 14

2 Trabalhos Relacionados 15

2.1 Sistemas de recomendação 15

2.1.1 Modelos de sistemas de recomendação 16

2.1.2 Baseados em conteúdo 16

2.1.3 Baseado em filtragem colaborativa 16

2.1.4 Baseado em nichos demográficos 16

2.1.5 Baseado em conhecimento 17

2.1.6 Baseado na comunidade 17

2.1.7 Sistemas híbridos de recomendação 17

2.2 Projeto Pure 17

2.2.1 Interface para registro de artigos de interesse do usuário 18

2.2.2 Treinamento do modelo probabilístico baseado nos artigos de interesse do usuário 19

2.2.3 Etapa de Seleção de palavras e atribuição de peso 19

2.2.4 Etapa de Geração do modelo probabilístico 19

2.2.5 Recuperação diária de novos artigos da base PubMed. 20

2.2.6 Recomendação dos artigos 20

3 GRNews 21

3.1 Arquitetura do sistema e sumário 21

3.2 Corpus 21

3.3 Extração de padrões 21

3.4 Seleção de candidatos 21

3.5 Decisões de Projeto 21

4 Experimento 23

5 Conclusões e trabalhos futuros 24

6 Bibliografia 25

|  |
| --- |
| Lista de imagens |
|  |

Figure 1 - Função de Recomendação 20

Figure 2 - Zcore 20

|  |
| --- |
| Lista de tabelas |

# Introdução

Com o aumento considerável de informações de notícias na web, a recomendação tem se tornado extramente importante para os usuários. Estudos indicam que o volume de páginas indexadas nos principais buscadores aproximasse da casa dos 20 bilhões [13]. Desta forma podemos perceber que os usuários não tem como absorver todo esse conteúdo produzido sem que um agente organize em contextos específicos. O sistemas de recomendação são os agentes responsaveis por organizar estas informações baseados em lógicas previamente definidas. Quando estamos navegando na web em um site de compra de jogos e nos deparamos com uma lista dos dez jogos mais procurados pelos usuários ou quando estamos numa locadora virtual e verificamos uma lista com os filmes mais vistos, estamos na verdade recebendo recomendações para facilitar nossas decisões.

Sistemas de recomendação tem sido largamente utilizados para atender a esta necessidade, pois em geral, estes sistemas são construídos com o intuito de predizer as necessidades do usuário de modo que se possa oferecer conteúdos com uma maior probabilidade de aproveitamento.

Para aprender a predizer as necessidades dos usuários, os sistemas de recomendação necessitam de acesso ao feedback dos usuários quando estes estão consumindo um determinado conteúdo. Este feedback pode ser dado de forma explicita quando um usuário informa ao sistema quais são as suas preferencias, ou de forma implícita quando o sistema descobre as preferencias do usuário baseado em suas interações.

## Objetivo da Dissertação

Diante deste cenário, propomos um sistema de recomendação que aprenda as necessidades dos usuários baseado no fluxo de navegação entre as matérias para sugerir conteúdos relacionados aos novos artigos produzidos.

Com a necessidade de recomendação de conteúdo relacionado, os editores do portal techtudo passaram a ter uma segunda função. Atualmente, além da criação da nota ou artigo sobre um determinado tema, eles ainda precisam relacionar outras notas e ou artigos no sistema de recomendação existente. Deste modo o processo de produção de conteúdo se torna lento e a totalidade do tempo do editor, que deveria ser para confecção mais elaborada da matéria, passar a ser destinado a um trabalho manual de mineração de texto para realização de relacionamentos de conteúdos.

Para gerar recomendações de conteúdos para uma matéria, o editor precisa utilizar um fluxo de trabalho que passa pelas seguintes etapas:

1. Identificação de palavras/frases chaves relevantes ao tema.
2. Consulta de conteúdos anteriores na base de conhecimento utilizando as chaves.
3. Reconhecimento da similaridade entre os conteúdos e o tema.
4. Relacionamento manual dos conteúdos.

A expectativa é:

* completar.

## Organização da Dissertação

O restante deste documento está organizado em cinco capítulos, da seguinte forma:

O **Capítulo 2** apresenta o estado da arte em sistemas de recomendação e as ferramentas e *frameworks* existentes comparados a ferramenta proposta.

O **Capítulo 3** descreve a ferramenta desenvolvida para apoiar a recomendação de conteúdo na produção de matérias do G1.

O **Capítulo 5** apresenta os resultados obitdos com a experimentação da ferramenta contra um corpus real.

O **Capítulo 6** apresenta as conclusões e trabalhos futuros.

# Trabalhos Relacionados

Existem três conceitos importantes que serão abordados durante essa dissertação: sistemas de recomendação, técnicas de recuperação da informação e a técnica de extração de entidades nomeadas. Foi realizada uma pesquisa na literatura para cada um desses tópicos. Na última seção será apresentada a diferença do que já foi realizado para o trabalho que está sendo desenvolvido.

## Sistemas de recomendação

Sistemas de recomendação são softwares especializados em apresentar “opções” para serem usadas por seus usuários [10,11]. Esses sistemas auxiliam pessoas que não possuem muita experiência ou competência para pesquisar dados acerca de um determinado assunto.

Em linhas gerais, os sistemas de recomendação procuram oferecer as melhores opções de resposta para as necessidades dos usuários num processo de tomada de decisão.

Os sistemas de recomendação podem oferecer sugestões em diferentes domínios como por exemplo: que item comprar em uma loja virtual, que artigo ler em um site de notícias ou mesmo que restaurante visitar em uma cidade.

No sistema de recomendação do **IMDB** (**Internet Movie Database**), quando o usuário seleciona um filme do catálogo para ler a respeito, o sistema apresenta também uma lista de sugestões de filmes relacionados para o usuário.

Assim como o **IMDB**, o site de compras **Amazon** apresenta para cada produto selecionado pelo usuário, uma lista de recomendações de outros produtos que possam servir para o usuário dentro de uma mesma compra.

Sistemas de recomendação são usualmente personalizados com base em características individuais ou coletivas, todavia, os sistemas de recomendação não personalizados também tem o seu espaço. Sistemas de recomendação não personalizados são mais simples de serem implementados e geralmente são usados para a recomendações mais gerais como por exemplo, a lista dos “10 mais” de um determinado assunto ou tema.

Sistemas de recomendação personalizados tentam antecipar as necessidades do usuário analisando informações do seu perfil e levando em consideração as restrições de domínio para então recomendar sugestões. Ainda no site de compras da **Amazon**, uma vez realizada uma compra, as informações são armazenadas no perfil do usuário de modo que para as compras futuras o usuário receba como recomendação, não apenas itens relacionados pela categoria em que se encaixam, mas também por que estão relacionados com a última compra feita por ele.

### Modelos de sistemas de recomendação

Sistemas de recomendação variam de acordo com a técnica utilizada para descobrir as preferencias do usuário. Entre os sistemas de recomendação mais comuns temos:

### Baseados em conteúdo

Sistemas de recomendação baseados em conteúdo tentam recomendar opções que são similares a algum item que o usuário já selecionou no passado. Por exemplo, se um usuário leu um artigo sobre política em um portal de notícias, este portal poderia sugerir, em consultas futuras, artigos relacionados a política.

### Baseado em filtragem colaborativa

Sistemas de recomendação baseados em filtragem colaborativa, tentam recomendar ao usuário opções que foram utilizadas por usuários com os mesmos interesses. Por exemplo, se um grupo de usuários selecionam o produto X e em seguida selecionam o produto Y, o sistema de recomendação entende que usuários que acessam o produto X também acessam o produto Y e passa a recomendar o produto Y toda vez que algum novo usuário se interessar pelo produto X.

### Baseado em nichos demográficos

Este tipo de sistema de recomendação baseia-se no perfil demográfico do usuário para recomendar opções. Usuários de um determinado estado recebem recomendações diferentes de usuários de outros estados. Os nichos demográficos podem ser: idade, sexo, língua etc.

### Baseado em conhecimento

Sistemas de recomendação baseados em conhecimento tentam recomendar ao usuário opções baseando-se no conhecimento específico do domínio do sistema. Este modelo também é reconhecido como um modelo baseado em caso onde o problema é a analise das necessidades do usuário e a solução é o conjunto de opções a serem recomendadas.

Sistemas de recomendação com esta técnica tendem a trabalhar melhor que os outros no início, porém se não são acompanhados de componente de aprendizagem se tornam ineficientes.

Neste sistema o principio de recomendação está baseado na similaridade de uma solução para um dado problema.

### Baseado na comunidade

Este tipo de sistema de recomendação procura basear-se nas relações que o usuário possui com sua rede de amigos para realizar as recomendações. Neste modelo acredita-se que a recomendação de pessoas ligadas ao usuário tendem a ser mais efetivas. Esta abordagem tem se tornado bastante atrativa tendo em vista o grande crescimento das redes de relacionamento.

### Sistemas híbridos de recomendação

Sistemas de recomendação híbridos procuram utilizar técnicas de outros sistemas de recomendação de forma a complementar as deficiências entre as técnicas.

## Projeto Pure

O projeto PURE [12] é um sistema para recomendação de artigos médicos que utiliza o princípio de recomendação baseado na filtragem de conteúdo. Em linhas gerais, o sistema executa uma classificação dos artigos preferidos do usuário para então recomendar outros artigos para leitura.

O sistema PURE pode ser entendido pelo fluxo de operações a seguir:

1. O usuário acessa o sistema para informar os artigos do seu interesse. Estes artigos são armazenados na base de dados do PURE.
2. Um sistema de aprendizado de máquina é aplicado para extrair as preferências do usuário com base nos seus artigos de interesse.
3. O sistema PURE consulta a base de dados PubMed para baixar os novos artigos publicados.
4. Os artigos baixados da PubMed são ordenados com base no modelo treinado com as preferências do usuário.
5. Os artigos são então apresentados para o usuário.

O propósito do PURE desenvolvido por TAKASHI YONEYA e HIROSHI MAMITSUKA, é oferecer uma interface que facilite o processo de busca de artigos relevantes pela comunidade científica na base de dados PubMed. A base de dados PubMed mantém um grande acervo de artigos de biologia e medicina com um volume diário de atualização da ordem de centenas artigos. Diante deste cenário, os autores perceberam que um sistema de recomendação baseado em filtragem de conteúdo poderia auxiliar a comunidade científica no trabalho de consulta de novos artigos. Os módulos a seguir fazem parte do sistema PURE.

### Interface para registro de artigos de interesse do usuário

Para utilizar o sistema, o usuário precisa registrar os seus artigos de interesse na base de dados do PURE. Para esta atividade o usuário acessa a interface web do sistema e seleciona os artigos de sua preferência em uma listagem. Os artigos selecionados são então gravados no perfil do usuário e armazenados no banco de dados do PURE.

O usuário tem a permissão de adicionar novos arquivos e alterar sua lista de interesse.

### Treinamento do modelo probabilístico baseado nos artigos de interesse do usuário

Os artigos de interesse do usuário são utilizados para a concepção de um modelo probabilístico que procura identificar as preferências do usuário para novos artigos. Esse modelo é dividido em duas etapas:

### Etapa de Seleção de palavras e atribuição de peso

Nesta etapa o sistema trata os artigos do PubMed como um vetor de palavras ordenadas por peso. Estas palavras são obtidas a partir da eliminação de palavras irrelevantes para o sistema classificadas como stopwords.

As stopwords são obtidas por duas estratégias distintas, a primeira consiste na geração do DF e TF-IDF das palavras oriundas de uma porção aleatória de artigos da base de dados do PubMed. As palavras com alto DF (número de documentos onde a palavra aparece) ou com baixo TF-IDF (métrica que informa quanto a palavra é relevante para o documento) são consideradas stopwords. A segunda estratégia consiste em considerar como stopwords as palavras que respeitam as seguintes regras: 1) palavras com menos de 3 letras; 2) palavras sem caracteres alfabéticos; 3) palavras que aparecem no **Journal of Business Research** de Jan. 2005 a 2006.

Após a eliminação das stopwords é dado um peso para cada palavra restante do documento. Este peso é obtido pela verificação da distribuição da palavra pelo documento (TF).

### Etapa de Geração do modelo probabilístico

As palavras selecionadas na etapa anterior são usadas para treinar um classificador probabilístico que será usado para gerar uma métrica de recomendação para os novos artigos. A função utilizada para computar o grau de recomendação de um artigo é dada pela fórmula abaixo:

Sendo d um artigo, z a variável correspondente ao cluster, s um campo existente na estrutura do artigo exe., título e w uma palavra do artigo temos:

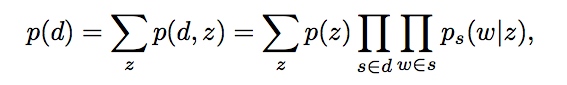


Figure 1 - Função de Recomendação

Em seguida os autores treinam os parâmetros de probabilidade p(z) e ps(w|z) a partir dos artigos preferidos do usuário utilizando o algoritmo de Maximização de Expectativa (EM).

### Recuperação diária de novos artigos da base PubMed.

O sistema PURE diariamente executa uma operação de recuperação dos novos artigos publicados na base de dados do PubMed. Os novos artigos são armazenados na base de dados do sistema para serem classificados de acordo com as preferências de cada usuário.

### Recomendação dos artigos

Para cada artigo recuperado, são extraídas palavras que são usadas como base para geração do critério de ordenação que é dado de acordo com a função de recomendação definida anteriormente. Como forma de ajustar possíveis desvios do algoritmo de recomendação, os autores apresentam um score adicional para cada artigo. O Z-score é obtido através do agrupamento dos artigos em conjuntos de artigos com o mesmo número de palavras. Deste modo é separado o mínimo e o máximo graus de recomendação do grupo onde o Z-score do artigo é dado pela formula:



Figure 2 - Zcore

Os artigos com o maior Z-score são então recomendados para o usuário.

# GRNews

ESCREVER

## Arquitetura do sistema e sumário

ESCREVER

## Corpus

ESCREVER

## Extração de padrões

ESCREVER

## Seleção de candidatos

ESCREVER

## Decisões de Projeto

As principais tecnologias descritas abaixo, foram utilizadas para projetar, desenvolver e suportar a ferramenta pois são tecnologias de código aberto, gratuitas, do conhecimento técnico do autor e por se adequarem as práticas utilizadas no contexto da empresa de onde será aplicado o experimento.

***Ruby on Rails*:** *framework* de código aberto escrito na linguagem de programação [*Ruby*](http://pt.wikipedia.org/wiki/Ruby_%28linguagem_de_programa%C3%A7%C3%A3o%29) que promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a banco de dados (*database-driven* web sites), uma vez que é possível criar aplicações com base em estruturas pré-definidas. As aplicações criadas utilizando o framework *Rails* são desenvolvidas com base no padrão de projeto [MVC](http://pt.wikipedia.org/wiki/MVC) (*Model-View-Controller*).

***Aptana Studio*** – ambiente integrado de desenvolvimento de código aberto e gratuito desenvolvido em Java e suporta as linguagens CSS, HTML, pode ser configurado para suportar *Ruby on Rails* através do plug-in *Aptana RedRails*. É baseado no Eclipse, programa similar que por sua vez desenvolve [linguagens de programação](http://pt.wikipedia.org/wiki/Linguagens_de_programa%C3%A7%C3%A3o). É distribuído em múltiplas plataformas ou seja, pode ser utilizado no [Windows](http://pt.wikipedia.org/wiki/Windows), [Linux](http://pt.wikipedia.org/wiki/Linux) e [Mac OS X](http://pt.wikipedia.org/wiki/Mac_OS_X).

**CVS:** CVS ou *Concurrent Version System* (Sistema de Versões Concorrentes) é um sistema de controle de versão de código aberto e gratuito que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os logs de quem e quando manipulou os arquivos. É especialmente útil para se controlar versões de um software durante seu desenvolvimento, ou para composição colaborativa de um documento.

***PostgreSQL*:** é um sistema de gerenciamento de banco de dados (SGBD) que utiliza a linguagem SQL Como interface. Desenvolvido como projeto de código aberto.

***Capistrano*:** ferramenta de código aberto e gratuita que permite rodar scripts em múltiplos servidores. Seu principal uso é para realização de *deploy* de aplicações WEB. Com ele é possível automatizar o processo de subida para produção de uma aplicação permitindo a execução de várias tarefas em múltiplos servidores.

***Selenium Remote Control* (RC):** é uma ferramenta que permite automatizar testes em aplicações web. É dividido em duas partes:

* Um servidor que automaticamente inicia e encerra o navegador web, e atua como um proxy HTTP para solicitações web provindas desses navegadores.
* Biblioteca cliente para montar e executar os scripts de teste. Disponível para diferentes linguagens de programação.

# Experimento

ESCREVER

# Conclusões e trabalhos futuros

ESCREVER

# Bibliografia

1. Agile Manifesto web site**;** Disponível em: <http://agilemanifesto.org/>. Acesso em: 09 fev. 2010.
2. 1 - [BOFENG ZHANG, XIN XU, JINSHU SU - 2007]. **An Ensemble Method for Multi-class and Multi-label Text Categorization:** National University of Defense Technology.
3. 2 - [Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze - 2008] **Introduction to Information Retrieval**, Cambridge University Press.
4. 3 - [Min-Ling Zhang, Jose M. Penã and Victor Robles – 2009]. **Feature Selection for Multi-Label Naive Bayes Classification**: College of Computer and Infor- mation Engineering, Hohai University, Nanjing 210098, China;
5. 4 - [Arni Darliani Asy’arie, Adi Wahyu Pribadi – 2009]. **Automatic News Articles Classification in Indonesian Language by Using Naive Bayes Classifier Method**: Universitas Islam Negeri Syarif Hidayatullah Jakarta, Universitas Pancasila Jl. Srengseng Sawah, Jagakarsa Jakarta.
6. 5 - [Chin-Yew Lin, Eduard Hovy – 1997]. **Identifying Topics by Position,** Information Sciences Institute of the University of Southern California
7. 6 - [Dipanjan Das, Andre F.T. Martins - 2007] **A Survey on Automatic Text Summarization,** Language Technologies Institute Carnegie Mellon University.
8. 7 - [Jacob Perkins - 2010]. Python Text Processing with NLTK 2.0 Cookbook.
9. 8 -[LUIZ CLÁUDIO GOMES MAIA – 2008] Uso de Sintagmas Nominais na Classificação Automática de Documentos Eletrônicos. UFMG.
10. 9 -[Bruno Magalhães Nogueira - 2009]. Avaliação de métodos não-supervisionados de seleção de atributos para Mineração de Textos. ICMC-USP.
11. 10 -[Francesco Ricci, Lior Rokach, Bracha Shapira - 2010]. **Introduction to Recommender Systems Handbook**.
12. 11 -[Bracha Shapira, Lior Rokach - 2010]. **Recommender Systems and Search Engines – Two sides of the same Coin!?**. Department of Information Systems Engineering, Ben-Gurion University.
13. 12 -[TAKASHI YONEYA , HIROSHI MAMITSUKA1 - 2007] **PURE: A Pubmed Article Rcommendation System Based on Content-Based Filtering.** Bioinformatics Center, Kyoto University, Gokasho Uji, Japan and Discovery Research Laboratories, Kirin Pharma, Miyahara, Takasaki, Japan
14. 13 – [WORLD WIDE WEB SIZE] **Daily Estimated Size**, 2011 Disponivel em: <http://www.worldwidewebsize.com/>. Acesso em: 08 Jul 2011.
15. 14 – [MINIWATTS MARKETING GROUP] **Internet World Stats**, 2011. Disponivel em: <http://www.internetworldstats.com/stats.htm>. Acesso em: 19 Jan 2011.
16. 15 – [Greg Linden, Brent Smith, Jeremy York] **Amazon.com Recommendations Item-to-Item Collaborative Filtering**, 2003 - IEEE INTERNET COMPUTING.
17. 16 - [James Davidson, Benjamin Liebald, Junning Liu] - **The YouTube Video Recommendation System**, 2010, Google Inc
18. 17 - [Ying Huang] - **An Intelligent adaptative news filtering system**, 2001, University of Science & Technology of China.
19. 18 - [Abhinandan Das, Mayur Datar Google,] **Google News Personalization: Scalable Online Collaborative Filtering**, 2007, Google Inc
20. 19 – [Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, Yi Chang] **Learning to model relatedness for news recommendation,** 2011, Yahoo! Labs, Sunnyvale, CA, USA.
21. 20 – [Yin Yang, Panagiotis Ipeirotis, Wisam Dakka, Dimitris Papadias, Nilesh Bansal, Nick Koudas] **Query by Document,** 2009**,** Computer Science University of Toronto koudas@cs.toronto.edu

# APENDICE A

1. **ESCREVER**