

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Coordenação de Pós-Graduação em Ciência da Computação

## DBFIRE: Recuperação de Documentos Relacionados a Consultas a Banco de Dados

Vladimir Soares Catão

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da  
Computação da Universidade Federal de Campina Grande – Campus Campina  
Grande como parte dos requisitos necessários para obtenção do grau de Doutor  
em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Sistemas de Informação e Bancos de Dados

Ulrich Schiel e Marcus Costa Sampaio

(Orientadores)

Campina Grande, Paraíba, Brasil

Novembro/2014

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

C357d

Catão, Vladimir Soares.

DBFIRE: recuperação de documentos relacionados a consulta de bancos de dados / Vladimir Soares Catão. – Campina Grande, 2014.

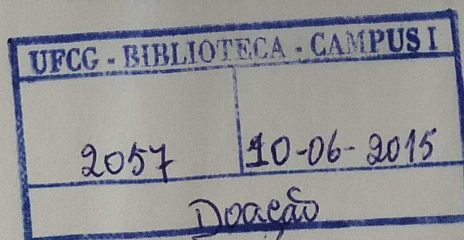
132 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2014.

"Orientação: Prof. Dr. Ulrich Schiel, Prof. Dr. Marcus Costa Sampaio".  
Referências.

1. Recuperação de Informação. 2. Integração SGBD / SRI. I. Schiel, Ulrich. II. Sampaio, Marcus Costa. III. Título.

CDU 004.65(043)

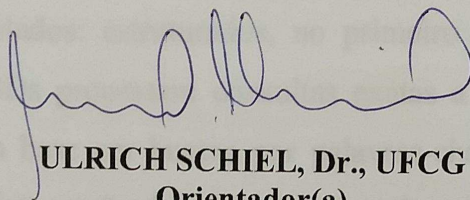




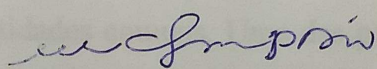
**"DBFIRE: RECUPERAÇÃO DE DOCUMENTOS RELACIONADOS A CONSULTAS A BANCO DE DADOS"**

**VLADIMIR SOARES CATÃO**

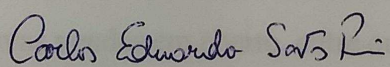
**TESE APROVADA EM 21/11/2014**



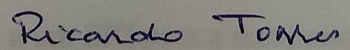
**ULRICH SCHIEL, Dr., UFCG**  
**Orientador(a)**



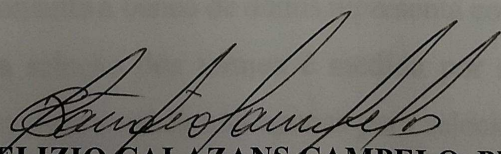
**MARCUS COSTA SAMPAIO, Dr., UECE**  
**Orientador(a)**



**CARLOS EDUARDO SANTOS PIRES, Dr., UFCG**  
**Examinador(a)**



**RICARDO DA SILVA TORRES, D.Sc., UNICAMP**  
**Examinador(a)**



**CLAUDIO ELIZIO CALAZANS CAMPELO, PhD., UFCG**  
**Examinador(a)**

**EDLENO SILVA DE MOURA, D.Sc., UFAM**  
**Examinador(a)**

**CAMPINA GRANDE - PB**

## Resumo

Bancos de dados e documentos são comumente mantidos em separado nas organizações, controlados por Sistemas Gerenciadores de Bancos de Dados (SGBDs) e Sistemas de Recuperação de Informação (SRIs), respectivamente. Essa separação tem ligação com a natureza dos dados manipulados: estruturados, no primeiro caso; não estruturados, no segundo. Enquanto os SGBDs processam consultas exatas a bancos de dados, os SRIs recuperam documentos com base em buscas por palavras-chave, que são inerentemente imprecisas. Apesar disso, a integração desses sistemas pode resultar em grandes ganhos ao usuário, uma vez que, numa mesma organização, bancos de dados e documentos frequentemente se referem a entidades comuns. Uma das possibilidades de integração é a recuperação de documentos associados a uma dada consulta a banco de dados. Por exemplo, considerando a consulta "Quais os clientes com contratos acima de X reais?", como recuperar documentos que possam estar associados a esta consulta, como os próprios contratos desses clientes, propostas de novas vendas em aberto, entre outros documentos? A solução proposta nesta tese baseia-se numa abordagem especial de expansão de busca para a recuperação de documentos: um conjunto inicial de palavras-chave é expandido com termos potencialmente úteis contidos no resultado de uma consulta a banco de dados; o conjunto de palavras-chave resultante é então enviado a um SRI para a recuperação dos documentos de interesse para a consulta. Propõe-se ainda uma nova forma de ordenação dos termos para expansão: partindo do pressuposto de que uma consulta a banco de dados representa com exatidão a necessidade de informação do usuário, a seleção dos termos é medida por sua difusão ao longo do resultado da consulta. Essa medida é usada não apenas para selecionar os melhores termos, mas também para estabelecer seus pesos relativos na expansão. Para validar o método proposto, foram realizados experimentos em dois domínios distintos, com resultados evidenciando melhorias significativas em termos da recuperação de documentos relacionados às consultas na comparação com outros modelos destacados na literatura.

# Conteúdo

<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>1</b>
1.1 RECUPERAÇÃO DE DOCUMENTOS UTILIZANDO BANCOS DE DADOS COMO FONTES DE TERMOS PARA A BUSCA .....	3
1.2 DBFIRE: <i>DATA</i> BASES <i>FOR</i> <i>INFORMATION</i> <i>RETRIEVAL</i> .....	4
1.3 ORGANIZAÇÃO DA TESE .....	6
<b>CAPÍTULO 2 - MÉTODOS PARA INTEGRAÇÃO ENTRE SGBDS E SRIS.....</b>	<b>8</b>
2.1 INTRODUÇÃO .....	8
2.2 CONSULTA A BDs ATRAVÉS DE PALAVRAS-CHAVE .....	9
2.3 RECUPERAÇÃO DE DOCUMENTOS ATRAVÉS DE LINGUAGENS DE CONSULTA ESTRUTURADAS.....	11
2.4 BUSCA POR DOCUMENTOS XML .....	11
2.5 EXTRAINDO PALAVRAS-CHAVE PARA RECUPERAÇÃO DE DOCUMENTOS A PARTIR DE CONSULTAS A BDs .....	12
2.5.1 <i>SCORE</i> .....	12
2.5.2 <i>SEMEX</i> .....	15
2.6 DISCUSSÃO.....	18
<b>CAPÍTULO 3 - EXPANSÃO AUTOMÁTICA DE BUSCAS.....</b>	<b>20</b>
3.1 INTRODUÇÃO .....	20
3.2 EXPANSÃO AUTOMÁTICA DE BUSCAS (AQE).....	21
3.3 FUNCIONAMENTO DE MÉTODOS DE PRF .....	23
3.3.1 <i>PRF baseada em distribuições de termos</i> .....	24
3.3.2 <i>PRF baseada em modelos de linguagem (language models)</i> .....	25
3.4 DISCUSSÃO.....	26
<b>CAPÍTULO 4 - O MÉTODO DBFIRE .....</b>	<b>28</b>
4.1 INTRODUÇÃO .....	28
4.2 DBFIRE EM DETALHES .....	29
4.2.1 <i>Arquitetura</i> .....	29
4.2.2 <i>Ordenação dos termos em DBFIRE</i> .....	31
4.2.3 <i>Quantos termos devem ser usados na expansão? Quantas tuplas devem ser analisadas?</i> .....	32
4.2.4 <i>Qual o peso dos termos enviados ao SRI?</i> .....	34
4.3 UM EXEMPLO.....	37
<b>CAPÍTULO 5 - AVALIAÇÃO VIA COLEÇÕES DE TESTE .....</b>	<b>40</b>
5.1 INTRODUÇÃO .....	40
5.2 JULGAMENTOS INCOMPLETOS .....	41
5.3 MÉTRICAS DE AVALIAÇÃO .....	42
5.3.1 <i>MAP – Mean Average Precision</i> .....	43
5.3.2 <i>Bpref – Binary Preference</i> .....	44
5.3.3 <i>Relacionando Precisão e Revocação</i> .....	45
5.4 TESTES DE SIGNIFICÂNCIA.....	46

<b>CAPÍTULO 6 - AMBIENTE DE TESTES.....</b>	<b>48</b>
6.1 COLEÇÃO INEX-DC.....	49
6.1.1 <i>Macro-Cenário 1: Induzindo um BD a partir dos Documentos XML</i> .....	50
6.1.2 <i>Macro-Cenário 2: Busca Estruturada nos Documentos XML</i> .....	53
6.2 COLEÇÃO INEX-LOD.....	54
6.3 DISCUSSÃO.....	57
<b>CAPÍTULO 7 - AVALIAÇÃO DE DBFIRE.....</b>	<b>59</b>
7.1 INTRODUÇÃO .....	59
7.1.1 <i>Comparando DBFIRE com um baseline</i> .....	60
7.1.2 <i>Comparando DBFIRE com outros métodos de integração SGBD-SRI</i> .....	60
7.1.3 <i>Comparando a ordenação de termos de DBFIRE com outros métodos de ordenação</i> .....	61
7.1.4 <i>Comparando com os sistemas participantes dos workshops INEX</i> .....	61
7.2 EFEITOS DOS PARÂMETROS <i>K, N E B</i> PARA A EXPANSÃO.....	62
7.3 COMPARATIVOS NO MACRO-CENÁRIO 1.....	65
7.3.1 <i>Comparativos no ambiente padrão de testes</i> .....	65
7.3.2 <i>Inserindo arquivos irrelevantes pertencentes a outro domínio na coleção de documentos</i> .....	68
7.3.3 <i>Usando menos palavras-chave do usuário na busca expandida</i> .....	70
7.3.4 <i>Excluindo campos do resultado da consulta</i> .....	72
7.4 COMPARATIVOS NO MACRO-CENÁRIO 2.....	74
7.4.1 <i>Comparativos no ambiente padrão de testes</i> .....	75
7.4.2 <i>Expansão com os literais da consulta</i> .....	77
7.4.3 <i>Removendo campos dos resultados das consultas</i> .....	79
7.5 COMPARATIVOS NO MACRO-CENÁRIO 3.....	80
7.5.1 <i>Comparativo nas configurações padrão</i> .....	81
7.5.2 <i>Efetuando a expansão com os literais da consulta SQL</i> .....	83
7.5.3 <i>Removendo campos do resultado da consulta SPARQL</i> .....	85
7.6 COMENTANDO OS RESULTADOS OBTIDOS .....	87
<b>CAPÍTULO 8 - CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>90</b>
8.1 INCORPORANDO OUTROS FATORES AOS PESOS DOS TERMOS PARA EXPANSÃO .....	90
8.2 ATRIBUIÇÃO DE PESOS DIFERENTES PARA CADA COMPONENTE DA FÓRMULA DE ORDENAÇÃO DE TERMOS.....	91
8.3 SELECIONANDO AS TUPLAS COM MAIOR POTENCIAL PARA EXPANSÃO .....	91
8.4 INTEGRAÇÃO SGBDs/SRI A PARTIR DE CONSULTAS A BDs VIA PALAVRAS-CHAVE .....	92
8.5 VARIANDO O PESO MÁXIMO DOS TERMOS PARA EXPANSÃO DEPENDENDO DA CONSULTA .....	92
8.6 EXPLORAR RECURSOS DA PRÓPRIA LINGUAGEM DE CONSULTA ESTRUTURADA.....	93
8.7 RECONHECIMENTO DE FRASES NOMINAIS .....	93
8.8 NOVOS EXPERIMENTOS.....	93
<b>ANEXO A - CONSULTAS SQL PARA A COLEÇÃO INEX-DC .....</b>	<b>95</b>
<b>ANEXO B - CONSULTAS SPARQL PARA A COLEÇÃO INEX-LOD .....</b>	<b>103</b>
<b>ANEXO C – VERSÃO SQL DAS CONSULTAS SPARQL DA COLEÇÃO INEX-LOD.....</b>	<b>115</b>
<b>REFERÊNCIAS.....</b>	<b>125</b>

# Lista de Símbolos

AQE – *Automatic Query Expansion*  
BD – *Banco de Dados*  
DBFIRE – *DataBases For Information Retrieval*  
DFR – *Divergence From Randomness*  
DTD – *Document Type Definition*  
IMDB – *Internet Movie DataBase*  
INEX-DC – *INitiative for the Evaluation of XML Retrieval – coleção Data Centric*  
INEX-LOD – *INitiative for the Evaluation of XML Retrieval – coleção Linked Data*  
IQE – *Interactive Query Expansion*  
KLD – *Kullback-Lieber Distance*  
RDF – *Resource Description Framework*  
RI – *Recuperação de Informação*  
RM – *Relevance Model*  
SCORE – *Symbiotic Context-Oriented Information REtrieval*  
SEMEX – *SEMantic EXplorer*  
SGBD – *Sistema Gerenciador de Banco de Dados*  
SPARQL – *SPARQL Protocol and RDF Query Language*  
SRI – *Sistema de Recuperação de Informação*  
XML – *eXtensible Markup Language*

# Lista de Figuras

Figura 1.1 - Como a informação é vista por SGBDs e SRIs .....	2
Figura 1.2 - Funcionamento de DBFIRE .....	5
Figura 2.1 - Diagrama ER Person-Movie para o IMDB.....	13
Figura 2.2 - Consulta SQL para os filmes de Martin Scorsese.....	14
Figura 2.3 - Grafo para a consulta sobre filmes de Martin Scorsese .....	17
Figura 2.4 – Versão simplificada para o grafo da Figura 2.3.....	17
Figura 4.1 - Diagrama de caso de uso para a nova funcionalidade do SGBD .....	30
Figura 4.2 - Diagrama de sequência para a consulta com documentos relacionados .....	30
Figura 4.3 - Probabilidades de DBFIRE .....	31
Figura 4.4 - Ocorrências de <i>john, smith</i> e <i>john smith</i> no Google Books (1900-2008) .....	36
Figura 4.5 - Consulta SQL para os filmes de Martin Scorsese.....	37
Figura 5.1 - Precisão interpolada versus revocação para a Tabela 5.2 .....	46
Figura 6.1 - Fragmento da DTD sobre filmes .....	49
Figura 6.2 - Fragmento da DTD sobre pessoas .....	50
Figura 6.3 - Diagrama ER relativo aos fragmentos das DTDs INEX-DC.....	51
Figura 6.4 - Arquivo XML para o filme "The Departed" .....	51
Figura 6.5 – BD após carga dos dados do filme "The Departed" .....	52
Figura 6.6 - Tópico 2011104 da coleção INEX-DC .....	52
Figura 6.7 - Consulta SQL para o tópico 2011104 .....	52
Figura 6.8 - Tópico 2011107 .....	53
Figura 6.9 - Consulta SQL para o tópico 2011107 .....	53
Figura 6.10 - Tupla referente ao exemplo na Figura 6.4 .....	54
Figura 6.11 - Tópico 2013311 da coleção INEX-LOD .....	55
Figura 6.12 - Versão SPARQL para o tópico 2013311.....	55
Figura 6.13 - Conversão de triplas SPARQL em tuplas DBFIRE .....	56
Figura 6.14 - Versão SQL para o tópico 2013311.....	57
Figura 7.1 - Efeito do parâmetro $k$ .....	63
Figura 7.2 - Efeito do parâmetro $n$ .....	64
Figura 7.3 - Efeito do parâmetro $\beta$ .....	64
Figura 7.4 - Precisão x Revocação no macro-cenário 1 (ambiente padrão de testes) .....	66
Figura 7.5 - Precisão x Revocação (sistemas INEX) .....	67
Figura 7.6 - Precisão x Revocação (incluindo documentos TREC).....	69
Figura 7.7 - Precisão x Revocação com literais da consulta como palavras-chave .....	71
Figura 7.8 - Precisão x Revocação com literais da consulta como palavras-chave versus sistemas INEX.....	72
Figura 7.9 - Precisão x Revocação removendo campos do resultado da consulta .....	73
Figura 7.10 - Precisão x Revocação frente a sistemas INEX (remoção de campos) .....	74
Figura 7.11 - Precisão x Revocação (comparativo padrão no macro-cenário 2) .....	75



Figura 7.12 - Precisão x Revocação (comparativo padrão no macro-cenário 2 frente aos sistemas INEX) .....	76
Figura 7.13 - Precisão x Revocação (expansão com os literais) .....	78
Figura 7.14 - Precisão x Revocação no macro-cenário 2 (expansão com os literais) .....	78
Figura 7.15 - Precisão x Revocação macro-cenário 2 (removendo campos) .....	79
Figura 7.16 - Precisão x Revocação macro-cenário 2 com sistemas INEX (removendo campos) ....	80
Figura 7.17 – Precisão x Revocação (comparativo padrão) .....	82
Figura 7.18 – Precisão x Revocação (comparativo padrão com sistemas INEX) .....	83
Figura 7.19 – Precisão x Revocação (expandido a partir dos literais) .....	84
Figura 7.20 – Precisão x Revocação (comparativo padrão com sistemas INEX) .....	84
Figura 7.21 – Precisão x Revocação (removendo campos da consulta) .....	86
Figura 7.22 – Precisão x Revocação (comparativo padrão com sistemas INEX) .....	87

# Lista de Tabelas

Tabela 2.1 - Fragmento do resultado da consulta exemplo .....	14
Tabela 2.2 - 10 primeiros links da busca com SCORE .....	15
Tabela 2.3 – Resultado para a busca com SEMEX.....	18
Tabela 4.1 – Sobrecarga adicionada à busca para diversos valores de $k$ e $n$ .....	34
Tabela 4.2 - Fragmentos da consulta sobre filmes de Martin Scorsese .....	38
Tabela 4.3 - Probabilidades e peso para os termos destacados na Tabela 4.2 .....	38
Tabela 4.4 - 10 primeiros links da busca expandida.....	39
Tabela 5.1 - Precisão/revocação para uma lista fictícia de documentos .....	46
Tabela 5.2 - Precisão interpolada para os dados da Tabela 5.1.....	46
Tabela 7.1 - Razão Máximo/Minimo para a métrica MAP devido à variação de $k$ .....	63
Tabela 7.2 - Razão Máximo/Minimo para a métrica MAP devido à variação de $n$ .....	64
Tabela 7.3 - Razão Máximo/Minimo para a métrica MAP devido à variação de $\beta$ .....	65
Tabela 7.4 – Comparativo no ambiente padrão de testes .....	66
Tabela 7.5 - Comparativo com sistemas INEX (ambiente padrão de testes).....	66
Tabela 7.6 – Comparativo com inclusão de documentos da coleção TREC.....	69
Tabela 7.7 – Comparativo usando literais da consulta como palavras-chave para expansão .....	71
Tabela 7.8 - Comparativo com sistemas INEX usando literais da consulta como palavras-chave para expansão.....	71
Tabela 7.9 – Comparativo removendo campos do resultado da consulta .....	73
Tabela 7.10 - Comparativo com sistemas INEX removendo campos do resultado da consulta .....	73
Tabela 7.11 – Comparativo padrão no macro-cenário 2.....	75
Tabela 7.12 - Comparativo padrão com sistemas INEX.....	76
Tabela 7.13 – Comparativo macro-cenário 2 (expansão com os literais) .....	77
Tabela 7.14 - Comparativo com sistemas INEX no macro-cenário 2 (expansão com os literais) ....	78
Tabela 7.15 – Comparativo macro-cenário 2 (removendo campos).....	79
Tabela 7.16 - Comparativo macro-cenário 2 com sistemas INEX (removendo campos) .....	80
Tabela 7.17 – Resultados macro-cenário 3 (comparativo padrão) .....	82
Tabela 7.18 – Comparativo padrão com sistemas INEX).....	82
Tabela 7.19 – Resultados macro-cenário 3 (expandido a partir dos literais) .....	83
Tabela 7.20 – Comparativo com sistemas INEX expandido a partir dos literais.....	84
Tabela 7.21 – Resultados macro-cenário 3 (removendo campos da consulta).....	85
Tabela 7.22 – Comparativo com sistemas INEX expandido a partir dos literais.....	86
Tabela 7.23 – DBFIRE x RM (macro-cenário 1) .....	89
Tabela 7.24 – DBFIRE x RM (macro-cenário 3) .....	89

# Lista de Equações

Equação 2.1 - Função de ordenação de elementos de tupla de SCORE.....	13
Equação 3.1 - Ranking de similaridade entre documentos e buscas .....	21
Equação 3.2 - Similaridade entre busca/documento (versão expandida) .....	22
Equação 3.3 - Fórmula Rocchio.....	23
Equação 3.4 - Peso de um termo para o método KLD .....	25
Equação 3.5 - Peso de um termo para o método DFR .....	25
Equação 3.6 - Peso de um termo para o método RM .....	26
Equação 3.7 - Método RM (versão com <i>smoothing</i> ).....	26
Equação 4.1 - Probabilidade de $t$ na sequência $s$ .....	32
Equação 4.2 - Probabilidade de $t$ nos elementos das tuplas .....	32
Equação 4.3 - Peso do termo $t$ na ordenação.....	32
Equação 4.4 – Peso do termo $t$ na expansão.....	35
Equação 5.1- Cálculo de AP .....	43
Equação 5.2 - Cálculo de AP para $S_1$ e $S_2$ .....	43
Equação 5.3 - Cálculo de $B_{pref}$ .....	44
Equação 5.4 - Cálculo de $B_{pref}$ para $S_1$ e $S_2$ .....	44
Equação 5.5 - Precisão interpolada.....	45

# Capítulo 1 - Introdução

Bancos de dados e documentos são as principais fontes de informação da maioria das organizações, sejam elas empresas, instituições de ensino, órgãos governamentais ou não governamentais. No entanto, os sistemas que controlam os bancos de dados (Sistemas Gerenciadores de Bancos de Dados - SGBDs) e os sistemas que gerenciam documentos (Sistemas de Recuperação de Informação - SRIs) são normalmente mantidos em ambientes separados. O isolamento dos dois sistemas está relacionado com a natureza distinta da informação que cada um gerencia.

Bancos de dados (BDs) são exemplos de informação estruturada, a qual possui regras rígidas de construção, como as tabelas do modelo de dados relacional. A estrutura dos dados é conhecida, estando disponível através de esquemas de bancos de dados. Como exemplo, pode-se citar a clássica tríade clientes-produtos-pedidos. A forma de consulta é baseada em linguagens formais (SQL é a linguagem relacional padrão), nas quais a necessidade de informação é definida precisamente. Sendo assim, o resultado de uma consulta, dado por um conjunto de tuplas, é também exato, com todas as tuplas igualmente relevantes.

Documentos, por sua vez, são normalmente escritos em texto livre, sem nenhuma organização pré-definida: e-mails, manuais ou contratos são exemplos. Diferentemente das consultas com SGBDs, a recuperação de documentos com SRIs é baseada em buscas<sup>1</sup> por palavras-chave, as quais devem resumir a necessidade de informação do usuário a partir dos termos que ele *julga* serem os mais importantes.

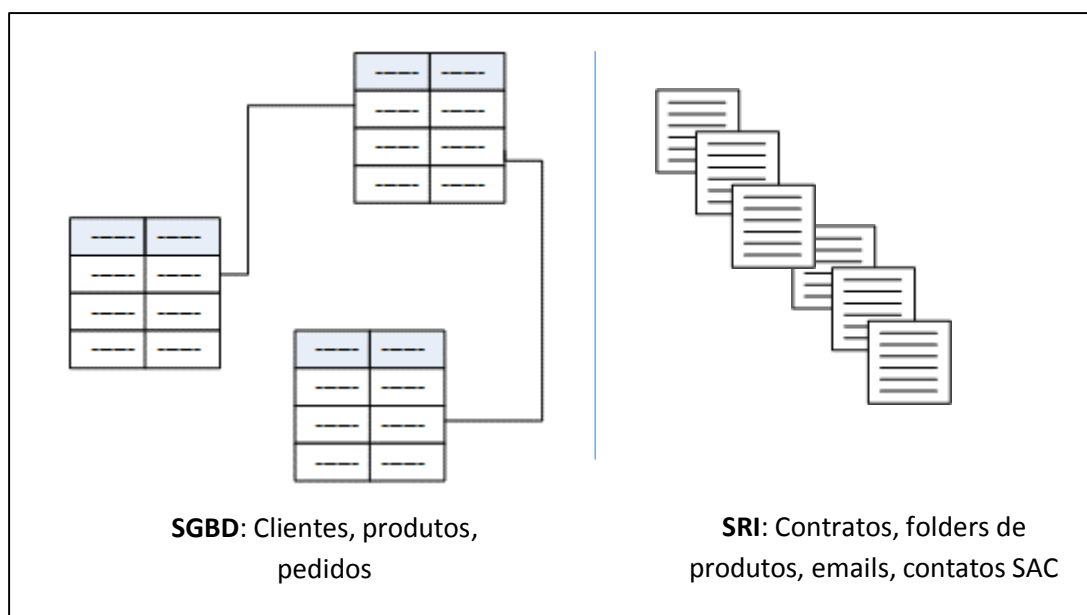
A lista de documentos recuperados é ordenada por critérios de relevância próprios do SRI, sendo que um fator importante de qualquer critério é a frequência das palavras-chave ao longo dos documentos recuperados ou mesmo ao longo da coleção completa de

---

<sup>1</sup> O termo “consulta” pode estar associando tanto a SGBDs como a SRIs; ao longo desta tese optou-se por usar o termo “consulta” exclusivamente para a recuperação de tuplas num SGBD, enquanto que para a recuperação de documentos num SRI se utilizará o termo “busca”.

documentos. No entanto, o fato de um documento conter uma determinada palavra-chave não implica que ele seja necessariamente relevante. Esse é o efeito conhecido por *incompatibilidade de termos* (*term mismatch* [35]): os termos usados para a escrita dos documentos podem ser diferentes daqueles que escolhemos para recuperá-los, ou podem até ser os mesmos, mas com sentidos diferentes. Esse efeito, aliado à inerente imprecisão na tradução da necessidade de informação para palavras-chave, fazem com que frequentemente apareçam documentos irrelevantes entre os documentos recuperados por um SRI.

De forma resumida, enquanto SGBDs tratam a consulta como uma tarefa de “casamento” (*matching*) através de predicados lógicos, os SRIs encaram as consultas como uma tarefa de “ordenação” (*ranking*) baseada em modelos estatísticos [96]. As diferentes naturezas dos dados manipulados por SGBDs e SRIs são ilustradas na Figura 1.1, motivando sua separação dentro das organizações.



**Figura 1.1 - Como a informação é vista por SGBDs e SRIs**

No entanto, é possível alterar esse cenário. Com o intuito de facilitar a integração entre os mundos dos BDs e dos documentos, alguns autores [26, 38, 96] sugerem paradigmas para uma plataforma única, que manipule dados dos dois mundos da forma mais transparente possível. Assim, é possível fazer buscas por palavras-chave a BDs [39, 42, 55, 56, 71], ou consultas a BDs induzidos a partir de coleções de documentos [15, 46, 50, 69]. A recuperação de documentos XML [5, 8, 52, 58] é outra linha de pesquisa que une as áreas de BD e RI.

Uma outra promissora vertente se volta para a utilização de informação presente em consultas a BDs com o intuito de recuperar documentos [57, 75]. Sua motivação passa pela



constatação de que é bastante provável que as mesmas entidades ou objetos de uma dada organização possuam informações disseminadas tanto entre seus bancos de dados quanto em suas coleções de documentos. É nesse contexto que esta tese se enquadra.

## **1.1 Recuperação de Documentos Utilizando Bancos de Dados como Fontes de Termos para a Busca**

Considere-se que o mundo estruturado esteja confinado aos BDs relacionais. Dessa forma, imaginemos que os SGBDs forneçam um serviço extra: além de apresentarem normalmente o resultado das consultas, possam também indicar documentos associados a essas consultas; esses documentos residiriam em repositórios gerenciados por SRIs. Adiantam-se alguns cenários em que esse recurso pode ser importante:

1. A partir de uma consulta sobre as conferências nas quais um determinado pesquisador publicou, recuperar os textos completos dos artigos publicados, ou mesmo os slides de suas apresentações;
2. Dada uma consulta indicando produtos que apresentaram algum defeito de fabricação, recuperar contatos de clientes realizados via SAC (Serviço de Atendimento ao Consumidor) relacionados com esses produtos ou com seus defeitos;
3. Considerando uma consulta por processos pendentes em uma dada organização jurídica (escritório de advocacia, tribunal, etc.), recuperar e-mails relativos aos processos, réus, ou partes interessadas;
4. Na área farmacêutica, partindo de uma consulta por medicamentos banidos, recuperar bulas de outros medicamentos com os quais tais medicamentos possam estar relacionados (compostos ou reações adversas semelhantes);
5. Considerando os filmes dirigidos por um determinado diretor, recuperar documentos contendo as resenhas ou sinopses desses filmes.

Uma estratégia possível para se chegar aos documentos de interesse seria extrair termos<sup>2</sup> presentes na consulta a banco de dados (seja no seu resultado, seja no seu corpo SQL), para então formular uma busca por palavras-chave a ser enviada a um SRI. Infelizmente, não é qualquer termo escolhido dessa forma que poderá de fato recuperar

---

<sup>2</sup> Apesar de “termos” e “palavras-chave” poderem ser usados indistintamente, preferimos usar “termos” quando estivermos no contexto SGBD e “palavras-chave” quando estivermos no contexto SRI.

documentos relevantes.

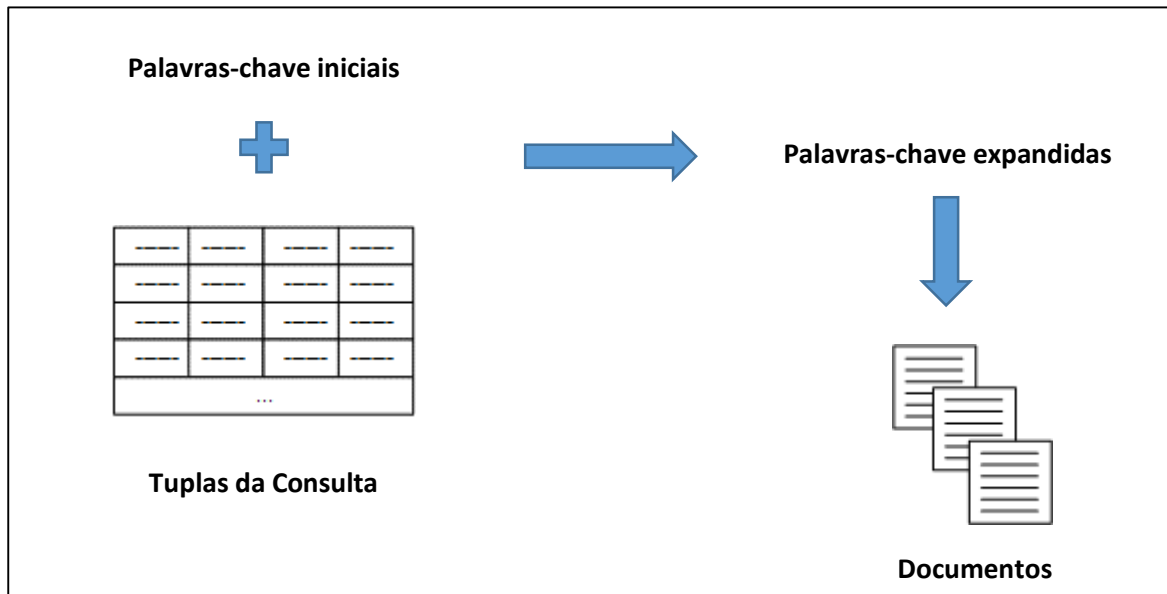
Nesta tese são avaliadas diversas abordagens que seguiram essa ideia geral (ver Capítulo 7), tendo-se verificado que em muitos casos os resultados podem ser até mesmo nocivos, isto é, piores do que com uma simples busca direta por palavras-chave fornecidas pelo próprio usuário. Ciente desse fato, a linha base (*baseline*) que norteou a solução proposta foi que qualquer modelo de recuperação de documentos associados a BDs deve produzir resultados no mínimo melhores do que os de uma busca direta por palavras-chave fornecidas pelo próprio usuário.

## **1.2 DBFIRE: *DataBases For Information Retrieval***

Considerem-se os métodos de expansão automática de buscas (AQE - *Automatic Query Expansion* [19]), disponíveis nativamente em diversos SRIs: partindo de uma descrição inicial da necessidade de informação (palavras-chave fornecidas pelo usuário) realiza-se uma primeira busca com o SRI. Os primeiros documentos retornados na busca são analisados e os termos com maior potencial para a expansão são selecionados.

Diversas heurísticas estão disponíveis para se estimar o potencial de um termo para expansão: é comum levar em conta sua distribuição na lista dos primeiros documentos retornados e/ou na coleção completa de documentos. Os termos com as melhores estimativas são concatenados aos termos originais do usuário, expandindo-os, constituindo assim uma nova submissão de palavras-chave ao SRI. Apenas o resultado da busca definitiva é apresentado ao usuário.

O método DBFIRE adapta esse mecanismo aplicando-o ao contexto de integração BDs-documentos conforme esquema ilustrado na Figura 1.2: considerando o resultado de uma consulta a banco de dados, e um conjunto de palavras-chave (que pode ser fornecido pelo usuário ou extraído a partir dos literais do corpo SQL da consulta), o método expande as palavras-chave iniciais com termos que potencialmente ofereçam melhoria no resultado da consulta.



**Figura 1.2 - Funcionamento de DBFIRE**

Além de adaptar a expansão de buscas à integração BDs-documentos, outra inovação do método está na forma de se chegar aos termos de expansão. A hipótese que se investigou foi a seguinte: quanto mais um determinado termo estiver difundido ao longo do resultado da consulta, mais útil para expansão ele deverá ser.

Para isso, considere-se que o resultado de uma consulta a BD é formada por um conjunto de tuplas, e cada tupla  $t_i$  é representada por uma sequência de  $n$  elementos (ou seja,  $t_i = \langle e_1, e_2, \dots, e_n \rangle$ ). Por sua vez, cada elemento de tupla  $e_j$ , possui um multi-conjunto com  $m$  termos (ou seja,  $e_j = \{w_1, w_2, \dots, w_m\}$ ).

Assim, para estimar a difusão de um termo, DBFIRE baseia-se em duas probabilidades: a probabilidade de que o termo simplesmente apareça no resultado da consulta (refletindo a proporção de sua frequência entre todos os termos de todas as tuplas), e a probabilidade de que o termo apareça em qualquer elemento das tuplas do resultado da consulta. Ou seja, considerando todos os elementos encontrados nas linhas e colunas do resultado da consulta ao BD, qual a probabilidade de se encontrar o termo candidato em qualquer desses elementos?

Perceba-se que a medida da difusão de um termo aqui proposta está ligada exclusivamente às suas ocorrências dentro do corpus de análise; no caso de métodos para expansão de buscas tradicionais isso equivaleria aos primeiros documentos relativos à primeira rodada de busca por palavras-chave ao SRI. Isso de certa forma contraria um paradigma forte na área de RI em geral e na de AQE mais especificamente, de que a utilidade

de um termo estaria ligada a sua forte presença no conjunto de análise *em conjunto* com a sua raridade ao longo de todos os documentos da coleção. A medida da raridade de um termo na coleção é popularmente conhecida como o componente IDF (*Inverse Document Frequency* [73]), sendo usada diretamente ou através de variações por diversos métodos seja de AQE, seja mesmo para a ordenação de documentos recuperados por um SRI.

Para validar o método DBFIRE, diversos cenários de teste foram montados em dois domínios diferentes. Em particular, comparou-se o método com um *baseline* formado por uma busca direta por palavras-chave em um SRI. Além disso, comparou-se o método DBFIRE com outros métodos de integração BDs-documentos, tendo-se realizado ainda uma comparação específica entre o método de ordenação de termos de DBFIRE com outros métodos largamente usados em SRIs. Em todos esses cenários DBFIRE apresentou resultados superiores com boa margem de diferença. Como produtos do trabalho, podemos citar dois artigos aceitos para publicação em conferências [24, 25] e outro submetido a periódico em processo de revisão final [23].

### 1.3 Organização da Tese

Os capítulos seguintes detalham as questões mencionadas ao longo desta Introdução. Assim, a tese está estruturada como segue:

O **Capítulo 2** ilustra a área de pesquisa em integração de informação estruturada e não estruturada. Mais especificamente, descrevem-se métodos para a integração entre SGBDs e SRIs. Detalhes das abordagens mais afinadas com esta tese são apresentados, ilustrando em particular seus pontos fracos.

O **Capítulo 3** é dedicado a métodos de expansão automática de buscas, detalhando seu funcionamento e indicando as bases que serviram de inspiração para esta tese.

O **Capítulo 4** apresenta o método DBFIRE propriamente dito, sua arquitetura e detalhes sobre seu funcionamento.

O **Capítulo 5** ilustra os conceitos mais importantes sobre avaliação em RI necessários para a compreensão da validação aplicada a DBFIRE.

Os **Capítulos 6 e 7** detalham os experimentos realizados ditos para validação do método. No **Capítulo 6** se explica a montagem do ambiente de testes, enquanto que no **Capítulo 7** mostram-se os resultados dos experimentos, incluindo as comparações com abordagens similares mencionadas nos Capítulos 2 e 3.

O **Capítulo 8** conclui o trabalho, com uma avaliação dos resultados obtidos, e um levantamento de pontos em aberto, incluindo oportunidades para sua evolução.



# Capítulo 2 - Métodos para Integração entre SGBDs e SRIs

## 2.1 Introdução

A despeito de suas diferentes naturezas, o volume mais e mais crescente de informações provenientes de BDs e documentos tem impelido a comunidade a propor alternativas para sua integração [3, 4, 26, 96]. As estratégias de integração são as mais diversas, desde sugestões para uma nova álgebra para consultas, até arquiteturas que permitam fornecer o melhor dos dois mundos: a otimização na execução de consultas dos BDs junto com a ordenação por relevância dos SRIs.

Uma dessas arquiteturas é aquela que vê BDs e documentos como parte de um “espaço de dados” único, o chamado *dataspace* [38, 63]. Esse espaço poderia ser consultado tanto através de palavras-chave como via linguagens estruturadas do tipo SQL. Uma vez que a consulta pode operar sobre ambientes distintos (SGBDs e SRIs), seus resultados podem ser também heterogêneos, mesclando tanto tuplas de BDs como documentos provenientes de SRIs; todos os itens de resposta seriam ordenados por relevância. Trabalhos voltados à área de *Enterprise Search* (recuperação de informação aplicada ao domínio empresarial ou organizacional) também vislumbram um ambiente semelhante [30].

Tal ambiente plenamente integrado ainda está distante, e com vários problemas a serem tratados [63]. Os pontos em aberto vão desde a manipulação integrada de esquemas diferentes, passando por questões de privacidade e segurança de dados, chegando até à interação do usuário com uma lista de resposta contendo itens tão diversos quanto documentos e tuplas. Além disso, como esses itens deveriam ser coerentemente ordenados?

Apesar desses problemas, vários trabalhos têm se voltado à solução das questões

mais básicas que devem ser endereçadas para a plena viabilidade da visão de *dataspaces*. Assim, considerando apenas as linguagens de consulta e suas estruturas de entrada e saída, engenhos de *dataspaces* lidam idealmente com quatro tipos de cenários [50]:

1. Consultas por palavras-chave sobre BDs.
2. Consultas por palavras-chave sobre documentos.
3. Consultas estruturadas (estilo SQL) sobre BDs.
4. Consultas estruturadas (estilo SQL) sobre documentos.

Os itens 2 e 3 já são nativos em SRIs e SGBDs, respectivamente. O item 1 diz respeito à tarefa de recuperar as tuplas de um BD dado um conjunto de palavras-chave, sem a necessidade de o usuário ter conhecimento prévio do esquema do BD. O item 4 está ligado à recuperação de tuplas de BD inferidas de coleções de documentos; neste caso, esquemas e dados são extraídos diretamente dos documentos, alimentando BDs para posterior consulta. Os itens 1 e 4 serão abordados em mais detalhes nas próximas seções.

Apesar de não estar ligada diretamente ao ambiente de *dataspaces*, a consulta a documentos XML é outro nicho de pesquisa que alia tecnologias tanto da área de BD como de RI. Isso porque marcações XML podem conter tanto texto livre como conteúdo estruturado, inclusive com reconhecimento de tipos de dados. Esse nicho de pesquisa também será abordado neste capítulo.

Por fim, a área mais afeita a esta tese parte de uma consulta a BD e devolve documentos relativos à consulta [57, 65, 75]. Os documentos são recuperados indiretamente através de buscas por palavras-chave. As palavras-chave, por sua vez, são obtidas a partir de heurísticas que determinam os termos mais importantes presentes na consulta (seja no corpo SQL, seja nas tuplas resultantes); ao final, esses termos é que são submetidos ao SRI.

Dado este panorama, as seções seguintes ilustram as diferentes abordagens citadas, identificando pontos fortes e questões ainda em aberto.

## **2.2 Consulta a BDs através de palavras-chave**

Consultas por palavras-chave<sup>3</sup> em BDs permitem utilizar a interface de consulta

---

<sup>3</sup> Mantem-se o termo “consulta”, por se tratar de BD. Recorde-se que o termo “busca” é usado para recuperação de documentos.

popularmente disseminada em SRIs, sem a necessidade de formalismos (como SQL, por exemplo) nem o conhecimento de estruturas internas do BD, como suas tabelas, atributos ou relacionamentos. Dessa forma, num banco de dados sobre filmes, uma consulta pelo nome de um ator e seu personagem poderia recuperar o(s) filme(s) em que ele atuou naquele papel; num banco de publicações científicas, uma consulta pelos nomes de dois autores poderia recuperar os nomes dos artigos que foram escritos em conjunto, ou aqueles em que o primeiro autor cita o segundo, por exemplo. Tudo isso sem que seja necessário conhecer o esquema ou a estrutura interna do BD.

Internamente, essas consultas exploram o esquema do BD: tuplas de tabelas contendo as palavras-chave explicitadas, junções de tabelas interconectadas, etc. As tuplas resultantes são idealmente relevantes para a consulta por palavras-chave.

No entanto, a tarefa não é trivial. Diferentemente da RI tradicional, em que os documentos são indexados a priori, não é possível indexar todas as combinações de todas as possíveis interconexões de tuplas que um BD pode conter [27]. Uma abordagem comumente usada consiste em construir um grafo, em que os nodos correspondem a tuplas e os arcos representam as interconexões das tuplas. O desafio então é eficientemente organizar subgrafos, ordenados por relevância à consulta, a fim de criteriosamente explorar os mais relevantes.

Diversos trabalhos podem ser citados [10, 43, 55, 56, 59, 71]. A principal crítica que se faz a todos eles é com relação à qualidade dos resultados das consultas [9, 27, 29, 95]. Também não se tem investido em metodologias padronizadas, análogas àquelas usadas na área de RI como as definidas em conferências TREC [92], que permitam comparar sistemas diferentes num mesmo ambiente, sob as mesmas condições.

Só para se ter uma ideia das discrepâncias já encontradas, um estudo [27] que se dedicou a avaliar nove trabalhos da área segundo os mesmos critérios e em três domínios diferentes concluiu que nenhum deles consegue superar os outros de forma consistente; vale salientar que os métodos avaliados abrangiam publicações ao longo de 7 anos (de 2002 a 2009), um intervalo considerável para que diferenças significativas pudessem ter sido registradas. Outro estudo mais recente dos mesmos autores [28] mostra que vários sistemas têm desempenho inaceitável dentro de ambientes de teste mais realistas do que aqueles apresentados nos artigos originais. Todas essas críticas vão no sentido da necessidade de mais padronização e uniformização da validação dos sistemas de consultas a BD por palavras-chave.

## 2.3 Recuperação de documentos através de linguagens de consulta estruturadas

Apesar de serem tradicionalmente tratados como fontes de dados não estruturados, documentos também podem conter trechos relativamente estruturados, como na associação de preço a um produto, ou na referência ao autor de uma publicação, ou na mera localização de uma loja. Esse tipo de informação possibilita desde a identificação de entidades (“autor”, “loja”, “produto”) até a associação atributo-valor, características comuns em BDs. Mas para que possam fazer parte de um BD, essas informações precisam ser inferidas a partir do contexto em que se encontram. Com isso, é possível efetuar consultas através de linguagens estruturadas, semelhantes a SQL, oferecendo maior poder de expressão da necessidade de informação do que o uso de palavras-chave.

Na maior parte dos trabalhos [15, 45, 46, 50], a informação estruturada é coletada diretamente dos documentos através de técnicas de EI - Extração de Informação [80], sendo que em [69] parte-se para o uso de técnicas de aprendizado de máquina (*Machine Learning* [31, 82]) para a identificação das entidades e seus valores. O reconhecimento da informação estruturada pode ser feito tanto de forma *off-line*, via pré-processamento dos documentos [15, 69], ou em tempo de consulta, como ocorre em [45, 46, 50].

Os métodos podem ser classificados também quanto à natureza dos esquemas de dados a serem reconhecidos: na maioria dos trabalhos as relações que serão reconhecidas devem ser determinadas a priori, mas há autores que propõem que os esquemas sejam descobertos de forma automática [15, 69].

Diferentemente das buscas convencionais sobre documentos o retorno das consultas são tuplas. Dessa forma, se reforça a ideia de uma abordagem pergunta-resposta, semelhante a trabalhos de *question-answering* em RI [36]: aqui se deseja a resposta a uma necessidade de informação, e não documentos que contenham a resposta.

## 2.4 Busca por documentos XML

A recuperação de documentos XML, também conhecida como XML RI, é uma área com grande quantidade de publicações [5, 34, 58, 81]. Sua importância para a integração entre documentos e BDs está relacionada à estrutura dos documentos XML, os quais compartilham características tanto das áreas de RI como de BDs. Assim, é possível o uso de texto livre, mas também se permite a definição de trechos com regras mais rígidas,

assemelhando-se aos esquemas definidos em um BD. Não por acaso é uma solução popular para troca de dados entre diferentes aplicações: é possível mapear todo um BD em XML, abstraindo detalhes das aplicações, e unificando a representação de dados.

A área objetiva a recuperação não apenas de documentos com informações relevantes à busca, mas principalmente dos trechos dentro do documento com maior chance de relevância. Assim, é útil para repositórios contendo documentos extensos, e que precisem cobrir uma grande quantidade de tópicos [52]. Por exemplo, considerando um acervo de manuais mapeados em XML, resultados relevantes devem apontar os trechos mais importantes dentro de um documento (capítulos, parágrafos, seções, etc.) e não apenas o documento completo.

A área possui também uma comunidade muito ativa voltada à avaliação de resultados, com Workshops anuais semelhantes às conferências já estabelecidas na área de RI, como as TREC's [92]. Nestes Workshops (como o INEX - *INitiative for the Evaluation of XML Retrieval* [5, 52], por exemplo), coleções de documentos e métricas de avaliação são constantemente aperfeiçoadas, constituindo um fórum importante para discussões entre a comunidade da área.

## **2.5 Extrair palavras-chave para recuperação de documentos a partir de consultas a BDs**

Como encontrar documentos relevantes à necessidade de informação contida numa consulta a BD? Na tentativa de responder a essa questão, algumas estratégias de integração partem para a associação indireta entre consultas a SGBDs e buscas a SRIs. Dessa forma, termos encontrados na consulta são submetidos a um SRI para a recuperação dos documentos correspondentes. Os termos podem estar presentes no corpo SQL da consulta ou nas tuplas referentes ao seu resultado.

Um bom número de referências [15, 30, 40, 61, 67, 69] apontam dois trabalhos como estado da arte nesta área: os métodos SCORE (*Symbiotic Context-Oriented Information REtrieval*) [65, 66, 75, 76] e SEMEX (*SEMantic EXplorer*) [57]. Sendo esses métodos os mais diretamente relacionados com esta tese, eles serão tratados com mais detalhe nas próximas seções. Ao final, um exemplo ilustra o funcionamento de ambos.

### **2.5.1 SCORE**

O método SCORE se volta exclusivamente ao resultado da consulta do BD, ordenando



cada elemento de tupla com relação aos demais da sua coluna. A ordenação se baseia num paradigma clássico em RI: elementos que possuam maior frequência no resultado e baixa frequência no resto do BD são tratados com maior peso. A função de ordenação segue a definição da Equação 2.1:

$$W(A, t) = N_Q(A, t) \log\left(\frac{1 + |R| - |Q(R)|}{1 + N_R(A, t) - N_Q(A, t)}\right)$$

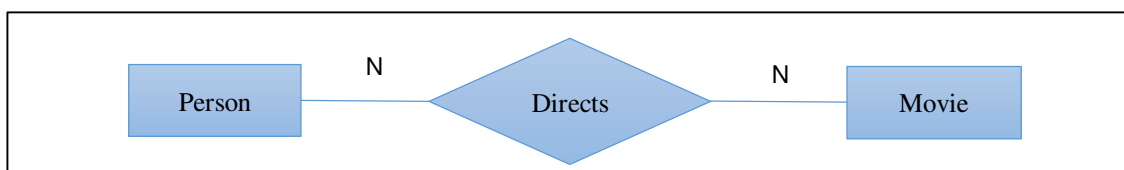
### Equação 2.1 - Função de ordenação de elementos de tupla de SCORE

A equação permite calcular o peso do elemento  $t$  com relação à coluna  $A$  –  $W(A, t)$  – relativo a uma tabela  $R$  com  $|R|$  tuplas, e uma consulta sobre  $R$  –  $Q(R)$  – contendo  $|Q(R)|$  tuplas. O número de vezes que o elemento  $t$  aparece na coluna  $A$  do resultado da consulta é representado por  $N_Q(A, t)$ , enquanto que  $N_R(A, t)$  representa o número de vezes em que  $t$  aparece na coluna  $A$  na tabela  $R$  como um todo. Os  $N$  elementos mais bem colocados após a ordenação são sugeridos como palavras-chave ao SRI, sendo  $N$  um parâmetro de configuração do método que deve ser ajustado manualmente.

SCORE também permite expandir a tabela  $R$  original através de junções com outras tabelas que possuam ligação com  $R$  via chaves estrangeiras. Após a junção,  $W$  é calculado da mesma forma como detalhado anteriormente.

### Um Exemplo

Para acompanhar o funcionamento do método, considere-se o exemplo a seguir. Seja o domínio “Filmes”, usando o BD disponível no IMDB (*Internet Movie DataBase*<sup>4</sup>). Um sub-esquema conceitual consistindo do relacionamento  $N \times N$  entre duas entidades (*Person* e *Movie*) é ilustrado na Figura 2.1.



**Figura 2.1 - Diagrama ER Person-Movie para o IMDB**

Considerem-se agora os filmes de Martin Scorsese; a Figura 2.2 ilustra uma possível consulta em SQL para essa necessidade de informação. Fragmentos do resultado da consulta são exibidos na Tabela 2.1.

<sup>4</sup> <http://imdb.com>, acessado em 21/10/2014

```

SELECT DISTINCT M.title, M.plot
FROM person as P, directs as D, movie as M
WHERE P.idperson=D.idperson and M.idmovie=D.idmovie
and P.name='Martin Scorsese'

```

**Figura 2.2 - Consulta SQL para os filmes de Martin Scorsese**

<b>M.title</b>	<b>M.plot</b>
gangs of new york (2002)	1863. america was born in the streets. in this movie, we see amsterdam vallon returning to the five points of america's, ...
new york, new york (1977)	the day wwii ends, jimmy, a selfish and smooth-talking musician, meets francine, a lounge singer...
shine a light (2008)	martin scorsese and the rolling stones unite in "shine a light," a look at the rolling stones....

**Tabela 2.1 - Fragmento do resultado da consulta exemplo**

Note-se que para os dados acima, SCORE vai gerar o mesmo valor de  $W(A,t)$  para todos os elementos de tupla no fragmento da consulta. Isso ocorre porque a frequência no resultado da consulta (componente  $N_Q$  da Equação 2.1) é a mesma para todos os elementos: todos aparecem apenas uma vez. O mesmo ocorre para o componente  $N_R$  na tabela completa  $R$ . Isso se explica pelo fato de que os nomes dos filmes são praticamente chave única; só haverá mais de uma ocorrência de um título de filme em caso de homônimos, o que é relativamente raro, e não ocorre no exemplo mostrado aqui. Além disso, dificilmente haverá repetição da trama de um filme (atributo “*plot*”).

Disso já decorre a primeira desvantagem do método, que considera os elementos de tupla como unidades básicas de ordenação. No caso deste exemplo, todos terão o mesmo peso. Como não há uma forma de desempate para elementos de mesmo peso, considerou-se a execução do método para  $N=6$ , isto é, usando todos os elementos do resultado como palavras-chave ao SRI. Cabe notar que esse seria um valor até conservador, já que os autores sugerem  $N=10$  como norma. O resultado é mostrado na Tabela 2.2.

Dos 10 primeiros links, 4 não são relevantes para a necessidade de informação (links de número 6, 7, 9 e 10). No entanto, numa visita às páginas relativas a esses links constata-se que os termos “new” e “york” aparecem com certa frequência; houve assim um desvio da busca no sentido de páginas contendo “new” e “york”, pois esses são também os termos mais frequentes no meio dos elementos de tupla da consulta. Assim se

configura outro problema com SCORE: as páginas recuperadas são dominadas pelos termos de maior frequência nos elementos de tupla selecionados (o que também ocorre com DBFIRE), mas sem que tenham necessariamente ligação com a necessidade de informação.

#	Título	Link
1	New York, New York (1977)	<a href="http://www.imdb.com/Title?New%20York,%20New%20York%20(1977)">http://www.imdb.com/Title?New York, New York (1977)</a>
2	Gangs of New York (2002)	<a href="http://www.imdb.com/Title?Gangs%20of%20New%20York%20(2002)">http://www.imdb.com/Title?Gangs of New York (2002)</a>
3	Shine a Light (2008)	<a href="http://www.imdb.com/Title?Shine%20a%20Light%20(2008)">http://www.imdb.com/Title?Shine a Light (2008)</a>
4	Ron (I) Wood	<a href="http://www.imdb.com/name/nm0939976">http://www.imdb.com/name/nm0939976</a>
5	Goodfellas (1990)	<a href="http://www.imdb.com/Title?Goodfellas%20(1990)">http://www.imdb.com/Title?Goodfellas (1990)</a>
6	Joni Mitchell	<a href="http://www.imdb.com/name/nm0593474">http://www.imdb.com/name/nm0593474</a>
7	The Musketeers of Pig Alley (1912)	<a href="http://www.imdb.com/Title?The%20Musketeers%20of%20Pig%20Alley%20(1912)">http://www.imdb.com/Title?The Musketeers of Pig Alley (1912)</a>
8	Martin Scorsese	<a href="http://www.imdb.com/name/nm0000217">http://www.imdb.com/name/nm0000217</a>
9	On the Town (1949)	<a href="http://www.imdb.com/Title?On%20the%20Town%20(1949)">http://www.imdb.com/Title?On the Town (1949)</a>
10	Billie Jack	<a href="http://www.imdb.com/name/nm1879891">http://www.imdb.com/name/nm1879891</a>

**Tabela 2.2 - 10 primeiros links da busca com SCORE**

Por fim, o uso do conteúdo completo dos elementos de tuplas como unidades de ordenação faz com que não se tenha controle da sobrecarga do método sobre o tempo de busca: cada elemento pode conter um número indeterminado de termos, e ao se incluir o seu conteúdo inteiro na busca ao SRI o tempo de execução pode variar também de forma indeterminada. Mais ainda: vários termos que em princípio não deveriam ser usados na busca (*stopwords* [32], por exemplo), podem ser incluídos com a mesma importância que os demais.

### 2.5.2 SEMEX

A recuperação de documentos associados a uma consulta é apenas um dos recursos disponíveis no sistema SEMEX. Na verdade, constitui-se num ambiente bem mais amplo, voltado ao gerenciamento de informações pessoais (*Personal Information Management*). Apenas a parte que está relacionada a DBFIRE (apresentada no artigo [57]) é que será

detalhada aqui.

Enquanto SCORE explora somente o resultado da consulta ao BD, SEMEX se volta exclusivamente para o corpo SQL da consulta, de onde são extraídos os termos sugeridos para a busca ao SRI.

Inicialmente o método faz uma análise da sentença, na busca por literais, nomes de tabelas e atributos. Com isso, a consulta é modelada como um grafo com nodos e arcos rotulados pelos itens detectados. Os nodos são formados pelos literais em cláusulas WHERE e nomes de tabelas. Um nodo especial é criado para cada atributo que aparece na cláusula SELECT. Atributos referentes a ids de tabelas são descartados.

Os arcos por sua vez são determinados por pares atributo/literal em cláusulas WHERE, unindo o nodo referente à tabela que contém o atributo até o literal; o rótulo do arco é o nome do atributo; arcos são também criados a partir de junções entre tabelas. Os rótulos dos arcos nestes casos são os nomes dos atributos envolvidos.

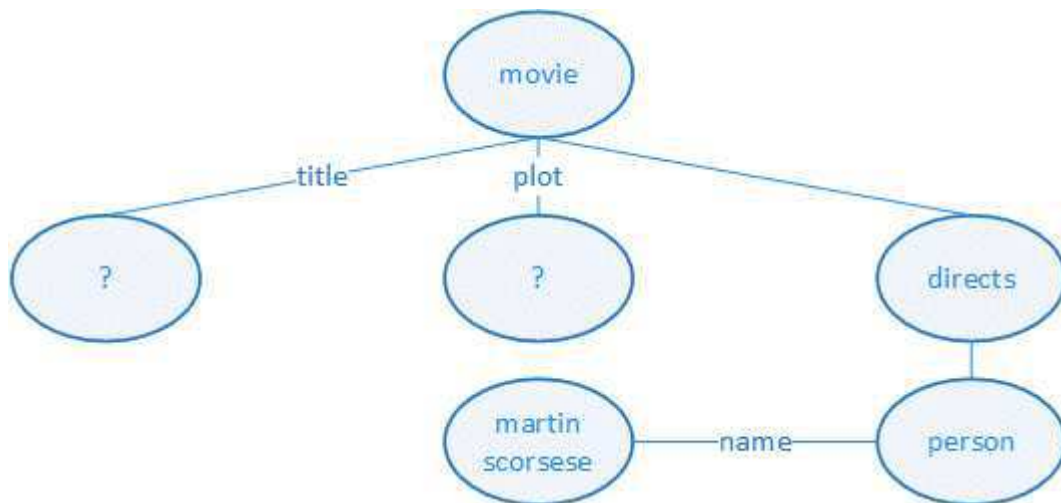
A construção do grafo também permite que ele seja simplificado caso haja uma sequência de nodos  $N_0, \dots, N_t$ , em que cada um dos nodos  $N_i$ ,  $i \in [1, t - 1]$ , possua apenas dois vizinhos,  $N_{i-1}$  e  $N_{i+1}$ . Os nodos intermediários são removidos, criando um único arco entre  $N_0$  e  $N_t$ , rotulado com os rótulos dos nodos removidos.

Tanto os nodos como os arcos do grafo possuem pesos que refletem sua probabilidade de retornar documentos relevantes. A seleção das palavras-chave que serão enviadas ao SRI é feita após uma série de percursos no grafo.

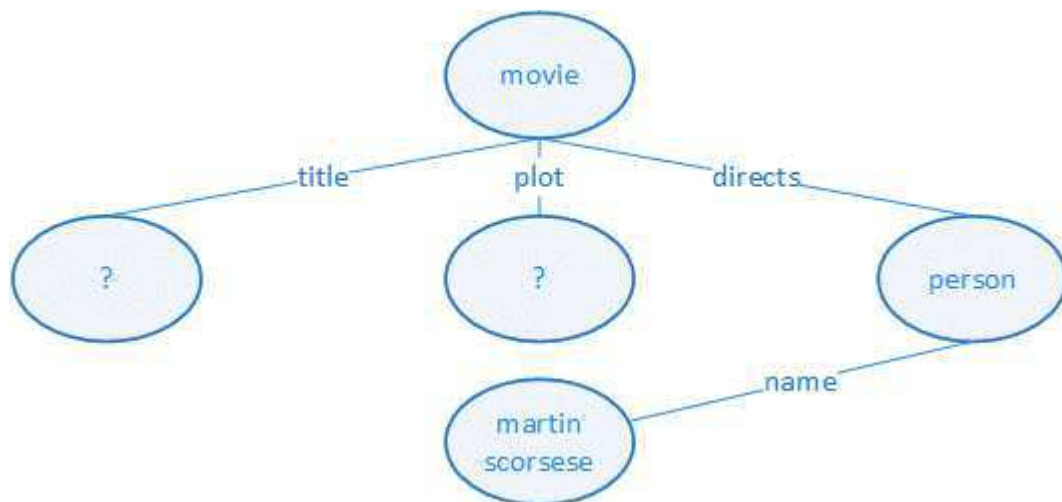
Cada percurso seleciona o item (nodo ou arco) com maior peso, o qual, uma vez selecionado, determina o recálculo dos pesos dos demais: a noção aqui é que cada item selecionado determina uma diminuição nos pesos dos itens associados a ele. O processo é repetido até que o maior peso encontrado seja menor que um valor mínimo estabelecido. As palavras-chave sugeridas correspondem aos rótulos dos itens selecionados ao longo dos vários percursos no grafo. Por padrão, os literais possuem os maiores pesos, e são sempre sugeridos.

### **Exemplo**

Considere-se o mesmo exemplo apresentado na seção 2.5.1, a consulta por filmes de Martin Scorsese. O grafo relativo à consulta mostrada na Figura 2.2 é apresentado na Figura 2.3 e sua simplificação é exibida na Figura 2.4, como sendo a versão definitiva para esta consulta.



**Figura 2.3 - Grafo para a consulta sobre filmes de Martin Scorsese**



**Figura 2.4 – Versão simplificada para o grafo da Figura 2.3**

Para este grafo, as palavras-chave sugeridas são:

martin scorsese movie directs

Os 10 primeiros links com o resultado da busca são mostrados na Tabela 2.3.

Considerando que a necessidade de informação é “*films directed by Martin Scorsese*”, pode-se dizer que o método fornece uma boa aproximação para o que se quer. E o resultado condiz com isso, pois apenas 1 link da lista poderia não ser considerado relevante: o filme de número 9, pois apesar de ser estrelado por Martin Scorsese, não foi dirigido por ele.

#	Título	Link
1	Martin Scorsese	<a href="http://www.imdb.com/name/nm0000217">http://www.imdb.com/name/nm0000217</a>
2	Raging Bull (1980)	<a href="http://www.imdb.com/Title?Raging Bull (1980)">http://www.imdb.com/Title?Raging Bull (1980)</a>
3	Thelma Schoonmaker	<a href="http://www.imdb.com/name/nm0774817">http://www.imdb.com/name/nm0774817</a>
4	Taxi Driver (1976)	<a href="http://www.imdb.com/Title?Taxi Driver (1976)">http://www.imdb.com/Title?Taxi Driver (1976)</a>
5	A Personal Journey with Martin Scorsese Through American Movies (1995) (TV)	<a href="http://www.imdb.com/title?A Personal Journey with Martin Scorsese Through American Movies (1995) (TV)">http://www.imdb.com/title?A Personal Journey with Martin Scorsese Through American Movies (1995) (TV)</a>
6	Cape Fear (1991)	<a href="http://www.imdb.com/Title?Cape Fear (1991)">http://www.imdb.com/Title?Cape Fear (1991)</a>
7	John Schoonraad	<a href="http://www.imdb.com/name/nm0593474">http://www.imdb.com/name/nm0593474</a>
8	The Age of Innocence (1993)	<a href="http://www.imdb.com/Title?The Age of Innocence (1993)">http://www.imdb.com/Title?The Age of Innocence (1993)</a>
9	Scorsese on Scorsese (2004) (TV)	<a href="http://www.imdb.com/Title?Scorsese on Scorsese (2004) (TV)">http://www.imdb.com/Title?Scorsese on Scorsese (2004) (TV)</a>
10	The Last Temptation of Christ (1988)	<a href="http://www.imdb.com/title? The Last Temptation of Christ (1988)">http://www.imdb.com/title? The Last Temptation of Christ (1988)</a>

**Tabela 2.3 – Resultado para a busca com SEMEX**

Essa pode ser apontada como a desvantagem do método: apesar de se sugerir termos próximos à necessidade de informação, não se aproveita do resultado da consulta, que em tese contém a resposta para o que se quer. Isso poderia eliminar a recuperação de *falsos positivos*, como foi o caso mostrado aqui.

Mesmo a aproximação da necessidade de informação não consegue ser melhor que uma busca por palavras-chave feita diretamente pelo usuário, como será mostrado nos experimentos do Capítulo 7. Além disso, o método não é capaz de tratar uma variedade de consultas: aquelas em que não haja literais para cláusulas WHERE (como em: SELECT \* FROM TABELA), ou que sejam relativas a operadores lógicos que não o de igualdade (como em: SELECT \* FROM TABELA WHERE ATRIBUTO <> VALOR).

## 2.6 Discussão

Vê-se que dentre as alternativas disponíveis para recuperação de documentos relacionados a consultas a BD, há problemas que justificam a busca por um novo método, que forneça ao usuário mais qualidade de resultados. Esse método pode usar algumas das

ideias de SEMEX e SCORE, mas de uma forma mais apropriada.

SCORE não utiliza nenhum dado sobre a necessidade de informação do usuário, e usa elementos de tupla inteiros como átomos para o SRI. Se o usuário puder especificar o que deseja e se os itens básicos enviados ao SRI forem termos, fica mais fácil estabelecer níveis mais apropriados de importância entre eles.

Quanto a SEMEX, o corpo da consulta SQL fornece uma boa indicação da necessidade de informação do usuário, mas o resultado da consulta pode estabelecer um contexto melhor do que se quer.

Essas ideias motivaram o método descrito mais adiante no Capítulo 4. Ele utiliza conceitos de expansão de buscas, tema que será abordado no capítulo a seguir.

# Capítulo 3 - Expansão Automática de Buscas

## 3.1 Introdução

Como já mencionado no Capítulo 1, um dos problemas clássicos na área de RI é a chamada incompatibilidade de termos (*term mismatch*) [35]: os termos usados pelos usuários em suas buscas por palavras-chave podem ser diferentes, ou com semântica diversa daqueles efetivamente encontrados nos documentos da coleção. Esse problema se apresenta comumente em duas dimensões: sinonímia e polissemia.

O problema da sinonímia relaciona-se com o uso de palavras diferentes mas de mesmo sentido ou similar. Por exemplo, na busca pela palavra “crime”, é possível que documentos contendo “delito”, “infração” ou variações como “criminoso” ou “criminal” sejam também relevantes. No entanto, o usuário não pode (ou não deseja) antecipar todos seus diferentes sinônimos em tempo de busca. Fazendo-se a busca apenas pela palavra “crime”, documentos potencialmente relevantes que não contenham os termos de busca, mas que contenham suas variações vão necessariamente ficar de fora da lista resultante. Esse problema impacta negativamente na *revocação* do SRI, i.e., a fração dos documentos relevantes que foram de fato retornados na busca<sup>5</sup>.

A questão da polissemia, por sua vez, está relacionada aos diversos sentidos que uma mesma palavra pode assumir. Um exemplo clássico é a palavra “banco”: ela pode se referir tanto a um termo da Computação (“banco de dados”), ou a uma instituição financeira (“Banco Central”) ou mesmo a um objeto (“banco de plástico”). É comum que não se consiga explicitar o contexto desejado na própria sentença de busca, o que pode fazer com que se recuperem documentos irrelevantes. Com isso, tem-se um impacto

---

<sup>5</sup> Mais detalhes sobre métricas e avaliação de desempenho em RI no Capítulo 5.



negativo na métrica *precisão*, i.e., o percentual de documentos relevantes presentes no resultado da busca<sup>6</sup>.

Esses problemas são ainda mais agudos quando se considera que grande parte das buscas é realizada com até 3 palavras (dados de 2012 relativos aos EUA)<sup>7</sup>. Uma forma bastante difundida de atacar ambos os casos é através da expansão da busca inicial do usuário, incluindo outros termos que melhor definam sua necessidade de informação. Assim, uma busca mais expressiva pode tratar a sinonímia (com a inclusão de termos equivalentes) mas também a polissemia (incluindo termos que ajudem a definir o contexto correto que se deseja na recuperação dos termos originais de busca).

A expansão pode ocorrer de diversas formas. Há, por exemplo, métodos interativos de expansão (chamados de IQE – *Interactive Query Expansion*) [51, 77] que incluem o usuário como parte ativa do processo, cabendo a ele a decisão sobre quais termos devem ser usados. A dinâmica de DBFIRE pressupõe a execução de uma consulta a BD e o envio automático ao SRI dos termos de expansão. Por este motivo, o foco aqui será apenas em métodos automáticos de expansão (chamados de AQE – *Automatic Query Expansion*).

### 3.2 Expansão Automática de Buscas (AQE)

Apesar de um grande número de abordagens se voltarem à estratégia de AQE, elas compartilham princípios semelhantes de funcionamento. Esse funcionamento é diretamente ligado à forma como os SRIs efetuam a ordenação (*ranking*) dos documentos numa busca.

De modo geral, esse *ranking* leva em conta a maior similaridade entre os documentos recuperados e os termos de busca, a qual é comumente medida de acordo com a Equação 3.1 [19]. Nesta equação, a função  $sim(q, d)$  relaciona uma busca  $q$  a um documento  $d$ , sendo que o resultado da busca é apresentado em ordem decrescente de  $sim(q, d)$ , formando um *ranking* de similaridade.

$$sim(q, d) = \sum_{t \in q \cap d} w_{t,q} w_{t,d}$$

#### Equação 3.1 - Ranking de similaridade entre documentos e buscas

---

<sup>6</sup> Mais detalhes sobre métricas e avaliação de desempenho em RI no Capítulo 5.

<sup>7</sup> <http://press.experian.com/United-States/Press-Release/experian-marketing-services-reports-google-share-of-searches-at-65-percent-in-may-2012.aspx>, acessado em 21/10/2014

Nesta equação,  $w_{t,q}$  e  $w_{t,d}$  representam os pesos do termo  $t$  com relação à busca  $q$  e ao documento  $d$ , respectivamente. A maior parte dos sistemas de RI se baseia nessa heurística, seja total ou parcialmente. Neste caso, os pesos  $w_{t,q}$  e  $w_{t,d}$  são diretamente proporcionais às suas frequências na busca  $q$  e no documento  $d$ , respectivamente.

A partir destas definições, o objetivo da expansão da busca é gerar uma nova busca  $q'$  com mais termos e novos pesos  $w'$ , de maneira que a similaridade possa ser calculada de forma análoga à da Equação 3.1:

$$sim(q', d) = \sum_{t \in q' \cap d} w'_{t,q'} w_{t,d}$$

### **Equação 3.2 - Similaridade entre busca/documento (versão expandida)**

Os novos pesos  $w'$  são calculados de acordo com o tipo de abordagem usada para expansão. No trabalho de Carpineto [19], apresenta-se uma classificação dessas abordagens segundo os princípios que cada uma utiliza. Assim, há desde abordagens que se utilizam de relações morfo-sintáticas ou semânticas, extraindo termos a partir de thesauri ou ontologias, até aquelas que analisam logs de buscas, ou mesmo expandindo buscas a partir do resultado de buscas na Web. Vê-se que essa classificação diz respeito ao *corpus de expansão*, ou seja, à *origem* dos termos que serão usados.

Ainda seguindo a classificação de Carpineto, há uma outra classe de métodos, classificados como métodos de *análise local*, que utilizam como corpus de expansão um pequeno extrato relativo ao resultado da busca original do usuário. Os métodos dessa classe são também conhecidos como métodos de *retroalimentação por pseudo-relevância* (*pseudo-relevance feedback* – PRF).

Tais métodos possuem bastante semelhança com o contexto em que se dará a nossa proposta de integração SGBDs/SRI: consideramos que o SGBD dispõe do resultado de uma consulta a BD. Como esse resultado é assumidamente exato, faz sentido pensar numa abordagem que o utilize como corpus para expansão. É um procedimento análogo àquele utilizado por métodos de PRF, os quais supõem que os primeiros documentos relativos à busca do usuário são relevantes; conjunto dos primeiros documentos da busca inicial é usado como corpus de expansão.

Daqui em diante, sempre que se fizer referência a métodos de expansão de buscas, estaremos nos referindo a essa classe de métodos.

### 3.3 Funcionamento de métodos de PRF

Esses métodos derivam de outro método bastante conhecido em RI, voltado também para expansão de buscas, mas no qual o usuário participa do processo. É a chamada retroalimentação por relevância (*relevance feedback - RF*) [74]. Em RF, uma primeira rodada de busca com os termos originais do usuário é realizada, e os primeiros documentos são manualmente avaliados pelo usuário como relevantes ou não; pode-se dizer, portanto, que RF é também um tipo de método de expansão interativa (IQE).

Após a avaliação dos documentos, uma abordagem possível é escolher os termos para expansão de acordo com a fórmula Rocchio (com modificações sugeridas por Salton e Buckley em [78]):

$$\vec{q}_f = \alpha \vec{q}_0 + \beta \sum_{d \in D_r} \frac{\vec{d}}{|D_r|} - \gamma \sum_{d \in D_{nr}} \frac{\vec{d}}{|D_{nr}|}$$

**Equação 3.3 - Fórmula Rocchio**

Nesta equação, considera-se  $q_f$  como sendo o vetor final com os termos para expansão,  $q_0$  correspondendo à busca inicial também na forma vetorial,  $D_r$  representando o conjunto dos documentos marcados como relevantes e  $D_{nr}$  o conjunto daqueles marcados como não relevantes. Cada elemento de  $D_r$  e  $D_{nr}$  é também um vetor de termos. Nesta representação vetorial, a  $i$ -ésima posição em um vetor representa o peso do  $i$ -ésimo termo do vocabulário, em função de sua frequência seja na busca inicial  $q_0$  seja em documentos  $d$  pertencentes a  $D_r$  ou  $D_{nr}$ .

Os coeficientes  $\alpha$ ,  $\beta$  e  $\gamma$  variam de 0 a 1, calibrando a retroalimentação. Assim, pode-se dar mais peso aos termos da busca inicial (com o coeficiente  $\alpha$ ), ou àqueles encontrados nos documentos relevantes (através de  $\beta$ ) ou àqueles encontrados nos documentos não relevantes (com o valor de  $\gamma$ ). Uma configuração típica para esses parâmetros é  $\alpha=1$ ,  $\beta=0.75$  e  $\gamma=0.15$  [62].

Ao se efetuar a expansão via PRF com base na fórmula Rocchio, exclui-se o usuário do processo, assumindo como relevantes os primeiros  $k$  documentos retornados pela busca inicial; considera-se  $k$  como parâmetro de configuração. Como não se tem evidências de documentos não relevantes, o normal é tratar o coeficiente  $\gamma$  da fórmula Rocchio como zero.

Após a análise dos primeiros  $k$  documentos, cria-se uma lista de termos candidatos

ordenados por seu potencial para a expansão; os  $n$  primeiros termos da lista são usados para a expansão, onde  $n$  é também um parâmetro configurável.

Abordaremos aqui três métodos de PRF, extensamente referenciados na literatura, os quais foram utilizados também nos comparativos frente a DBFIRE. Sua escolha deveu-se por comporem a infraestrutura nativa para expansão de buscas de dois SRIs bastante populares: Indri<sup>8</sup> e Terrier<sup>9</sup>. Os métodos escolhidos, por sua vez, constituem exemplos de duas subclasses de métodos de PRF: métodos baseados em distribuições de termos e métodos baseados em modelos de linguagem.

Apesar de reconhecermos que o estado da arte em expansão de buscas passa por métodos de aprendizado de máquina (como em [16], por exemplo), optamos por deixar seu comparativo frente a DBFIRE para os trabalhos futuros (ver Capítulo 8).

### 3.3.1 PRF baseada em distribuições de termos

Vários trabalhos se utilizam das diferenças entre as distribuições dos termos no conjunto dos  $k$  primeiros documentos e aquela encontrada no restante da coleção. A motivação para essa heurística vem do fato de que os termos com alta frequência nos documentos tratados como relevantes e baixa frequência na coleção completa tendem a discriminar melhor entre documentos relevantes e não relevantes, levando com isso a melhorar também o resultado das buscas expandidas.

Abordaremos aqui dois desses métodos: *Kullback-Lieber Distance* (KLD [17]) e *Divergence from Randomness* (DFR [2]). Ambos levam em conta dois conjuntos de documentos: o conjunto dos documentos relevantes, formado pelos  $k$  primeiros documentos retornados (chamado aqui de  $R$ ), e o conjunto de todos os documentos da coleção (tratado aqui como  $C$ ).

No método KLD, a motivação é de que quanto maior o grau de desordem (i.e., entropia relativa) entre as distribuições de um termo no conjunto dos documentos relevantes e no conjunto completo, maior a importância desse termo para expansão. Essa diferença pode ser medida usando-se um conceito conhecido na Teoria da Informação, chamado de distância Kullback-Lieber. A aplicação dessa ideia para PRF leva à Equação 3.4:

---

<sup>8</sup> <http://www.lemurproject.org/indri.php>, acessado em 21/10/2014

<sup>9</sup> <http://terrier.org>, acessado em 21/10/2014

$$ord(t) = \sum_{d \in R} p_r(t) \log\left(\frac{p_r(t)}{p_c(t)}\right)$$

### Equação 3.4 - Peso de um termo para o método KLD

Nesta equação,  $ord(t)$  é o peso final do termo  $t$  para a retroalimentação,  $p_r(t)$  e  $p_c(t)$  indicam, respectivamente, a probabilidade de ocorrência do termo  $t$  no conjunto de documentos relevantes e a probabilidade de ocorrência de  $t$  no conjunto de todos os documentos da coleção. As probabilidades  $p_r$  e  $p_c$  são estimadas a partir da fração das ocorrências do termo  $t$  com relação ao conjunto em questão ( $R$  ou  $C$ ); essa fração trata o conjunto base como sendo uma única longa sequência composta por vários termos.

O método DFR por sua vez, infere a utilidade de um termo a partir da divergência entre sua distribuição nos documentos relevantes e numa distribuição aleatória da coleção de documentos. O método permite que várias distribuições de probabilidade modelem essa distribuição aleatória; dentre elas, os melhores resultados foram obtidos com a chamada distribuição Bose-Einstein [70]. Com isso, os termos são ordenados de acordo com a Equação 3.5:

$$ord(t) = tf_R(t) \log_2\left(\frac{1 + P_n}{P_n}\right) + \log(1 + P_n)$$

$$P_n = \frac{tf_C(t)}{N}$$

### Equação 3.5 - Peso de um termo para o método DFR

Na equação acima,  $tf_R(t)$  e  $tf_C(t)$  correspondem, respectivamente, à frequência absoluta do termo  $t$  nos conjuntos  $R$  e  $C$ , e  $N$  é o número total de documentos em  $C$ .

### 3.3.2 PRF baseada em modelos de linguagem (*language models*)

Outra abordagem bastante usada em PRF é a construção de um modelo estatístico para a busca (conhecido como modelo de linguagem – *language model*), especificando uma distribuição de probabilidade sobre termos; os termos mais úteis são aqueles com as maiores probabilidades [19]. Aqui os documentos são tratados como modelos e as buscas são consideradas amostras aleatórias relativas a esses modelos [53].

Um dos métodos que segue essa linha é o chamado modelo de linguagem baseado em relevância (*Relevance-Based Language Model*, aqui chamado *RM*): ele procura estimar a probabilidade de se observar um novo termo  $t$  dado que já se observaram os  $n$

termos da busca. Os termos podem ser então ordenados em função dessas probabilidades para posterior expansão. Supondo que as ocorrências dos termos são independentes entre si, a função  $ord(t)$  pode ser então descrita como na Equação 3.6:

$$ord(t) = \sum_{d_i \in R} p(d_i) * p(t|d_i) * \prod_{q \in Q} p(q|d_i)$$

$$p(d_i) = \frac{k - i + 1}{k}$$

### **Equação 3.6 - Peso de um termo para o método RM**

Nesta equação,  $p(t|d_i)$ , assim como  $p(q|d_i)$ , correspondem às probabilidades de se ver um termo genérico  $t$  ou um termo de busca  $q$  em um dos documentos  $d_i$  da retroalimentação. Já  $p(d_i)$  serve para dar maior peso aos documentos nos primeiros postos dentre os  $k$  primeiros documentos em  $R$ ; pode ser estimado pela posição relativa em que o documento aparece.

Um problema neste método é que certamente haverá documentos que não conterão termos da busca inicial. Nestes casos, a probabilidade  $p(q|d_i)$  será nula, anulando também a contribuição do documento  $d_i$  para a expansão. Uma segunda versão do método [72], considera também a frequência dos termos de busca na coleção completa, calculando a probabilidade de um termo aparecer num documento através da função  $p'$  de acordo com a Equação 3.7. Essa modificação é o que se chama de suavização (*smoothing*).

$$p'(q|d) = 0.2 p_c(q) + 0.8 p(q|d)$$

### **Equação 3.7 - Método RM (versão com *smoothing*)**

Essa versão utiliza a probabilidade de ocorrência do termo de busca na coleção completa ( $P_c$ ) junto com a probabilidade propriamente dita de ele ocorrer no documento  $d$ . Caso ela seja nula, o termo ainda contribui com seu  $P_c$ , ainda que com menor peso. É essa a versão utilizada nos comparativos com DBFIRE.

## **3.4 Discussão**

Métodos de PRF tendem a melhorar a qualidade dos resultados em buscas por palavras-chave, o que é atestado por um grande número de resultados publicados na literatura. No entanto, esses mesmos resultados evidenciam que seu comportamento varia bastante entre buscas, podendo mesmo gerar resultados piores do que aqueles sem expansão. Um estudo aponta que há mais termos que tendem a piorar os resultados do que melhorá-los, sendo

que em sua grande maioria os termos candidatos a expansão tendem a não exercer influência alguma [16].

Um dos principais pontos para o sucesso ou fracasso da expansão (qualquer que seja o método, é bom que se diga) é o corpus inicial de expansão, o conjunto  $R$ : se ele for rico em informação relevante, maior a chance de a expansão extrair termos úteis. Mas como ele é gerado a partir da busca inicial do usuário, sujeita aos clássicos problemas de sinonímia e polissemia, nada garante que eles venham a ser de fato relevantes. Isso significa que usá-los como evidência de relevância pode levar à sugestão de termos sem a mínima relação com a necessidade de informação original do usuário.

Há portanto uma oportunidade a ser explorada: caso o conjunto de documentos usado para expansão seja originado de uma fonte com maior certeza de correteza (por exemplo, o resultado de uma consulta a BD), há uma chance de a expansão produzir melhores resultados, já que em tese usa-se um conjunto com mais informação relevante.

Outro fator que pode impactar negativamente na qualidade dos resultados é a necessidade de calibragem dos parâmetros dos métodos. No mínimo, dois parâmetros precisam ser ajustados: o número de documentos da retroalimentação (valor de  $k$ ), e a quantidade de termos adicionados (o valor de  $n$ ). Como as buscas podem variar bastante, usar valores fixos para esses parâmetros em todas as buscas pode não ser a melhor decisão. Há várias evidências de que os resultados tendem a melhorar sensivelmente se os parâmetros puderem ser ajustados caso a caso [11, 18]. Por outro lado, é muito difícil fazê-lo fora do ambiente controlado dos experimentos em laboratório, sujeito a buscas imprevisíveis, de tamanhos diversos e com coleções de documentos em constante alteração [19].

Outra questão importante se relaciona à sobrecarga adicionada à busca, seja devido ao processamento dos  $k$  primeiros documentos da retroalimentação, seja pela própria sobrecarga relativa à inclusão dos  $n$  novos termos. Destes dois fatores, o mais crítico é a quantidade de termos adicionada para expansão: alguns experimentos reportam que valores de  $n$  entre 10 e 20 acrescentam um fator de 10 para o tempo final de processamento [19]. Diminuindo o valor dos parâmetros  $k$  e  $n$  têm-se uma redução no tempo final de execução da retroalimentação, mas pode levar à diminuição da qualidade dos resultados das buscas [22]. Dessa forma, o ajuste desses parâmetros também deve levar em conta esse fator.

# Capítulo 4 - O Método DBFIRE

## 4.1 Introdução

Esta tese propõe um método para integrar BDs e documentos através da expansão de buscas, usando como corpus de análise para expansão as tuplas referentes ao resultado de uma consulta a BD. Dessa forma, um conjunto inicial de palavras-chave deve ser expandido com termos potencialmente úteis extraídos do resultado da consulta ao BD. As palavras-chave iniciais podem ser fornecidas pelo usuário, ou extraídas a partir do próprio corpo da consulta, considerando seus literais como sementes da expansão.

O ponto de entrada do método, portanto, é uma consulta a um BD, a qual se supõe conhecida de antemão. Tal consulta pode ter sido gerada por um relatório gerencial da organização ou mesmo por uma consulta direta feita por um usuário. O objetivo do método é então sugerir os melhores termos que possam trazer documentos relevantes para essa consulta ao BD.

Apesar de a expansão parecer uma ideia promissora por si só, há várias possibilidades para se descobrir os termos potencialmente úteis para expansão. Em qualquer método de expansão, os termos candidatos são ordenados de acordo com alguma heurística que visa estimar sua utilidade para expansão. No entanto, como se verá nos experimentos detalhados no Capítulo 7, usar métodos tradicionais de expansão pode não ser muito melhor do que uma consulta direta elaborada pelo próprio usuário ao SRI onde estão armazenados os documentos. Dessa forma, viu-se uma oportunidade para inovação considerando o contexto em que a expansão será realizada: resultados de consultas a BDs.

Já que tais resultados podem ser considerados exatos, a hipótese investigada foi de que quanto mais difundido um termo estiver ao longo das tuplas de resposta à consulta, mais útil esse termo deverá ser para expansão. Para isso, considere-se a seguinte definição já apresentada no Capítulo 1: o resultado de uma consulta a BD é formado por um



conjunto de tuplas, e cada tupla  $t_i$  é representada por uma sequência de  $n$  elementos (ou seja,  $t_i = \langle e_1, e_2, \dots, e_n \rangle$ ). Por sua vez, cada elemento de tupla  $e_j$ , possui um multi-conjunto com  $m$  termos (ou seja,  $e_j = \{ w_1, w_2, \dots, w_m \}$ ).

Dessa forma, estima-se a difusão dos termos atribuindo maior peso àqueles com maior frequência ao longo de todas as tuplas do resultado da consulta a BD, mas que também apareçam em um número maior de elementos dessas tuplas.

A função de ordenação de termos serve não só para estabelecer uma ordem de importância entre os termos candidatos à expansão, mas também para definir que peso o SRI deve atribuir a eles na busca expandida. Como visto no Capítulo 3, esse é um fator de grande importância para o sucesso da expansão, pois a adição direta de termos ao conjunto inicial de palavras-chave pode fazer com que a qualidade da recuperação diminua bastante, fazendo o SRI trazer muitos documentos irrelevantes: é o efeito do desvio de foco da busca, ou *query drift* [64].

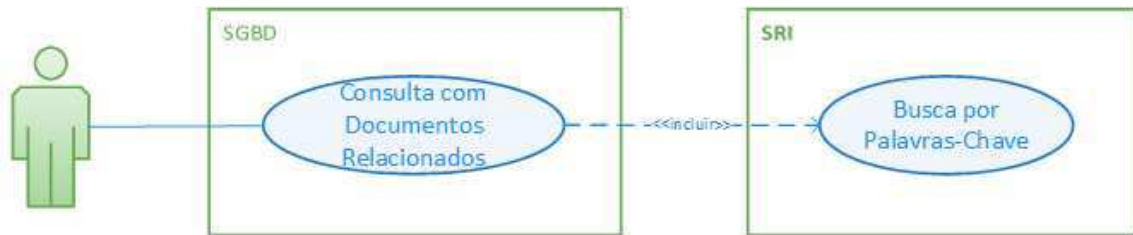
Essas são as linhas gerais do método DBFIRE (*DataBases For Information Retrieval* – Bancos de Dados para Recuperação de Informação), objeto desta tese, o qual será detalhado nas seções a seguir. O capítulo termina com um exemplo do funcionamento de DBFIRE em um domínio específico, “Filmes”.

## 4.2 DBFIRE em Detalhes

As próximas sub-seções apresentam mais detalhes sobre o método DBFIRE, sua arquitetura, a forma de ordenação de termos, e a maneira como o sistema monta buscas a um SRI a partir de consultas direcionadas a um SGBD.

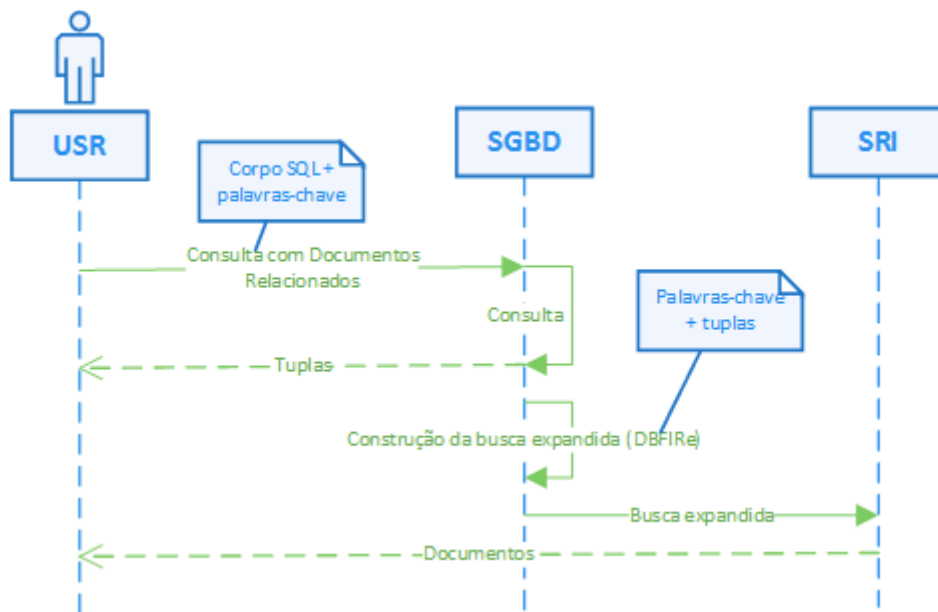
### 4.2.1 Arquitetura

Considere-se que do ponto de vista do usuário, o SGBD proverá uma nova funcionalidade, a consulta com documentos relacionados. Para isso, o SGBD interage com o SRI para devolver os documentos de interesse ao usuário. Isso é ilustrado no diagrama de caso de uso [12] apresentado na Figura 4.1. Vê-se que o caso de uso “Consulta com Documentos Relacionados” inclui o caso de uso “Busca por Palavras-Chave”, que é a funcionalidade principal de um SRI.



**Figura 4.1 - Diagrama de caso de uso para a nova funcionalidade do SGBD**

O fluxo de mensagens usuário/SGBD/SRI é detalhado no diagrama de sequência [12] exibido na Figura 4.2.



**Figura 4.2 - Diagrama de sequência para a consulta com documentos relacionados**

Assim, a consulta com documentos relacionados é iniciada com o usuário fornecendo uma consulta SQL ao BD e um conjunto de palavras-chave descrevendo sua necessidade de informação; caso opte por não fornecê-las, elas serão extraídas do próprio corpo da consulta SQL, através dos literais que nela venham a ocorrer. No entanto, a opção de se usar os literais como palavras-chave a serem expandidas não faz sentido para consultas em que eles não estão disponíveis (*SELECT \* FROM TABELA*), ou se forem objeto de operadores de negação (usando operadores como *NOT*, ou  $\langle \rangle$ ), intervalos (com os operadores  $\langle$ ,  $\langle =$ ,  $\rangle$ ,  $\rangle =$ ), ou expressões regulares (usando-se o operador LIKE, por exemplo).

A consulta é executada normalmente, sendo seu resultado devolvido diretamente ao usuário. Para a recuperação dos documentos relacionados, as tuplas são então analisadas para a determinação dos melhores termos para expansão; é aí que entra o

método DBFIRE. Depois da sentença de busca definitiva ser determinada ela é enviada ao SRI, que a executa e devolve os documentos resultantes para o usuário.

#### 4.2.2 Ordenação dos termos em DBFIRE

DBFIRE estima a utilidade dos termos do resultado de uma consulta a BD através de uma heurística baseada em duas probabilidades relativas às ocorrências dos termos no conjunto de tuplas de resposta: a primeira probabilidade toma como espaço amostral todas as ocorrências de termos no resultado da consulta, considerando-o como uma longa sequência de termos – é o que se chamou de *probabilidade na sequência* -  $P_s$ ; a segunda probabilidade usa os elementos de cada tupla como origem do espaço amostral – essa é a *probabilidade nos elementos* -  $P_e$ . Essas duas probabilidades são ilustradas na Figura 4.3.

<i>Nome</i>	<i>Profissão</i>	<i>Naturalidade</i>
João <b>Paulo</b>	Engenheiro	Rio de Janeiro
José Antônio	Advogado	<b>São Paulo</b>
<b>Paulo</b> André	Professor	<b>São Paulo</b>

$$\begin{cases} P_s(\text{paulo}) = 4/16 = 0,25 \\ P_e(\text{paulo}) = 4/9 = 0,44 \end{cases}$$

**Figura 4.3 - Probabilidades de DBFIRE**

Na figura vê-se um resultado fictício de uma consulta a uma tabela de um BD, sendo exibidas um total de 3 tuplas, cada uma com 3 colunas (nome, profissão e naturalidade), totalizando 9 elementos, portanto. As tuplas retornadas apresentam, por sua vez, vários termos, totalizando 16 ocorrências relativas à sequência completa de termos. Assim, para o cálculo da probabilidade na sequência ( $P_s$ ) para um termo qualquer, considera-se a fração de suas ocorrências com relação ao total de ocorrências de todos os termos, enquanto que para a probabilidade nos elementos ( $P_e$ ) leva-se em conta o número de elementos em que o termo ocorre. Na Figura 4.3, as ocorrências individuais do termo *paulo* são reforçadas em negrito, e cada elemento de tupla em que o termo aparece é mostrado em estilo sombreado.

De maneira genérica, seja a sequência de termos  $s$  referente aos termos presentes nas tuplas do resultado de uma consulta a BD; seja  $f_s(t)$  a frequência do termo  $t$  na sequência  $s$ , e  $|s|$  a soma de todas as ocorrências de termos em  $s$ . Dessa forma, define-se

a probabilidade do termo  $t$  na sequência  $s$ , chamada aqui de  $P_s(t)$ , através da Equação 4.1:

$$P_s(t) = \frac{f_s(t)}{|s|}$$

#### **Equação 4.1 - Probabilidade de $t$ na sequência $s$**

De maneira análoga, mas agora considerando o espaço amostral  $E$  referente aos elementos das tuplas do resultado da consulta, define-se a probabilidade do termo  $t$  nos elementos de  $E$ , denominada  $P_e(t)$ , através da Equação 4.2.

$$P_e(t) = \frac{elem(t)}{|E|}$$

#### **Equação 4.2 - Probabilidade de $t$ nos elementos das tuplas**

Nesta equação, considera-se que a função  $elem(t)$  retorna o número de elementos de  $E$  em que o termo  $t$  aparece, independentemente do número de vezes que o termo venha a ocorrer nestes elementos. Ainda para a Equação 4.2, considera-se  $|E|$  como sendo o total de elementos nas tuplas do resultado da consulta.

A expressão final da função de ordenação de termos de DBFIRE corresponde ao produto das duas probabilidades expostas acima, constituindo o peso de um termo  $t$  para o método, como definido na Equação 4.3:

$$peso(t) = P_s(t)P_e(t)$$

#### **Equação 4.3 - Peso do termo $t$ na ordenação**

É importante notar que a função  $peso$  como descrita acima também pode gerar um alto valor para termos considerados *stopwords* [32, 62], formados por artigos, preposições, conjunções, etc. Esses termos têm também uma grande chance de estar razoavelmente difundidos ao longo das tuplas do resultado de uma consulta. No entanto, *stopwords* apresentam pouco valor semântico, já que como também estão bem distribuídas entre todos os documentos da coleção, não ajudam a separar os documentos relevantes dos irrelevantes durante a busca expandida. Dessa forma, são descartadas dos termos adicionados para expansão.

Os experimentos realizados nos próximos capítulos basearam-se em documentos em língua inglesa, sendo utilizado um conjunto padrão de *stopwords* [32] em inglês.

### **4.2.3 Quantos termos devem ser usados na expansão? Quantas tuplas**

### **devem ser analisadas?**

De modo geral, métodos de expansão de busca trazem ganhos na qualidade dos resultados retornados pelo SRI. No entanto, essa melhora no resultado vem com uma consequência: o tempo de processamento aumenta à medida que se aumenta a sobrecarga da expansão, seja pelo aumento de itens (documentos) processados, seja pela maior quantidade de termos enviada ao SRI. Cabe a quem for utilizar os métodos de expansão, regular a sobrecarga aceitável para o benefício que se pretende: um compromisso entre qualidade e tempo é inevitável.

Com DBFIRE é possível regular a extensão do corpus de expansão (no caso, a quantidade de tuplas), bem como o total de termos adicionados na expansão. De forma análoga a outros métodos, aqui se convencionou chamar esses parâmetros de  $k$  e  $n$ , respectivamente.

O método pode usar qualquer conjunto de  $k$  tuplas para análise, já que, em princípio, todas as tuplas do resultado da consulta são igualmente relevantes; por simplicidade, sugere-se o uso das  $k$  primeiras tuplas na ordem em que forem retornadas pelo SGBD. Já com relação à quantidade de termos para expansão, devem-se usar os  $n$  primeiros em ordem decrescente de seus pesos, conforme Equação 4.3.

Para se ter uma ideia da sobrecarga imposta às buscas, observe-se a Tabela 4.1, a qual mostra o impacto adicionado à busca expandida via DBFIRE em comparação à busca sem expansão para diferentes valores de  $k$  e  $n$ . Os dados têm como base um dos testes para validação do método, detalhados nos Capítulos 6 e 7; neste caso foi computado o tempo total relativo ao processamento de todas as 38 buscas ao SRI para uma das coleções de teste da avaliação<sup>10</sup>.

A tabela mostra o tempo em valores absolutos (segundos e/ou minutos) e a sobrecarga adicionada em termos percentuais. Para os testes utilizou-se o sistema Indri [44] como SRI.

---

<sup>10</sup> Coleções de teste são apresentadas no Capítulo 5.

	<b>Tempo</b>	<b>Sobrecarga Adicionada pela Expansão</b>
<i>Sem expansão</i>	53s	-
<b><i>k=10, n=10</i></b>	1 min e 36s	81.1%
<b><i>k=30, n=10</i></b>	1 min e 42s	92.4%
<b><i>k=10, n=30</i></b>	5 min e 41s	>500%
<b><i>k=30, n=30</i></b>	5 min e 47s	>500%

**Tabela 4.1 – Sobrecarga adicionada à busca para diversos valores de  $k$  e  $n$**

Vê-se que o aumento no número de tuplas processadas ( $k$ ) tem um impacto bem menor no tempo do que se aumentarmos a quantidade de termos para expansão ( $n$ ), o que, dependendo do seu valor, pode tornar a expansão impraticável.

É importante salientar que a sobrecarga adicionada ocorre em momentos diferentes: o aumento no valor de  $k$  implica uma sobrecarga que ocorre *antes* da busca realizada pelo SRI; já com o aumento de  $n$  afeta-se a busca propriamente dita, pois o SRI recebe mais termos para processar. O impacto total no tempo de busca deve considerar o impacto de cada componente individual. Dessa forma, considerando a Tabela 4.1, para  $k=10$  e  $n=10$  o tempo total de busca aumentaria em 81.1%.

#### 4.2.4 Qual o peso dos termos enviados ao SRI?

Como foi visto no Capítulo 3, atribuir pesos aos termos da expansão tende a minimizar o efeito de *query drift*, ou seja, desvio do foco da busca. Se todos os termos forem tratados com a mesma importância, o desvio tende a ser tão maior quanto mais termos forem adicionados. O ideal é que se adicionem termos à expansão com pesos relativamente menores que os pesos dos termos originais.

DBFIRE regula esse peso baseando-se na fórmula *Rocchio* [74], apresentada no Capítulo 3. No entanto, a versão usada aqui difere bastante. Ela é apresentada na Equação 4.4, a qual define como calcular o peso de um termo na expansão ( $peso_e$ ) em função do seu peso na ordenação, como apresentado na Equação 4.3.

$$peso_e(t) = \begin{cases} 1.0, & \text{se } t \text{ é uma das palavras – chave originais} \\ \beta \frac{peso(t)}{peso_{max}}, & \text{caso contrário} \end{cases}$$

#### Equação 4.4 – Peso do termo $t$ na expansão

Assim como na fórmula *Rocchio*, o peso de  $t$  para efeito da expansão é normalizado com relação ao maior peso encontrado no corpus de expansão ( $peso_{max}$ ). Também como na fórmula *Rocchio*, esse valor normalizado é regulado através do parâmetro  $\beta$ , o qual varia entre 0 e 1, e que determina a importância que se deve dar aos termos da expansão com relação às palavras-chave do usuário. É com relação ao peso das palavra-chave do usuário que DBFIRE difere da fórmula *Rocchio*.

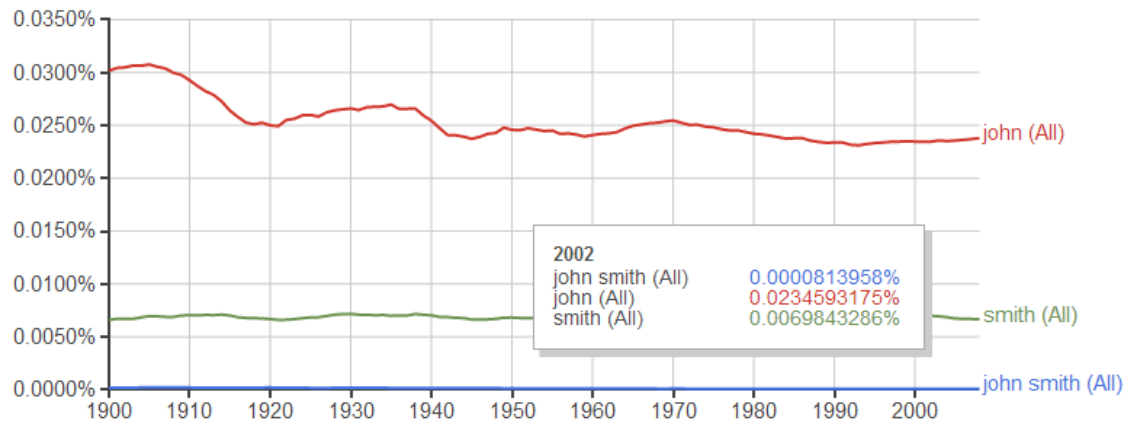
Aqui se considera que todos os termos do usuário têm o mesmo peso (1.0), independente da quantidade de ocorrências na sentença de busca ou no corpus de expansão. Com isso procura-se evitar um possível desvio do foco da busca provocado pela importância maior atribuída a um termo em detrimento de outros devido à sua possível maior frequência dentre as palavras-chave do usuário ou no corpus de expansão. Isso se explica através de dois exemplos.

De início, imagine-se que a consulta a BD faça menção a duas pessoas: *John Smith* e *John Davis*; assim, seria razoável definir as palavras-chave iniciais para essa necessidade de informação como *john smith john davis*. Ao se utilizar a fórmula *Rocchio* original, será atribuído um peso bem maior ao termo *john*, já que ele aparece mais vezes, permitindo um desvio no foco da busca na direção de documentos contendo apenas *john*. Ao se desprezar suas múltiplas ocorrências esse efeito termina sendo minimizado.

Como segundo exemplo, considere-se uma busca pelos termos *john smith* apenas. É bastante provável que em qualquer corpus de expansão, o número de ocorrências do termo *john* ou mesmo do termo *smith*, seja maior do que as ocorrências em conjunto de *john smith*. Exemplificando como isso de fato ocorre, tome-se um corpus disponível on-line, relativo ao projeto *Google Books*<sup>11</sup>. A Figura 4.4 mostra o percentual de livros digitalizados que contêm os termos *john*, *smith* ou *john smith*; o gráfico foi gerado usando a aplicação *Google Ngram Viewer*<sup>12</sup>, filtrando apenas livros em inglês publicados durante os anos 1900-2008.

<sup>11</sup> <http://books.google.com>, acessado em 21/10/2014

<sup>12</sup> <https://books.google.com/ngrams>, acessado em 21/10/2014



**Figura 4.4 - Ocorrências de *john*, *smith* e *john smith* no Google Books (1900-2008)**

Neste caso, o peso do termo *john* é bem maior que o de *smith*, e mais ainda que o de *john smith*, o que deve se refletir no seu peso no corpus de expansão. Aplicando a fórmula *Rocchio* diretamente pode-se criar um desvio da busca expandida em direção a documentos contendo apenas *john* ou apenas *smith* e não *john smith* como era a intenção original. Novamente, ao se considerar que todos os termos das palavras-chave iniciais devem ter o mesmo peso (desprezando seus pesos no corpus de expansão), esse efeito é também reduzido.

Em ambos os exemplos, considerou-se, por simplificação, buscas apenas pelos termos isolados: salienta-se que uma busca por *john smith* permite a recuperação de documentos contendo *john francis smith*, por exemplo. Se o usuário deseja documentos contendo estritamente o nome composto, ele pode assim especificá-lo (como na busca por “*john smith*”) e o nome composto é que será considerado um termo para DBFIRE. Situações como essa sugerem o tratamento de nomes compostos (casos específicos de frases nominais [6, 87]) também para o corpus de expansão; esse tema será retomado como sugestão de trabalhos futuros ao final desta tese.

Por fim, um comentário com relação à definição do parâmetro  $\beta$  presente na Equação 4.4. Normalmente se define um valor fixo a priori dependendo do experimento realizado (há casos relativamente díspares entre si, variando desde 0.5 [60] até 0.1 [68]). Em DBFIRE sugerimos um valor que deve ser adequar à maior parte dos cenários, qual seja  $\beta=0.5$ . Esse valor situa-se aproximadamente no meio do caminho do intervalo máximo possível para  $\beta$ .

No entanto, acredita-se que é possível regular o valor do parâmetro  $\beta$  dependendo da busca, pois  $\beta$  reflete a importância que se deve atribuir aos termos da expansão, e ela



pode variar dependendo do corpus analisado e das palavras-chave iniciais escolhidas. Elaborar uma forma de definir  $\beta$  de maneira automática é um ponto a ser abordado no futuro.

### 4.3 Um exemplo

Seja o exemplo apresentado no Capítulo 2, em que se deseja recuperar os filmes de Martin Scorsese; a Figura 4.5 reapresenta a consulta em SQL para essa necessidade de informação. Fragmentos do resultado da consulta são novamente exibidos na Tabela 4.2, mas agora com destaque para os termos mais frequentes dentro do fragmento (os que possuem pelo menos duas ocorrências).

```
SELECT M.title, M.plot
FROM person as P, directs as D, movie as M
WHERE P.idperson=D.idperson and M.idmovie=D.idmovie
and P.name='Martin Scorsese'
```

**Figura 4.5 - Consulta SQL para os filmes de Martin Scorsese**

Simulando a execução de DBFIRE com os dados exibidos, obtém-se os números exibidos na Tabela 4.3. Cada termo destacado na Tabela 4.2 aparece com sua probabilidade de sequência ( $P_s$ ), probabilidade nos elementos ( $P_e$ ), bem como seu peso em DBFIRE. Os termos aparecem em ordem decrescente de seus pesos. Para o cálculo dessas probabilidades, consideraram-se 68 palavras como o tamanho da sequência, e um total de 6 elementos de tupla.

As palavras-chave definitivas enviadas ao SRI são exibidas abaixo, incluindo seus pesos conforme a Equação 4.4 **Erro! Fonte de referência não encontrada.** Considerou-se  $k=3$ ,  $n=7$  e  $\beta=0.5$ , bem como os termos “martin scorsese movies” como palavras-chave iniciais a serem expandidas. A sintaxe usada aqui é a do SRI Indri<sup>13</sup>, usado nos experimentos do Capítulo 7, na qual os pesos são especificados antes dos termos de busca. Outros SRIs podem usar sintaxes diferentes<sup>14</sup>.

*1.0 martin 1.0 scorsese 1.0 movies 0.5 new 0.5 york 0.321 shine 0.321 light 0.142  
rolling 0.142 stones 0.142 america*

<sup>13</sup> <http://www.lemurproject.org/indri.php>, acessado em 21/10/2014

<sup>14</sup> Para recurso semelhante no SRI Google, veja <https://developers.google.com/custom-search/docs/refinements> (acessado em 17/12/2014)

M.title	M.plot
gangs of <b>new york</b> (2002)	1863. <b>america</b> was born in the streets. in this movie, we see amsterdam vallon returning to the five points of <b>america's</b> , ...
<b>new york, new york</b> (1977)	the day wwii ends, jimmy, a selfish and smooth-talking musician, meets francine, a lounge singer...
<b>shine a light</b> (2008)	martin scorsese and the <b>rolling stones</b> unite in " <b>shine a light</b> ," a look at the <b>rolling stones</b> ...

Tabela 4.2 - Fragmentos da consulta sobre filmes de Martin Scorsese

$t$	$P_s(t)$	$P_e(t)$	$peso(t) = P_s(t) P_e(t)$
new	$\frac{3}{68} = 0.044$	$\frac{2}{6} = 0.333$	0.014
york	$\frac{3}{68} = 0.044$	$\frac{2}{6} = 0.333$	0.014
shine	$\frac{2}{68} = 0.029$	$\frac{2}{6} = 0.333$	0.009
light	$\frac{2}{68} = 0.029$	$\frac{2}{6} = 0.333$	0.009
rolling	$\frac{2}{68} = 0.029$	$\frac{1}{6} = 0.166$	0.004
stones	$\frac{2}{68} = 0.029$	$\frac{1}{6} = 0.166$	0.004
america	$\frac{2}{68} = 0.029$	$\frac{1}{6} = 0.166$	0.004

Tabela 4.3 - Probabilidades e peso para os termos destacados na Tabela 4.2

Ao executar essa busca, obtém-se o resultado mostrado na Tabela 4.4, a qual contém os primeiros 10 links retornados. O primeiro link é a própria página sobre Martin Scorsese no IMDB, a qual contém todos os filmes que ele dirigiu. Os demais links são filmes dirigidos por ele, além de um link para Thelma Schoonmaker. Uma visita à página de Thelma Schoonmaker<sup>15</sup> no IMDB revela que ela fez a montagem de todos os filmes de Scorsese desde *Raging Bull*, em 1980. Dessa forma, pode-se considerar que todos os

<sup>15</sup> <http://www.imdb.com/name/nm0774817>, acessada em 15/12/2014

links retornados são relevantes para a necessidade de informação.

<b>Título</b>	<b>Link</b>
Martin Scorsese	<a href="http://www.imdb.com/name/nm0000217">http://www.imdb.com/name/nm0000217</a>
Mean Streets (1973)	<a href="http://www.imdb.com/Title?Mean Streets (1973)">http://www.imdb.com/Title?Mean Streets (1973)</a>
Raging Bull (1980)	<a href="http://www.imdb.com/Title?Raging Bull (1980)">http://www.imdb.com/Title?Raging Bull (1980)</a>
The Age of Innocence (1993)	<a href="http://www.imdb.com/Title?The Age of Innocence (1993)">http://www.imdb.com/Title?The Age of Innocence (1993)</a>
Taxi Driver (1976)	<a href="http://www.imdb.com/Title?Taxi Driver (1976)">http://www.imdb.com/Title?Taxi Driver (1976)</a>
Goodfellas (1990)	<a href="http://www.imdb.com/Title?Goodfellas (1990)">http://www.imdb.com/Title?Goodfellas (1990)</a>
After Hours (1985)	<a href="http://www.imdb.com/Title?After Hours (1985)">http://www.imdb.com/Title?After Hours (1985)</a>
Thelma Schoonmaker	<a href="http://www.imdb.com/name/nm0774817">http://www.imdb.com/name/nm0774817</a>
Shine a Light (2008)	<a href="http://www.imdb.com/Title?Shine a Light (2008)">http://www.imdb.com/Title?Shine a Light (2008)</a>
New York, New York (1977)	<a href="http://www.imdb.com/Title?New York, New York (1977)">http://www.imdb.com/Title?New York, New York (1977)</a>

**Tabela 4.4 - 10 primeiros links da busca expandida**

Percebe-se, no entanto, que os links para os filmes que estavam presentes no resultado da consulta ficam bem no final da lista; dentre eles, *Gangs of New York* não foi retornado. No entanto, os filmes acima deles têm muito a ver com Nova York: na verdade, *todos* se passam lá.

Pode-se dizer que houve um desvio no foco da busca em direção a filmes de Martin Scorsese que se relacionam com Nova York, devido ao peso maior dado aos termos *new* e *york* na sentença expandida. Neste caso, apesar do desvio a necessidade de informação ainda foi satisfeita. Será que isso se repetiria em outras situações? É o que será abordado ao longo dos procedimentos para validação de DBFIRE.

# Capítulo 5 - Avaliação via Coleções de Teste

## 5.1 Introdução

Como validar o método aqui proposto? Em vários pontos ao longo desta tese menciona-se que DBFIRE consegue recuperar *documentos relevantes à necessidade de informação presente em uma consulta a BD*. Assim, entende-se que DBFIRE deve ser avaliado com relação ao resultado que produz, ou seja, pela lista de documentos que ele consegue, indiretamente<sup>16</sup>, recuperar. E é com essa “régua” que se deve avaliar sua qualidade quando comparada a outros métodos.

Ao se ligar a qualidade do método à qualidade da lista de documentos retornada, é possível avaliar DBFIRE a partir do uso de metodologias bem estabelecidas na área de avaliação de sistemas de RI, utilizando coleções de teste [88, 89, 91]. Uma coleção de teste é definida por um conjunto de documentos, uma lista de tópicos (os quais definem as diferentes buscas que devem ser feitas pelos sistemas avaliados) e um conjunto de análises de relevância, as quais determinam qual documento é relevante para cada tópico.

Analisar se um documento é relevante para um tópico é uma tarefa inerentemente subjetiva, e, portanto, realizada por humanos. Para lidar com essa subjetividade, normalmente se leva em conta a opinião de mais de um “juiz”, usando um número ímpar de “juizes” em caso de empate; a determinação da relevância é feita através de votação por maioria. É fácil ver que, considerando o volume de análises necessário para se criar uma coleção minimamente usável, essa é a etapa mais custosa e demorada na construção de uma coleção de teste.

Do ponto de vista de DBFIRE, o ambiente ideal de validação seria uma coleção de

---

<sup>16</sup> A recuperação dos documentos de fato é feita pelo SRI, não por DBFIRE.

teste relativa a alguns dos domínios a que o método se propõe, como explicitado no Capítulo 1 (como, por exemplo, o domínio jurídico ou farmacêutico). A partir das consultas mais frequentes dentro do domínio escolhido, estabelecer qual a necessidade de informação relativa a cada uma, codificando-a como um tópico; o tópico teria detalhes sobre o que considerar relevante ou não para aquela necessidade de informação. Por fim, seria preciso efetuar as análises de relevância dos documentos retornados por cada busca, tópico a tópico. Esse ambiente seria o ideal, mas fazê-lo demandaria um montante de recursos e tempo inaceitáveis para o escopo desta tese.

Como consequência, optou-se por usar coleções de teste públicas, as quais têm sido geradas em workshops ao estilo TREC (*Text Retrieval Conference* [92]). Neste capítulo serão abordados os principais aspectos dessas coleções e que se relacionam mais diretamente com os experimentos para a validação de DBFIRE.

## 5.2 Julgamentos Incompletos

Dados um conjunto de documentos, um conjunto de tópicos e um conjunto de avaliações de relevância para os documentos em cada tópico é então possível medir a qualidade do resultado de qualquer sistema de RI, considerando a coleção, os tópicos e as relevâncias. Um sistema de RI seria tão melhor quanto mais suas avaliações de relevância dos documentos recuperados estivessem em sintonia com as avaliações dos especialistas. O problema é que normalmente não é factível realizar análises de relevância para *todos* os pares documento/tópico de uma coleção.

Como exemplo, sejam as coleções usadas para os experimentos com DBFIRE: cada uma tem acervos da ordem de milhões de documentos. Um ambiente de avaliação completa iria requerer julgar todos os documentos para todos os tópicos, e por pelo menos dois “juízes” diferentes; em caso de duas avaliações opostas, um terceiro juiz seria necessário para o desempate. Fica claro que seria impraticável.

No entanto, dado um tópico, é bastante provável que pouquíssimos documentos estarão realmente relacionados a ele; sendo assim, apenas uma pequena fração é que necessita de fato ser julgada. Mas como escolher esta fração? A resposta para isso é a técnica de *pooling* [48, 49].

Os documentos a serem julgados para um determinado tópico devem ser provenientes dos primeiros  $X$  documentos retornados por um conjunto de sistemas diferentes; esses sistemas formam o chamado *pool* de avaliação, daí o nome da técnica.

Nas conferências TREC o valor de  $X$  é normalmente definido por volta de 100, pois para a maior parte dos sistemas, até essa profundidade a maioria dos documentos relevantes que seriam detectados já terá sido retornada [92].

Dessa forma, cada sistema contribui com até 100 documentos, que são colocados no *pool* para serem julgados; documentos duplicados são removidos, sendo o *pool* definitivo formado por cerca de 1000 documentos. Os documentos são ordenados por um identificador apenas, de forma que um avaliador não tenha como saber a partir de qual sistema aquele documento foi gerado.

Uma série de experimentos [98] confirmou que essa forma de seleção de documentos assegura justiça na avaliação dos sistemas pertencentes ao *pool*. No entanto, é necessário cautela para a reutilização da coleção ao se avaliar outros sistemas. Neste caso, é possível que eventuais documentos relevantes retornados por esses sistemas não tenham tido sequer a chance de serem julgados. Isso pode lhes gerar um viés desfavorável com relação aos sistemas participantes das sessões de avaliação [14]: por definição, documentos não julgados são considerados não relevantes.

Esse fator é importante na avaliação de DBFIRE, especialmente ao compará-lo a sistemas que fizeram parte dos *pools* de avaliação quando da criação das coleções de teste aqui usadas. O viés devido a julgamentos incompletos pode ser contornado através de métricas que não considerem automaticamente documentos não-julgados como irrelevantes. Tais métricas fazem parte da discussão a seguir.

### **5.3 Métricas de Avaliação**

As medidas acerca da qualidade dos resultados de um sistema de RI derivam de duas noções básicas na área de RI: precisão – percentual de documentos relevantes dentre os documentos retornados – e revocação<sup>17</sup> – fração relativa ao conjunto de documentos relevantes que foram efetivamente retornados. Essas métricas consideram os resultados das buscas como conjuntos, ou seja, grupos não ordenados de documentos. No entanto, se tomarmos dois sistemas diferentes que retornem a mesma quantidade de documentos relevantes, terá mais qualidade aquele que retornar mais documentos relevantes nas primeiras posições. Assim, ao se medir a precisão e a revocação de um sistema são necessárias métricas que levem em conta listas de documentos ordenadas em função de

---

<sup>17</sup> Também conhecido pelo termo inglês *recall*

sua relevância.

As subseções seguintes mostram detalhes de algumas dessas métricas, as quais foram usadas nos experimentos realizados para a avaliação de DBFIRE no Capítulo 7.

### 5.3.1 MAP – Mean Average Precision

A métrica MAP é definida como sendo a média aritmética da métrica AP (*Average Precision*) considerando todos os tópicos da coleção. A métrica AP, por sua vez é aplicada na avaliação de um tópico isolado, sendo definida de acordo com a Equação 5.1.

$$AP = \frac{\sum_{i=1}^N P(i) rel(i)}{R}$$

#### Equação 5.1- Cálculo de AP

Considera-se que  $N$  é o número de documentos retornados pelo sistema,  $i$  é a posição do documento na lista retornada,  $rel(i)$  equivale a 0 ou 1, dependendo da análise de relevância do  $i$ -ésimo documento retornado,  $P(i)$  é o valor da precisão tomando como base os primeiros  $i$  documentos e  $R$  é o número total de documentos relevantes para o tópico.

Para ilustrar seu funcionamento, seja uma lista com 3 documentos A, B e C. Suponha-se que apenas o documento C seja relevante para um determinado tópico, e que se deseja medir o valor de AP para duas ordenações de documentos retornadas por um sistema S,  $S_1 = \{A, B, C\}$  e  $S_2 = \{B, C, A\}$ . Dessa forma, o valor de AP para as duas ordenações seria calculado como segue:

$$AP(S_1) = \frac{P(1)rel(A) + P(2)rel(B) + P(3)rel(C)}{1} = \frac{0 + 0 + \frac{1}{3}}{1} = 0.33$$

$$AP(S_2) = \frac{P(1)rel(B) + P(2)rel(C) + P(3)rel(A)}{1} = \frac{0 + \frac{1}{2} + 0}{1} = 0.5$$

#### Equação 5.2 - Cálculo de AP para $S_1$ e $S_2$

Notar que a métrica recompensa sistemas que retornem documentos relevantes em posições iniciais da lista; como no caso acima,  $S_2$  retorna o mesmo documento relevante que  $S_1$ , mas seu valor de AP é maior pelo fato de tê-lo feito numa posição mais adiante.

A métrica MAP (média de AP) é muito usada em RI, sendo que vários estudos apontam forte correlação entre ela e outras métricas [7, 13, 83]. Para avaliações de

propósito geral ela é considerada a melhor escolha [13].

### 5.3.2 *Bpref* – Binary Preference

Essa é uma das métricas usadas para tratar julgamentos incompletos. Ela reflete a preferência em se retornar documentos julgados relevantes à frente daqueles julgados irrelevantes. Ela é definida na Equação 5.3, considerando sua aplicação a um tópico isolado; para o cálculo de *Bpref* sobre todos os tópicos, efetua-se a média de seus valores tópico a tópico.

$$Bpref = \frac{1}{|R|} \sum_{r \in R} \left(1 - \frac{|\{c \in N', c \text{ retornado à frente de } r\}|}{\min(|R|, |N|)}\right)$$

#### Equação 5.3 - Cálculo de *Bpref*

Nesta equação,  $R$  é o conjunto de documentos julgados relevantes,  $N$  é o conjunto dos documentos julgados irrelevantes,  $r$  representa cada um dos documentos em  $R$ , enquanto  $c$  faz parte de  $N'$ , conjunto formado pelos primeiros  $R$  documentos julgados irrelevantes e retornados pelo sistema a ser medido. Dessa forma, o cálculo de *Bpref* depende da quantidade de elementos do conjunto  $N'$  que foram retornados à frente de elementos de  $R$ .

Como ilustração, seja o exemplo mencionado na seção anterior, com a lista de documentos A, B, C, e duas ordenações relativas a um sistema S,  $S_1 = \{A, C, B\}$  e  $S_2 = \{B, C, A\}$ . Neste caso, considere-se que C é marcado como relevante, A não possui avaliação, e B é marcado como irrelevante. Assim, *Bpref* seria calculado como segue:

$$Bpref(S_1) = \frac{1}{1} \left(1 - \frac{1}{1}\right) = 0$$

$$Bpref(S_2) = \frac{1}{1} \left(1 - \frac{0}{1}\right) = 1$$

#### Equação 5.4 - Cálculo de *Bpref* para $S_1$ e $S_2$

Notar que o valor de AP para ambas as ordenações  $S_1$  e  $S_2$  seria o mesmo – 0.5. Por outro lado, se usarmos *Bpref* há diferenças: *Bpref* foi mais rígida que AP para a ordenação  $S_1$ , e o contrário acontece com  $S_2$ . Qual é a visão correta?

O ideal é usar as duas métricas, especialmente quando se precisa comparar sistemas que não tiveram a chance de participar do *pool* de avaliação de uma coleção, e que por isso podem retornar muitos documentos não-julgados.



Normalmente,  $Bpref$  é fortemente correlacionada a AP [14]. Logo, conclusões discrepantes ao se usar as duas métricas seriam indicativos de um viés favorável aos sistemas participantes do *pool*.

### 5.3.3 Relacionando Precisão e Revocação

Normalmente não se considera medir a qualidade de um sistema tomando sua revocação isoladamente. Isso porque à medida que se aumenta a revocação, a precisão tende a diminuir [62]; no limite, para se ter 100% de revocação bastaria retornar *todos* os documentos da coleção, o que deixaria a precisão no seu valor mínimo! Assim, o comum é partir de um nível de revocação e determinar a precisão do sistema para aquele nível.

Uma métrica útil para isso é a precisão interpolada em 11 pontos de revocação. Dados 11 pontos pré-definidos (começando de 0.0 e indo até 1.0, em incrementos de 0.1), deve-se medir qual a precisão naquele ponto de revocação. O cálculo é feito através de interpolação pois os diferentes valores para a revocação de um sistema nem sempre seguirão a mesma escala definida pelos pontos pré-definidos.

A interpolação é calculada da seguinte forma: dado um nível de revocação  $r$ , a precisão interpolada é a maior precisão encontrada para os demais níveis  $r'$ , com  $r' \geq r$ . Isso é o que mostra a equação a seguir:

$$p_{interp}(r) = \max(p(r')), r' \geq r$$

#### Equação 5.5 - Precisão interpolada

A métrica é definida em termos de um único tópico, mas pode ser aplicada à coleção completa tomando como base sua média relativa a todos os tópicos. Quando consideradas as médias para todos os 11 pontos, é possível traçar um gráfico que relaciona o quanto a precisão do sistema se degrada à medida que sua revocação aumenta.

Como exemplo, seja a Tabela 5.1 contendo uma lista com os primeiros 10 documentos retornados por um sistema fictício. Na tabela,  $i$  equivale à posição do documento, e  $Rel(i)$  pode ser S ou N indicando se o  $i$ -ésimo documento é relevante ou não; e, por fim,  $P(i)$  e  $R(i)$  indicam a precisão e revocação em  $i$  documentos, respectivamente.

Seguindo a regra da Equação 5.5, o cálculo da precisão interpolada é apresentado na Tabela 5.2, montando-se então o gráfico exibido na Figura 5.1. No gráfico é possível ver a tendência de diminuição da precisão com o aumento da revocação.

<b>I</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>Rel(i)</b>	S	N	N	S	S	N	N	N	S	N
<b>P(i)</b>	1/1 = 1	1/2 = 0.5	1/3 = 0.33	2/4 = 0.5	3/5 = 0.6	3/6 = 0.33	3/7 = 0.42	3/8 = 0.37	4/9 = 0.44	4/10 = 0.4
<b>R(i)</b>	1/4 = 0.25	1/4 = 0.25	1/4 = 0.25	2/4 = 0.5	3/4 = 0.75	3/4 = 0.75	3/4 = 0.75	3/4 = 0.75	4/4 = 1	4/4 = 1

Tabela 5.1 - Precisão/revocação para uma lista fictícia de documentos

<b>Revocação</b>	<b>0.0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>	<b>0.6</b>	<b>0.7</b>	<b>0.8</b>	<b>0.9</b>	<b>1.0</b>
<b>Precisão</b>	1.0	1.0	1.0	0.6	0.6	0.6	0.6	0.6	0.44	0.44	0.44

Tabela 5.2 - Precisão interpolada para os dados da Tabela 5.1

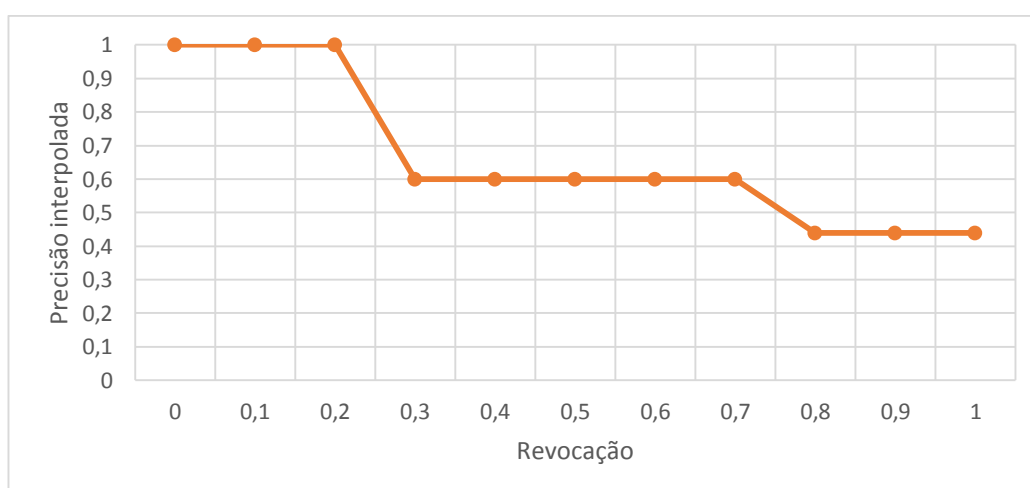


Figura 5.1 - Precisão interpolada versus revocação para a Tabela 5.2

## 5.4 Testes de Significância

As métricas aqui referenciadas refletem médias relativas a medições feitas no conjunto dos tópicos de uma coleção. No entanto, os valores dessas medições podem variar bastante *entre* tópicos. Dessa forma, junto com o valor das médias das métricas de avaliação, é comum acrescentar testes de significância quando se dispõe das distribuições tópico a tópico das métricas dos sistemas.

Testes de significância por sua vez constituem uma ferramenta estatística muito

popular utilizada em uma variedade de áreas de conhecimento. Vários autores propõem sua utilização também na avaliação em RI, como forma de conferir maior poder para as conclusões tiradas numa avaliação.

Os testes de significância são na verdade testes de hipótese [79]. A hipótese a ser testada (normalmente chamada de hipótese nula, ou  $H_0$ ) é a de que os sistemas avaliados possuem características semelhantes entre si e que eventuais diferenças devem ser atribuídas ao acaso. Dessa forma, os testes de significância estimam a probabilidade  $p$  de que  $H_0$  ocorra.

Apesar de não ser uma regra rígida, é comum considerar que as diferenças observadas sejam significativas para valores de  $p$  menores que 0.05, invalidando  $H_0$ . Caso contrário, não é possível afirmar que haja diferenças significativas entre eles.

Entre os testes mais usados em RI está o teste de postos de *Wilcoxon* [79]. Maiores detalhes sobre o teste podem ser encontrados em uma variedade de livros de Probabilidade e Estatística, com alguns deles voltados a aplicações em Ciência da Computação [47, 84].

Mesmo com a utilização de testes de significância, não é possível concluir que um determinado sistema seja melhor que outro fora do escopo da coleção testada: há relatos de experimentos comparando um determinado conjunto de sistemas que mostram resultados significativos a favor de um dado sistema em uma coleção, mas que passam a ser não significativos com o uso de uma coleção diferente [79]. Mesmo usando a mesma coleção de documentos, é possível que a determinação de qual sistema é melhor venha a mudar, caso o conjunto de tópicos utilizado seja alterado.

Apesar dessas ressalvas, à medida que os mesmos resultados venham a ser observados em coleções diferentes e com novos conjuntos de tópicos, o peso da conclusão aumenta bastante. Foi com esse objetivo que se optou por validar DBFIRE em duas coleções diferentes, refletindo domínios também distintos, as quais serão mostradas no próximo capítulo.

## Capítulo 6 - Ambiente de Testes

A maior parte das coleções de teste para avaliar SRIs são compostas por seus elementos básicos – documentos, tópicos e avaliações de relevância para pares documento/tópico. No entanto, esses elementos não incluem BDs. E para se testar DBFIRE a existência de um BD associado é fundamental.

Assim, optou-se por usar duas coleções de teste públicas geradas durante os workshops INEX (*Initiative for the Evaluation of XML Retrieval* [34, 52]). Essas coleções não vêm acompanhadas de BDs explicitamente, mas ambas contêm bastante informação estruturada, o que permite simular a existência do BD. Na verdade, o objetivo delas é estudar como a informação estruturada pode ser útil à recuperação de documentos, tendo, portanto, bastante relação com o ambiente de DBFIRE.

Uma das coleções refere-se à linha *Data Centric* [93] e será referida ao longo do texto como INEX-DC; ela é formada por arquivos referentes a filmes e personalidades do cinema e da TV encontrados na base de dados do IMDB, nas suas versões em inglês. A informação estruturada encontra-se nos próprios documentos, através de marcações XML.

A outra coleção é referente à linha *Linked Data* [37] e será tratada aqui como INEX-LOD: os documentos são artigos relativos à enciclopédia online Wikipédia<sup>18</sup>. A informação estruturada reside à parte através de bases de conhecimento como YAGO [41] e DBpedia [54]. Assim como na coleção INEX-DC, os arquivos aqui correspondem às suas versões em inglês dos respectivos artigos na Wikipédia.

Para as duas coleções considerou-se a tarefa de busca *ad-hoc* [37, 93], na qual dada uma necessidade de informação, deve-se retornar uma lista de documentos ordenada por

---

<sup>18</sup> <http://en.wikipedia.org>, acessado em 21/10/2014

sua relevância; todos os documentos foram indexados pelo SRI Indri<sup>19</sup>.

Cada ambiente de teste será doravante chamado de *macro-cenário*, pois cada um comporta uma série de outros cenários que serão abordados nos testes do próximo Capítulo. As seções seguintes detalham as adaptações realizadas para utilizar cada uma dessas coleções nos testes com DBFIRE.

## 6.1 Coleção INEX-DC

Essa coleção é formada por arquivos texto publicados no site do IMDB em abril de 2010, os quais foram convertidos em documentos XML fortemente estruturados. Cada arquivo XML é referente a um filme ou a uma pessoa envolvida num filme (como atores, diretores, roteiristas, etc.). Como exemplos de informações de um filme podem-se citar seu título, gêneros em que se enquadra, locais onde foi filmado, trama, entre outras; já para os arquivos referentes a pessoas, entre as informações disponíveis podem-se listar seu nome, data e local de nascimento, biografias, etc.

```

<!ELEMENT movie (title, overview?, cast?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT overview (directors?, writers?, genres?, plot?, year?)>
<!ELEMENT directors (director+)>
<!ELEMENT director (#PCDATA)>
<!ELEMENT writers (writer+)>
<!ELEMENT writer (#PCDATA)>
<!ELEMENT genres (genre+)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT plot (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT cast (actors?)>
<!ELEMENT actors (actor+)>
<!ELEMENT actor (name, character?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT character (#PCDATA)>
<!ELEMENT producers (producer+)>
<!ELEMENT producer (#PCDATA)>
...

```

**Figura 6.1 - Fragmento da DTD sobre filmes**

A Figura 6.1 ilustra fragmentos da definição dos documentos (DTD) sobre filmes, enquanto que a Figura 6.2 mostra um fragmento da DTD sobre pessoas. De posse da coleção, necessita-se de um BD para poder testar DBFIRE de forma adequada. Para isso, dois macro-cenários de testes foram preparados.

O primeiro macro-cenário foi baseado na construção de um BD a partir dos

<sup>19</sup> <http://www.lemurproject.org/indri>, acessado em 21/10/2014

documentos XML da coleção. O segundo macro-cenário utilizou-se da busca estruturada feita diretamente nos arquivos XML da coleção a partir da linguagem NEXI (Narrowed Extended XPath I [85, 86]). Os dois ambientes são detalhados a seguir.

```

<!ELEMENT person (name, overview?, filmography?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT overview (birth_name?, birth_date?, biographies?)>
<!ELEMENT birth_date (#PCDATA)>
<!ELEMENT birth_name (#PCDATA)>
<!ELEMENT biographies (biography+, by+)>
<!ELEMENT biography (#PCDATA)>
<!ELEMENT by (#PCDATA)>
<!ELEMENT filmography (act?, direct?, write?)>
<!ELEMENT act (movie+)>
<!ELEMENT direct (movie+)>
<!ELEMENT write (movie+)>
<!ELEMENT movie (title, year, character?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT character (#PCDATA)>
. . .

```

**Figura 6.2 - Fragmento da DTD sobre pessoas**

### 6.1.1 Macro-Cenário 1: Induzindo um BD a partir dos Documentos

#### XML

A montagem do BD começa pela identificação nas DTDs dos atributos de interesse (*title*, *name*, *plot*, *birth\_date*, ...), relacionamentos 1xN (como *biographies* e *genres*)<sup>20</sup> e relacionamentos NxN (*act*, *direct*, *write*, ...). Um modelo simplificado do BD que foi criado é ilustrado na Figura 6.3. O SGBD utilizado para armazenar os dados foi o MySQL versão 5.6<sup>21</sup>.

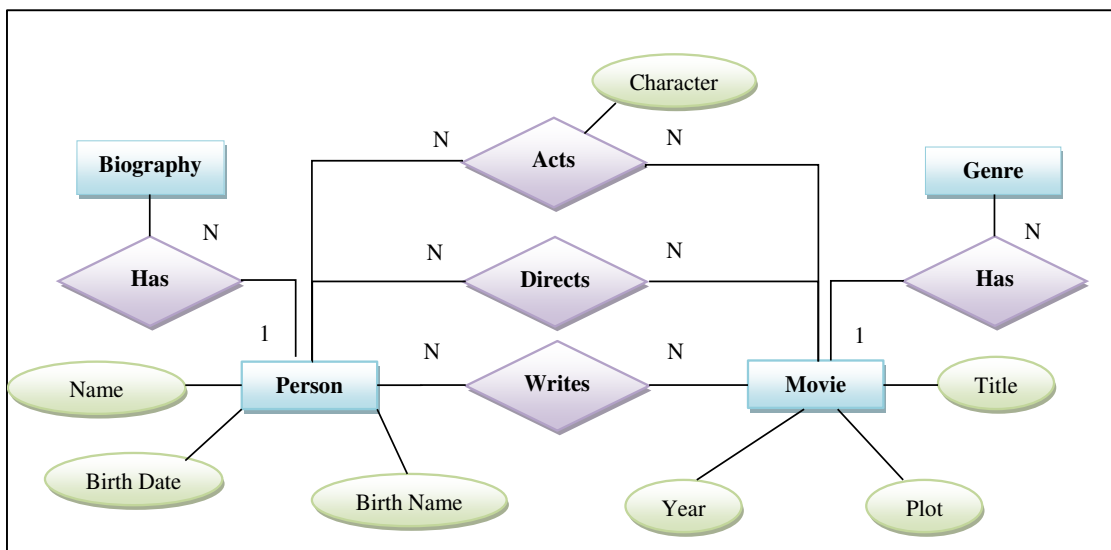
O passo seguinte da montagem do BD foi a extração dos dados dos arquivos XML a partir dos nodos folha contendo pares atributo/valor a serem armazenados em tabelas relacionais. A Figura 6.4 ilustra um documento contendo dados relativos a um filme (*The Departed*) e a Figura 6.5 mostra como esses dados foram carregados no BD.

O último passo foi a construção manual das consultas em SQL que representassem a necessidade de informação referente a cada tópico; as consultas foram criadas de forma *ad-hoc*. Na Figura 6.6 vê-se a descrição de um tópico da coleção, enquanto que a Figura 6.7 apresenta a correspondente consulta em SQL. A coleção completa possui um total de

<sup>20</sup> A rigor, relações como *genres-movie* ou *keywords-movie* deveriam ser NxN, mas para facilitar a implementação foram modeladas como 1xN.

<sup>21</sup> <http://dev.mysql.com/downloads/mysql>, acessado em 21/10/2014

38 tópicos, com uma mediana de 67 documentos julgados relevantes para cada tópico.



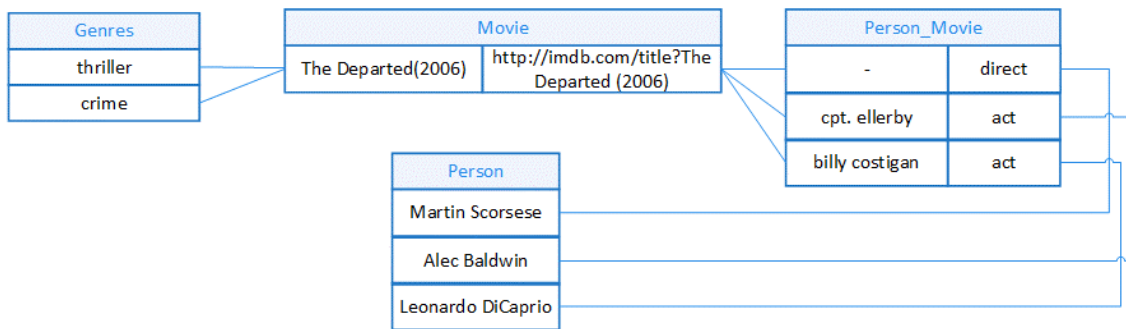
**Figura 6.3 - Diagrama ER relativo aos fragmentos das DTDs INEX-DC**

```

<movie>
  <title>the departed (2006)</title>
  <url>http://www.imdb.com/title?the departed (2006)</url>
  ...
  <directors>
    <director>martin scorsese</director>
  </directors>
  ...
  <genres>
    <genre>crime</genre>
    <genre>thriller</genre>
  ...
  <actors>
    <actor>
      <name>alec baldwin </name>
      <character>cpt. ellerby </character>
    </actor>
    <actor>
      <name>leonardo dicaprio</name>
      <character>billy costigan </character>
    </actor>
  ...</movie>

```

**Figura 6.4 - Arquivo XML para o filme "The Departed"**



**Figura 6.5 – BD após carga dos dados do filme "The Departed"**

```
<topic id="2011104" guid="23">
  <task>AdHoc</task>
  <title>movie Ellen Page thriller</title>
  <castitle>//movie[about(../actor, "Ellen Page") AND
    about(../genre, thriller)]
</castitle>
<description>I want the movies with Ellen Page which are Thrillers
</description>
<narrative>I like Ellen Page as an actress and I respect
  her movie choice but I'm more a thriller person so I
  want to find the
  movies she played which are thriller.
</narrative>
</topic>
```

**Figura 6.6 - Tópico 2011104 da coleção INEX-DC**

```
SELECT DISTINCT *
FROM movie as M, person as P, person_movie as PM, genres as G
WHERE M.idmovie=A.idmovie and PM.idperson=PM.idperson and
  PM.role='act' and G.idmovie=M.idmovie and
  P.name='Ellen Page' and G.genre='Thriller'
```

**Figura 6.7 - Consulta SQL para o tópico 2011104**

Embora a necessidade de informação presente no tópico 2011104 tenha sido facilmente traduzida para SQL, esse não foi o caso para a maioria dos outros tópicos da coleção. Nestes casos, foi necessário usar o recurso de buscas por palavras-chave para recuperar as tuplas do BD. Isso foi possível usando o índice FULL TEXT do MySQL<sup>22</sup>. Na Figura 6.8 ilustra-se um desses tópicos, enquanto na Figura 6.9 exibe-se a consulta SQL que recupera as respectivas tuplas.

<sup>22</sup> <http://dev.mysql.com/doc/refman/5.0/en/fulltext-search.html>



```

<topic id="2011107" guid="18">
  <task>AdHoc</task>
  <title>Tom Hanks biography</title>
  <castitle>//person[about(., Tom Hanks)]</ castitle>
  <description>I want to know more about Tom Hanks
  </description>
  <narrative>As I want to know more about the history of
    cinema in America. Tom Hanks is one of most
    famous actors AND I don't know who he is AND what
    he directed. In order to do so I try to find
    his biography.
  </narrative>
</topic>

```

**Figura 6.8 - Tópico 2011107**

```

SELECT DISTINCT * FROM person as P, biographies as B
WHERE P.idperson=B.idperson and
      match(B.biography) against ("Tom Hanks" in boolean mode)

```

**Figura 6.9 - Consulta SQL para o tópico 2011107**

Note-se que se procurou ao máximo aumentar a precisão dos resultados, em detrimento da revocação. Isso pode ser percebido através do uso do termo “Tom Hanks” entre aspas para o operador *match* do MySQL<sup>23</sup>, evitando que se recupere documentos que contenham apenas um dos termos da busca. Além disso, optou-se pela busca booleana, que se assemelha mais ao comportamento de uma consulta a BD. Mesmo assim, isso minimiza mas não impede a perda da exatidão nos resultados das consultas. Para o tópico 2011107, por exemplo, o fato de encontrarmos o termo “Tom Hanks” num campo biografia não significa que aquela biografia seja necessariamente *sobre* Tom Hanks.

As consultas criadas para a coleção estão disponíveis online<sup>24</sup> assim como no Anexo A ao final desta tese.

### 6.1.2 Macro-Cenário 2: Busca Estruturada nos Documentos XML

Uma vez que os documentos da coleção já estão num formato estruturado (XML), por que não consultá-los numa linguagem que permita a recuperação direta do arquivo XML, sem precisar da montagem do BD em paralelo? Isso é possível pois o SRI escolhido (Indri) possui recursos para se restringir a busca em trechos com marcação XML – a chamada busca estruturada. Ela se baseia na linguagem NEXI, derivada de outra

<sup>23</sup> <http://dev.mysql.com/doc/refman/5.5/en/fulltext-search.html>, acessado em 21/10/2014

<sup>24</sup> <https://sites.google.com/a/copin.ufcg.edu.br/tese/home/queries-for-data-centric-collection>, acessado em 21/10/2014

linguagem para manipulação de arquivos XML, a linguagem *XPath*<sup>25</sup>. Diferente das consultas para o BD induzido que tiveram que ser criadas manualmente, aqui utilizamos as consultas NEXI que já acompanham os tópicos da própria coleção Data Centric (elas estão descritas nas marcações *<casttitle>*; ver exemplos nas Figuras 6.6 e 6.8).

Como os arquivos retornados são os próprios documentos em XML, é necessário convertê-los em tuplas para sua utilização por DBFIRE. Na conversão considerou-se que a cada arquivo XML está associada uma tupla: essa tupla possuirá tantos elementos quantos forem os pares atributo/valor existentes no arquivo de origem. A Figura 6.10 ilustra como o arquivo exemplo mostrado na Figura 6.4 seria convertido numa tupla seguindo a lógica descrita acima.

The Departed(2006)	<a href="http://imdb.com/title?The&lt;br/&gt;Departed (2006)">http://imdb.com/title?The Departed (2006)</a>	crime	thriller	Martin Scorsese	Alec Baldwin	cpt. ellerby	Leonardo DiCaprio	billy costigan
-----------------------	---	-------	----------	--------------------	-----------------	-----------------	----------------------	-------------------

**Figura 6.10 - Tupla referente ao exemplo na Figura 6.4**

Como cada arquivo XML da coleção se refere a uma pessoa ou a um filme, cada tupla convertida segue a mesma lógica. Assim, o valor do parâmetro *k* de DBFIRE determina os primeiros arquivos do resultado a serem processados pelo método durante a expansão.

Uma vez que todo o conteúdo do arquivo é recuperado, junções entre tabelas que no caso de uma consulta a BD precisariam ser feitas explicitamente via SQL já estarão disponíveis diretamente no documento recuperado. Por exemplo, na Figura 6.10 tanto os gêneros do filme, como seu elenco, já aparecem serializados junto com o próprio filme numa única tupla. Dessa forma, valores de campos multivalorados também aparecerão como elementos separados numa mesma tupla (como *crime* e *thriller* na Figura 6.10).

## 6.2 Coleção INEX-LOD

Essa coleção é formada por artigos da versão inglês da Wikipédia, mas possui também conteúdo estruturado, representado por metadados referentes a esses artigos. Os metadados são disponibilizados pelas bases de conhecimento YAGO [41] e DBpedia [54]. Esses dados podem ser consultados através de uma linguagem ao estilo SQL, como por exemplo SPARQL<sup>26</sup>. Seguindo a nomenclatura para os ambientes de teste da coleção

<sup>25</sup> <http://www.w3.org/TR/xpath-30>, acessado em 21/10/2014

<sup>26</sup> <http://www.w3.org/TR/sparql11-query>, acessado em 21/10/2014

INEX-DC, o ambiente para essa coleção será chamado de *macro-cenário 3*.

As bases de conhecimento YAGO e DBpedia são formadas por triplas que seguem o padrão RDF (*Resource Description Framework*<sup>27</sup>) no formato *<recurso, propriedade, valor>*: um *recurso* identifica um artigo na Wikipédia, uma *propriedade* é o equivalente a um atributo no mundo relacional e se refere ao recurso da tripla, enquanto o *valor* é o valor propriamente dito da propriedade para aquele recurso.

A coleção completa é formada por um total de 144 tópicos, dos quais 72 são os chamados tópicos *Jeopardy*, ao estilo dos programas de perguntas e respostas da TV. Dentre estes, os testes de validação de DBFIRE usaram os 50 primeiros por representar um conjunto suficientemente representativo da coleção<sup>28</sup>.

Para cada tópico na coleção foram criadas manualmente consultas no padrão SPARQL, retornando triplas RDF. De forma análoga à criação das consultas para o macro-cenário 1, as consultas SPARQL também foram feitas de maneira *ad-hoc*.

A Figura 6.11 mostra um dos tópicos da coleção enquanto que a Figura 6.12 ilustra a versão da consulta SPARQL correspondente. A título de informação, a coleção apresenta uma mediana de 25 documentos julgados relevantes para cada tópico.

```
<topic id="2013311">
  <title>countries make up Central America</title>
  <description> These countries make up Central America.
  </description>
</topic>
```

**Figura 6.11 - Tópico 2013311 da coleção INEX-LOD**

```
SELECT DISTINCT ?subject ?property ?value WHERE {
  ?subject ?property ?value .
  ?x <http://dbpedia.org/property/data> ?city .
  ?city <http://dbpedia.org/ontology/country> ?subject .
  FILTER regex(?x, "Central", "i") .
  FILTER regex(?x, "America", "i") . }
```

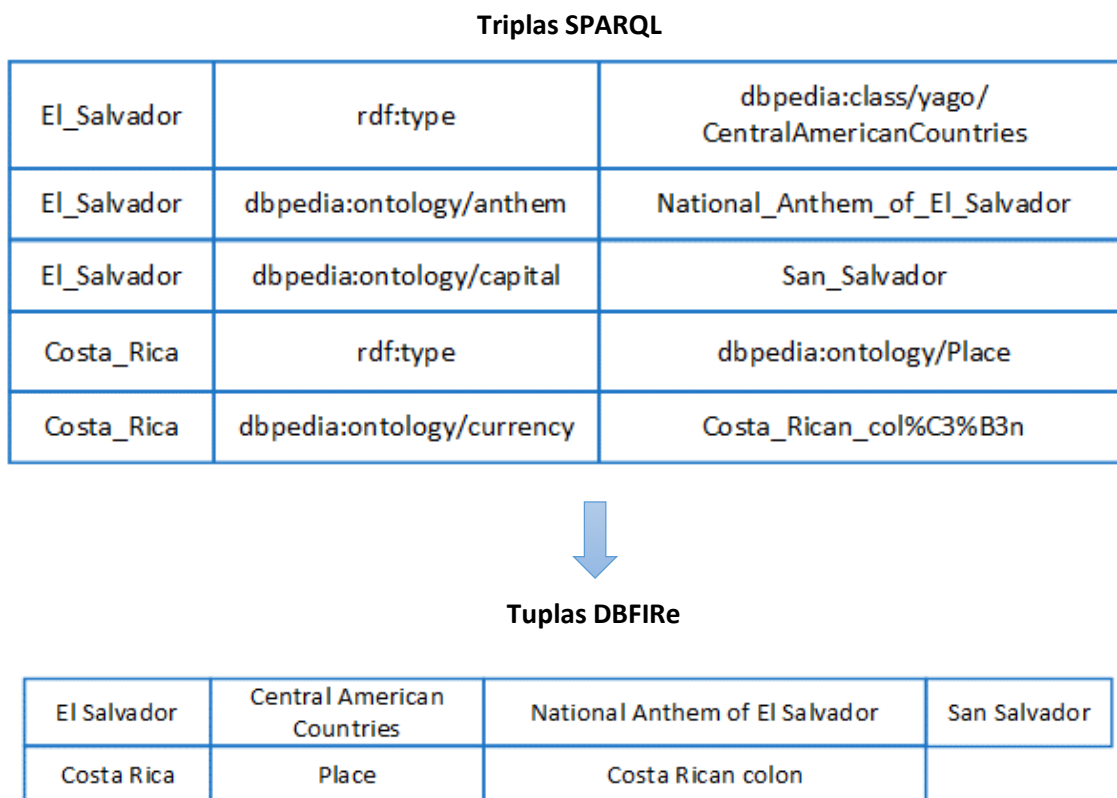
**Figura 6.12 - Versão SPARQL para o tópico 2013311**

Uma vez que se tem as triplas retornadas pela consulta SPARQL, essas são transformadas num conjunto de tuplas, na qual o primeiro elemento corresponde ao

<sup>27</sup> <http://www.w3.org/RDF>, acessado em 21/10/2014

<sup>28</sup> Apesar de a coleção ter mais tópicos, limitamos a 50 devido ao tempo para a construção das consultas SPARQL para todos os tópicos. Além disso, esse é um padrão para o conjunto mínimo de tópicos de uma coleção na área de RI [89].

recurso da tripla, enquanto que os demais elementos são todos os valores de propriedades para aquele recurso. Assim, todas as triplas referentes a um mesmo recurso são serializadas numa única tupla.



**Figura 6.13 - Conversão de triplas SPARQL em tuplas DBFIRE**

Na Figura 6.13 exibe-se um fragmento do resultado da consulta SPARQL para o tópico 2013311 e sua conversão num conjunto de tuplas a serem usadas por DBFIRE. De forma análoga à serialização de arquivos XML discutida na seção 6.1.2, cada tupla pode ter um número variável de elementos, dependendo de quantas triplas RDF tenham sido recuperadas para aquele recurso.

Toda sintaxe referente à taxonomia YAGO/DBpedia é removida antes de se adicionar um item a um elemento de tupla. Da mesma forma que nas consultas para a coleção INEX-DC, todas as consultas SPARQL estão disponíveis online<sup>29</sup> e no Anexo B ao final desta tese. Elas foram executadas através de um explorador SPARQL na Web<sup>30</sup>.

Além da conversão em tuplas, uma adaptação adicional foi necessária para os

<sup>29</sup> <https://sites.google.com/a/copin.ufcg.edu.br/tese/home/queries-for-lod-collection>, acessado em 21/10/2014

<sup>30</sup> <http://dbpedia.org/snorql>, acessado em 21/10/2014

comparativos frente ao método SEMEX [57]. Cada consulta SPARQL foi traduzida num equivalente SQL de forma que o método pudesse ser executado de acordo com suas especificações. Por exemplo, para a consulta relativa à Figura 6.12, sua versão em SQL ficou como especificado na Figura 6.14.

```
SELECT M.country
FROM M AS M
WHERE M.continent='Central America'
```

**Figura 6.14 - Versão SQL para o tópico 2013311**

Note-se o uso de uma tabela fictícia (M), e a suposição de que M possua um atributo chamado *continent*. Essa foi uma das simplificações que fizemos supondo um esquema de dados ideal. As demais consultas SQL para esse macro-cenário podem ser conferidas online<sup>31</sup> e no Anexo C ao final desta tese.

### 6.3 Discussão

A necessidade de se adaptar as duas coleções de teste ao contexto de DBFIRE não deixar de ser um fator negativo para sua avaliação, especialmente nos pontos em que foram necessárias interferências diretas: na confecção manual das consultas ao BD inferido (macro-cenário 1), assim como na criação das consultas SPARQL (macro-cenário 3). Além disso, o fator tempo determinou a criação manual de forma *ad-hoc* das consultas para os macro-cenários 1 e 3; reconhece-se que isso pode configurar um viés nos experimentos. No Capítulo 8 detalhamos como contornar esta questão no futuro.

No entanto, essa situação não seria muito diferente do ambiente para o qual DBFIRE poderá ser usado, pois ele se utiliza de consultas prontas, provavelmente elaboradas por experts nos sistemas organizacionais. De qualquer forma, essas questões são mitigadas no macro-cenário 2, já que se parte de consultas elaboradas por terceiros.

Outra questão importante é que a maior parte das consultas em todos os macro-cenários se utilizam de operadores mais afeitos à RI do que ao mundo BD (índices FULL TEXT nas consultas ao BD inferido a partir dos documentos XML, cláusulas *about* nas consultas NEXI, e mesmo os operadores FILTER nas consultas SPARQL).

Mesmo assim acreditamos que os experimentos são válidos, pois oferecem uma

<sup>31</sup> <https://sites.google.com/a/copin.ufcg.edu.br/tese/home/converting-sparql-into-sql>, acessado em 21/10/2014

visão do funcionamento de DBFIRE frente a outros métodos submetidos às mesmas condições. Apesar disso, vê-se a necessidade de que mais testes venham a ser feitos, seja com mais coleções de teste públicas, ou em ambientes reais de uso.

# Capítulo 7 - Avaliação de DBFIRE

## 7.1 Introdução

Este capítulo descreve os experimentos realizados para avaliar o método DBFIRE. Os experimentos foram agrupados nos 3 macro-cenários descritos no Capítulo 6, dois deles relativos à coleção de teste INEX-DC e o terceiro referente à coleção INEX-LOD. Em cada um desses cenários mais amplos, foi realizada uma série de comparativos entre DBFIRE e métodos concorrentes.

Dadas as possíveis configurações de DBFIRE, dependendo das diferentes combinações dos parâmetros  $k$ ,  $n$  e  $\beta$ , inicialmente fizemos três comparativos avaliando os efeitos da variação de cada parâmetro sobre os resultados das buscas expandidas. Em seguida, realizamos comparativos mais detalhados usando uma configuração fixa para os parâmetros  $k$ ,  $n$  e  $\beta$ , a qual acreditamos ser razoável para a maioria dos ambientes em que vislumbramos a execução de DBFIRE:  $k=10$ ,  $n=10$  e  $\beta=0.5$ . Essa configuração apresentou níveis aceitáveis de sobrecarga (próximo a 80%, como visto no Capítulo 4), mantendo o incremento na qualidade nos resultados. Em particular, para o parâmetro  $\beta$ , acreditamos que o valor de 0.5 seja razoável para a maior parte dos ambientes, situando-se no meio do intervalo (0,1), que compõe os valores limites para o respectivo parâmetro.

Nos comparativos detalhados, usamos as métricas MAP e  $B_{pref}$ , com os respectivos testes estatísticos para confirmação da significância dos resultados. Para cada comparativo, apresentamos também os gráficos de precisão interpolada versus revocação dos sistemas avaliados.

Para cada macro-cenário, 4 grupos básicos de testes foram feitos: testes com um *baseline* (linha base) equivalentes à busca sem expansão; testes com representantes de outros métodos de integração (SCORE [75] e SEMEX [57]); testes frente a outros

métodos de ordenação de termos (KLD [17], DFR [2] e RM [72]); e por fim, testes com os sistemas que participaram dos workshops para a criação das coleções de teste.

Além dos testes básicos, para cada macro-cenário foram feitos outros comparativos, tentando forçar situações mais hostis ao método DBFIRE, como por exemplo: menor número de palavras-chave iniciais do usuário, menos campos disponíveis para expansão, como também o incremento do tamanho da coleção forçando a inclusão de arquivos reconhecidamente não-relevantes.

### **7.1.1 Comparando DBFIRE com um *baseline***

Os comparativos frente ao *baseline* tiveram como objetivo verificar a hipótese de que um método de integração entre BDs e documentos deve ser no mínimo melhor do que uma busca direta a um SRI. Ou seja: supõe-se que o usuário só se submeterá à sobrecarga inerente ao processo de expansão se a busca expandida oferecer melhores resultados do que a busca manual que ele venha a fazer diretamente.

Para isso, simulou-se a busca direta do usuário através das palavras-chave encontradas nos resumos das necessidades de informação de cada coleção de teste; elas estão disponíveis no campo *<title>* de cada tópico, tanto na coleção INEX-DC como na coleção INEX-LOD.

### **7.1.2 Comparando DBFIRE com outros métodos de integração**

#### **SGBD-SRI**

Os comparativos aqui se deram a partir das implementações dos métodos SCORE e SEMEX, conforme detalhado em seus artigos. Algumas adaptações já foram necessárias para adequarmos o ambiente de teste aos métodos (por exemplo, a conversão SPARQL-SQL já discutida na seção 6.2 para a execução de SEMEX no macro-cenário 3). Outras adaptações foram necessárias ao método SCORE para que se adequasse aos ambientes de teste.

Uma vez que SCORE utiliza o conteúdo completo de elementos de tuplas na busca por palavras-chave ao SRI, não há controle sobre o número efetivo de termos enviados, pois cada elemento pode ter um número indeterminado de termos; assim, o tempo de busca pode ficar impraticável. A configuração padrão recomendada pelos autores é de se usar os 10 elementos de melhor escore (parâmetro  $N=10$ , portanto). Isso certamente originará muito mais que os 10 termos usados nos comparativos frente a DBFIRE ( $n=10$ , no nosso caso).



Dessa forma, procuramos limitar o número de termos por elemento que seriam usados nas buscas. Após testarmos vários valores, chegamos ao limite de 10 termos por cada elemento de tupla, considerando sempre no máximo os 10 primeiros de cada elemento e excluindo-se *stopwords* padrão [32]. Esta configuração foi a que rendeu os melhores números para SCORE.

### **7.1.3 Comparando a ordenação de termos de DBFIRE com outros métodos de ordenação**

Este comparativo foi feito usando outras abordagens de expansão automática de buscas apresentadas no Capítulo 3, mas adaptadas ao contexto de integração entre BDs e documentos. Foram escolhidos os métodos KLD, DFR e RM, descritos no Capítulo 3, por serem métodos bem estabelecidos na área de expansão de buscas, e estarem disponíveis em SRIs populares, como Terrier<sup>32</sup> e Indri<sup>33</sup>.

A execução de cada método consistiu em aplicar a mesma infraestrutura de DBFIRE, mudando apenas a forma de ordenação de termos. Dessa maneira, os métodos usam o mesmo conjunto de palavras-chave iniciais, a mesma definição de pesos dos termos para a busca expandida, assim como os mesmos valores para os parâmetros  $k$ ,  $n$  e  $\beta$  definidos na respectiva configuração de DBFIRE. O corpus de expansão é formado pelas tuplas resultantes das consultas ao BD, considerando cada tupla como um documento.

O objetivo aqui é responder à pergunta: será que usando um método qualquer para ordenação de termos obteríamos os mesmos resultados que DBFIRE?

### **7.1.4 Comparando com os sistemas participantes dos workshops**

#### **INEX**

Este último comparativo realizado em cada macro-cenário considerou os resultados de DBFIRE frente aos melhores sistemas participantes dos workshops INEX em cada coleção de teste, os quais fizeram parte dos *pools* de análise de relevância em cada coleção. A ideia aqui é estender o teste com o *baseline*: se o usuário pudesse fazer uma busca por palavras-chave usando os melhores sistemas já testados para aquelas coleções, como se comportaria comparativamente DBFIRE?

Para a coleção INEX-DC selecionamos os 3 melhores dentre os 9 sistemas

---

<sup>32</sup> <http://terrier.org>, acessado em 21/10/2014

<sup>33</sup> <http://www.lemurproject.org/indri>, acessado em 21/10/2014

participantes, usando como parâmetro seus resultados para a métrica MAP [93]. Os sistemas são identificados pelos acrônimos dos nomes das universidades onde foram desenvolvidos: Universidade de Amsterdã (UAMS [33]), Universidade Renmin (RUC [94]) e Universidade Kasetsart (KAS [97]).

Já para a coleção INEX-LOD, a trilha *ad-hoc* teve apenas 3 sistemas participantes, sendo que apenas um deles demonstrou desempenho competitivo. De qualquer forma, foi feito o comparativo de DBFIRE frente a todos os 3. Eles são identificados pelos acrônimos referentes aos centros de pesquisa onde foram desenvolvidos [36]: Universidade Renmin (RUC), Instituto Max Planck de Informática (MPI), e Universidade de Oslo e Akershus (OAUC).

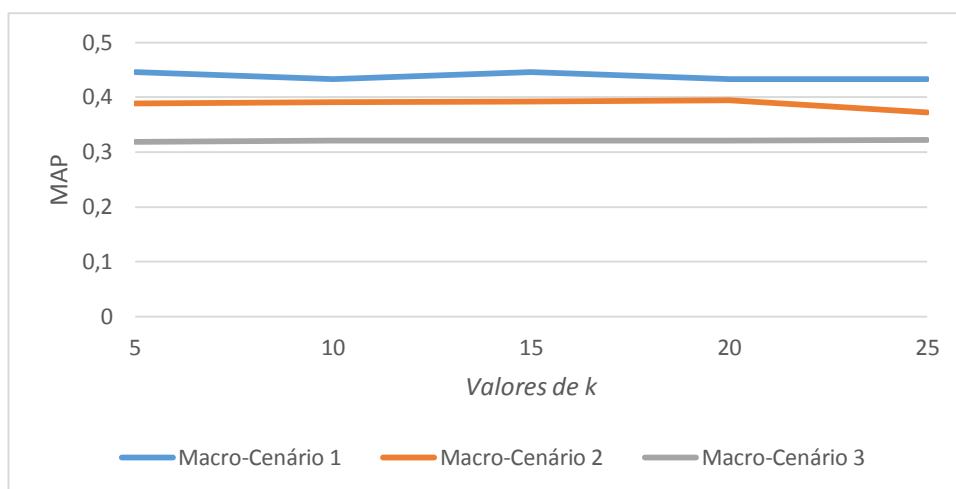
Em todos os casos, utilizamos as distribuições para as métricas AP e *Bpref* constantes nos relatórios sobre cada workshop [37, 93].

## 7.2 Efeitos dos parâmetros $k$ , $n$ e $\beta$ para a expansão

Com o objetivo de dar uma ideia de como os parâmetros de configuração de DBFIRE podem influir em seus resultados, mostramos aqui alguns gráficos que ilustram as alterações no desempenho à medida que se aumentam os valores de cada parâmetro, usando para isso a métrica MAP.

Na literatura não há uma regra explícita para isso, o que faz com que cada autor defina sua configuração empiricamente. Há exemplos [20] de testes usando 5 documentos e 30 termos de expansão (o que equivaleria, respectivamente aos parâmetros  $k$  e  $n$  em DBFIRE), além de outros autores [68] fixando o que seria o equivalente a  $k$  em 10 e variando  $n$  entre 25, 40 e até 75 termos. Em [17] há comparativos variando  $k$  desde 2 até 20, em intervalos de 2 documentos, e  $n$  de 10 até 100, com saltos de 10 termos. Com relação a  $\beta$  há relatos de experimentos usando valores variando desde 0.5 [60] até 0.1 [68]).

No nosso caso, definimos 5 níveis diferentes para cada parâmetro. A configuração de  $k$  e de  $n$  variou de acordo com os valores do conjunto {5, 10, 15, 20, 25} enquanto que os valores de  $\beta$  variaram ao longo do conjunto {0.1, 0.3, 0.5, 0.7, 0.9}. Ao se medir o efeito de um parâmetro, deixamos os demais fixos nos valores que utilizamos nos demais experimentos. Por exemplo, o efeito do parâmetro  $k$  é ilustrado na Figura 7.1; neste caso, os valores de  $n$  e  $\beta$  permaneceram fixos em 10 e 0.5, respectivamente.



**Figura 7.1 - Efeito do parâmetro  $k$**

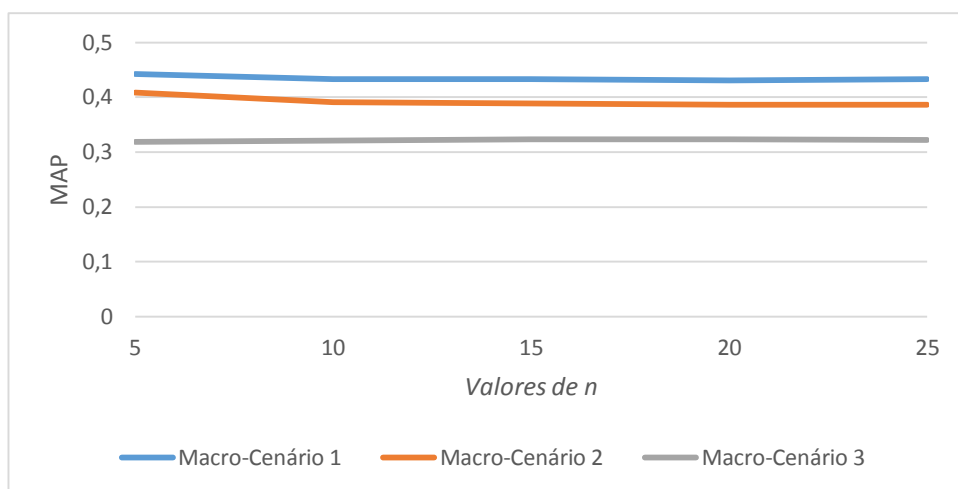
O comportamento de DBFIRE não parece ser muito afetado pela variação de  $k$ , especialmente com relação ao macro-cenário 3. Na verdade, é preciso registrar que neste ambiente são poucos os tópicos que apresentam mais que 5 tuplas; assim, diferentes valores de  $k$  não teriam mesmo como mudar os resultados obtidos. Podemos também confirmar o efeito da variação do parâmetro através da razão entre os valores máximo/mínimo da métrica MAP em cada macro-cenário. Esses dados são exibidos na Tabela 7.1.

	Macro-Cenário 1	Macro-Cenário 2	Macro-Cenário 3
<b>MAP Máximo/ MAP Mínimo</b>	1.031	1.062	1.011

**Tabela 7.1 - Razão Máximo/Mínimo para a métrica MAP devido à variação de  $k$**

Passemos agora ao efeito da variação do parâmetro  $n$  o qual é mostrado na Figura 7.2. Neste caso, os valores de  $k$  e  $\beta$  é que ficaram fixos em 10 e 0.5, respectivamente.

Novamente, vê-se pouquíssima alteração nos resultados com o incremento de  $n$ . De certa forma, isso faz sentido pois cada novo termo incluído na expansão vai ter um peso sempre menor que os anteriores; no limite, novos termos devem influenciar muito pouco os resultados. De forma análoga à discussão sobre o efeito do parâmetro  $k$ , a Tabela 7.2 ilustra a razão máximo/mínimo para a métrica MAP devida à variação do parâmetro  $n$ . Se compararmos com os números na Tabela 7.1, podemos ver que o efeito do parâmetro  $n$  foi ainda menor que aquele devido à variação de  $k$ .

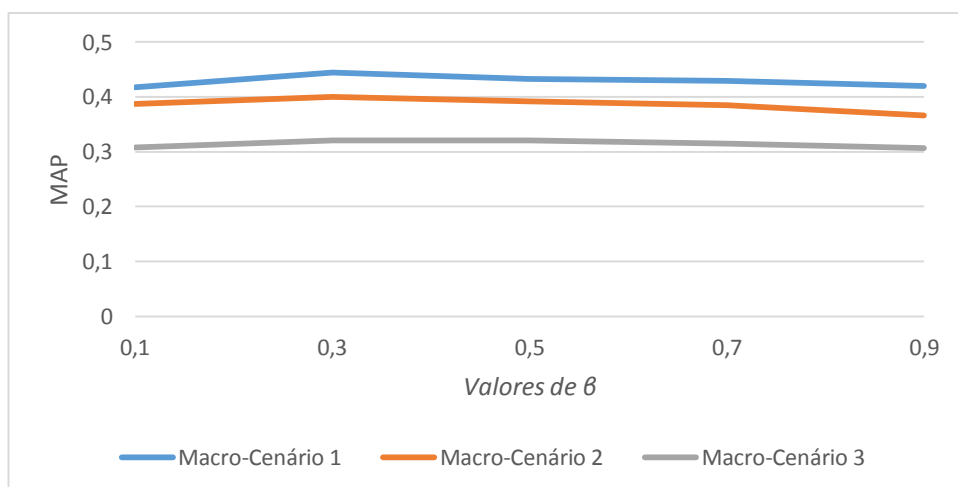


**Figura 7.2 - Efeito do parâmetro  $n$**

	Macro-Cenário 1	Macro-Cenário 2	Macro-Cenário 3
MAP Máximo/ MAP Mínimo	1.026	1.057	1.01

**Tabela 7.2 - Razão Máximo/Mínimo para a métrica MAP devido à variação de  $n$**

Por fim, o efeito da alteração do parâmetro  $\beta$  é ilustrado na Figura 7.3. Apesar de seu efeito ser também pequeno, a mudança no valor de  $\beta$  tem impacto maior na expansão do que as mudanças nos valores dos demais parâmetros, ocorrendo mais especificamente nos valores mínimo (0.1) e máximo (0.9). Dessa forma, podemos supor que o peso máximo dos termos na expansão nem deve ser muito baixo (menor que 0.3) nem muito alto (maior que 0.7). Vemos a importância maior desse parâmetro se compararmos a relação máximo/mínimo para a métrica MAP com os números na Tabela 7.1, na Tabela 7.2 e na Tabela 7.3.



**Figura 7.3 - Efeito do parâmetro  $\beta$**

	Macro-Cenário 1	Macro-Cenário 2	Macro-Cenário 3
MAP Máximo/ MAP Mínimo	1.064	1.090	1.045

**Tabela 7.3 - Razão Máximo/Mínimo para a métrica MAP devido à variação de  $\beta$**

Dada a pequena influência desses parâmetros na busca expandida, pode-se pensar que a escolha de qual valor exato a se usar pode ser aleatória. No entanto, levamos em conta dois fatores para a escolha que fizemos para os testes detalhados.

Em primeiro lugar, temos a questão da sobrecarga, especialmente quando se acrescenta muitos termos na expansão (este fator afeta os parâmetros  $k$  e  $n$ ): acreditamos que o valor de 10 para cada parâmetro seja razoável para equilibrar qualidade e tempo. Por fim, o peso máximo dos termos na expansão também deve ser variável, dependendo do domínio em questão, ou mesmo variando de tópico a tópico. Daí que um valor a meio do caminho entre os limites máximo e mínimo ( $\beta=0.5$ ) parece uma escolha razoável.

### 7.3 Comparativos no Macro-Cenário 1

Esta seção detalha os resultados dos experimentos realizados no primeiro macro-cenário de testes, usando como corpus de expansão os resultados de consultas realizadas num BD alimentado pelos documentos XML da coleção INEX-DC.

Inicialmente reportamos os comparativos básicos de DBFIRE frente a outros métodos, e em seguida alteramos o ambiente de testes para simular algumas situações que podem influenciar no desempenho de DBFIRE.

#### 7.3.1 Comparativos no ambiente padrão de testes

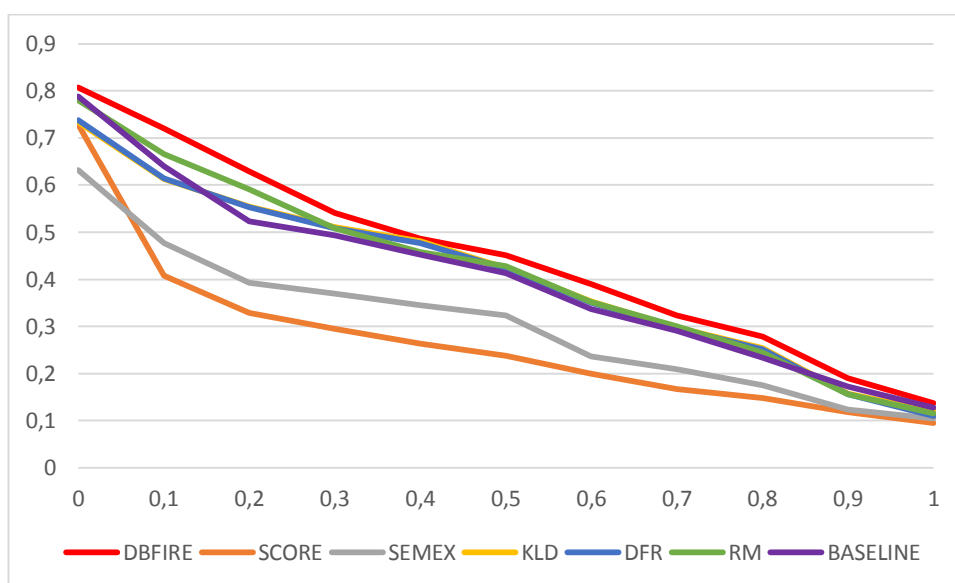
A Tabela 7.4 mostra os valores para as métricas MAP e  $B_{pref}$  para o método DBFIRE e os métodos já mencionados no início deste capítulo (*baseline*, SCORE, SEMEX, KLD, DFR, RM). Apenas reforçando o que já foi mencionado na seção anterior, as configurações dos parâmetros  $k$ ,  $n$  e  $\beta$ , válidos para a execução de DBFIRE, KLD, DFR e RM foram fixadas em  $k=10$ ,  $n=10$  e  $\beta=0.5$ .

Na tabela vê-se também as diferenças percentuais em favor de DBFIRE com relação aos resultados de cada método, assim como o respectivo *p-valor* mostrando o nível do teste estatístico feito para confirmar essas diferenças.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.4326	0.3790	0.2035	0.2840	0.3894	0.3869	0.3999
<b>Diferença</b>	-	<b>14.1%</b>	<b>&gt;100%</b>	<b>52.3%</b>	<b>11.1%</b>	<b>11.8%</b>	<b>8.18%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<i>Bpref</i>	0.4795	0.4120	0.2578	0.3678	0.4633	0.4636	0.4653
<b>Diferença</b>	-	<b>16.3%</b>	<b>86.0%</b>	<b>30.4%</b>	<b>3.5%</b>	<b>3.4%</b>	<b>3.0%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.06	0.16	0.45

**Tabela 7.4 – Comparativo no ambiente padrão de testes**

Como ilustração do desempenho dos métodos, exibimos o gráfico de precisão interpolada versus revocação na Figura 7.4.

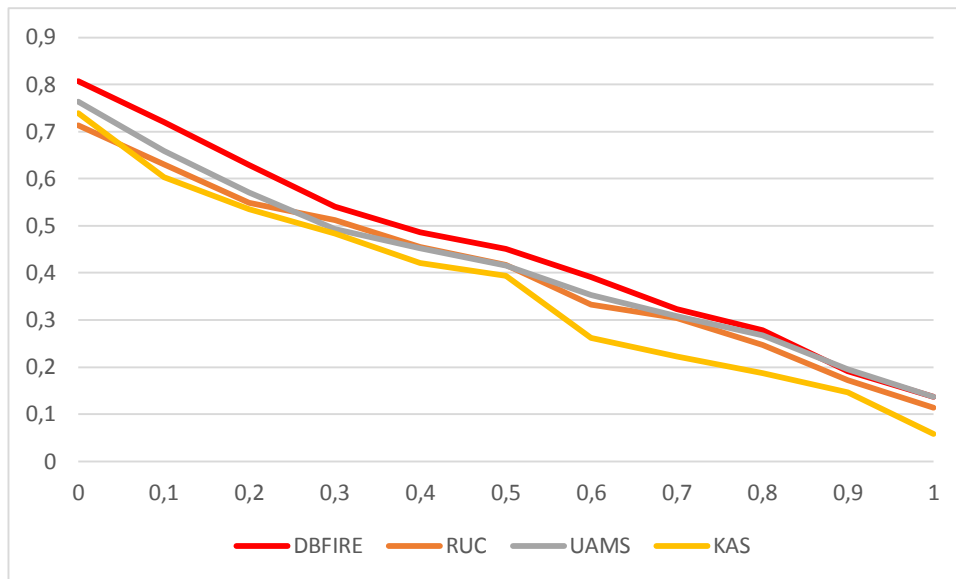


**Figura 7.4 - Precisão x Revocação no macro-cenário 1 (ambiente padrão de testes)**

Além dos métodos na Tabela 7.4, fizemos também comparativos com os sistemas INEX, como apresentado na seção 7.1. Os dados são exibidos na Tabela 7.5. O respectivo gráfico de precisão e revocação é mostrado na Figura 7.5.

	DBFIRE	KAS	RUC	UAMS
MAP	0.4326	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>24.3%</b>	<b>13.0%</b>	<b>8.9%</b>
<i>p</i> -valor	-	0.02	0.04	0.05
<i>Bpref</i>	0.4795	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>11.8%</b>	<b>17.7%</b>	<b>7.0%</b>
<i>p</i> -valor	-	0.24	0.03	0.02

**Tabela 7.5 - Comparativo com sistemas INEX (ambiente padrão de testes)**



**Figura 7.5 - Precisão x Revocação (sistemas INEX)**

### Discussão

Em todos os resultados para a métrica MAP vemos diferenças significativas em favor de DBFIRE, em todos os testes. As diferenças deixam de ser significativas apenas na comparação com a métrica *Bpref*, em especial com os métodos DFR, RM e KAS. De qualquer forma, os gráficos de precisão interpolada versus revocação mostram DBFIRE à frente de todos os métodos em quase todos os pontos.

O que os resultados não-significativos podem nos dizer? Que não é possível afirmar com certeza que DBFIRE supera esses métodos com os dados analisados, sendo necessários mais testes para a métrica *Bpref*. No entanto, a grande diferença relativa ao método KAS (pouco mais de 10%) é um bom indicativo a favor de DBFIRE.

Por outro lado, na comparação frente a outros métodos de ordenação de termos (KLD, DFR e RM), vê-se que as pequenas diferenças favoráveis a DBFIRE pedem mais testes para qualquer afirmação definitiva com respeito à métrica *Bpref*.

Outro ponto positivo em favor de DBFIRE é que a diferença relativa à métrica MAP entre os métodos KLD, DFR e RM com relação ao *baseline* foi muito pequena, e não-significativa<sup>34</sup>. Assim, não podemos dizer que eles bateram o *baseline*, mas certamente podemos fazer essa afirmação com relação a DBFIRE.

É de se registrar ainda o desempenho negativo do método SEMEX, e,

<sup>34</sup> Os testes estatísticos nestes casos mostraram um *p-valor* sempre acima de 0.2 para todos os métodos em comparação com o *baseline*.

principalmente, do método SCORE. Como antecipado no Capítulo 2, SCORE peca por usar o conteúdo completo de elementos de tupla como unidade de ordenação: com isso, não consegue diferenciar o potencial de cada termo individualmente para a melhoria dos resultados ou para desviar do foco da busca. Além disso, o fato de não usar nenhum indicativo da necessidade de informação do usuário, também lhe impacta negativamente<sup>35</sup>.

Já SEMEX se beneficia dos literais como indicação da necessidade de informação, mas o corpo da consulta SQL não parece ser uma boa fonte de termos adicionais para a busca.

### **7.3.2 Inserindo arquivos irrelevantes pertencentes a outro domínio na coleção de documentos**

A ideia de se criar um BD a partir dos documentos pode inserir um possível viés nos experimentos: o BD induzido fica um exato espelho da coleção de documentos. E isso deve ser muito raro acontecer numa organização típica: é bastante provável que lá existam documentos que nada têm a ver com os BDs da organização.

Com vistas a eliminar esse viés, fizemos o mesmo conjunto de testes da seção anterior, mas usando uma coleção de documentos “estendida”, formada pelos documentos da coleção INEX-DC em conjunto com documentos de uma outra coleção de testes. Assim, utilizamos os documentos de uma coleção TREC, referente à linha robusta (*Robust Track*), utilizada nos workshops de 2003 e 2004 [90].

Os comparativos frente ao *baseline* e aos métodos SCORE, SEMEX, KLD, DFR e RM são mostrados na Tabela 7.6, com o respectivo gráfico de precisão X revocação na Figura 7.6. Como a coleção de documentos foi alterada, o comparativo frente aos sistemas INEX (UAMS, RUC e KAS) não faria sentido, uma vez que seria necessário reexecutar todas as buscas para cada um deles (lembrando que as distribuições MAP e Bpref usados nos comparativos nesta tese são aqueles disponíveis nos artigos já publicados). Dessa forma, o comparativo frente aos sistemas INEX foi excluído deste cenário.

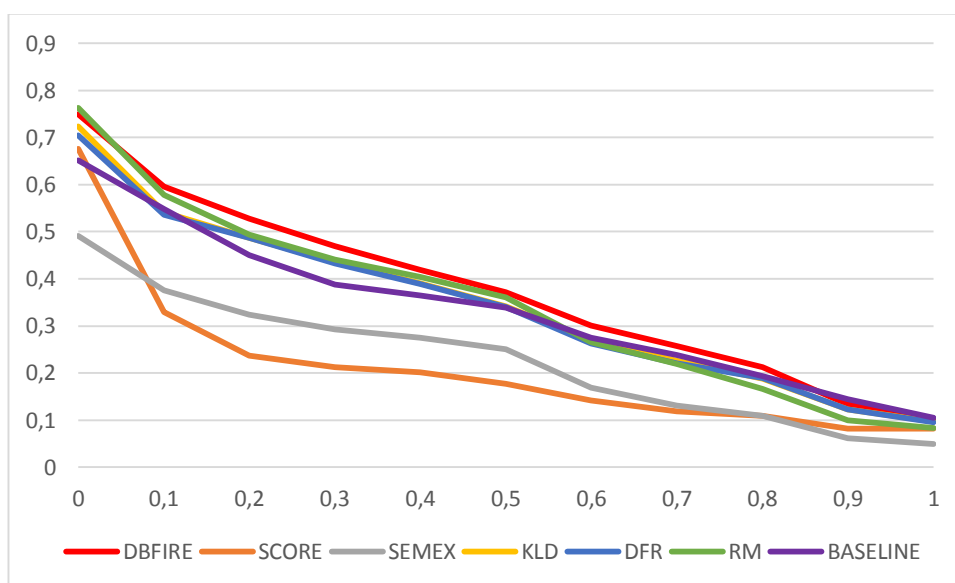
---

<sup>35</sup> A título de teste, realizamos a busca com SCORE expandindo as palavras-chave do usuário. Apesar de haver uma relativa melhora (seu MAP vai de 0.20 para 0.23) ainda não foi suficiente para ao menos se equiparar a SEMEX, por exemplo.



	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3518	0.3119	0.1871	0.2125	0.3199	0.3168	0.3230
<b>Diferença</b>	-	<b>12.7%</b>	<b>88.0%</b>	<b>65.5%</b>	<b>9.9%</b>	<b>11.0%</b>	<b>8.9%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<i>Bpref</i>	0.4673	0.4239	0.2574	0.3744	0.4478	0.4444	0.4414
<b>Diferença</b>	-	<b>10.2%</b>	<b>81.5%</b>	<b>24.8%</b>	<b>4.3%</b>	<b>5.1%</b>	<b>5.8%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.08	0.2	0.13

**Tabela 7.6 – Comparativo com inclusão de documentos da coleção TREC**



**Figura 7.6 - Precisão x Revocação (incluindo documentos TREC)**

### Discussão

Comparando com os dados exibidos na seção anterior, vê-se uma diminuição de desempenho em todos os métodos, quando consideramos o quesito precisão (seja por MAP, seja pelo gráfico precisão X revocação). Quando consideramos a métrica *Bpref*, a diferença não foi tão expressiva, até por que, pela própria natureza de *Bpref*, os eventuais documentos recuperados da coleção TREC são desconsiderados, pois não foram julgados.

Ainda assim, DBFIRE é superior a todos os concorrentes, com diferenças menos significativas novamente na métrica *Bpref*, agora frente aos métodos KLD e RM. Note-se que as diferenças relativas aumentaram um pouco, e o *p*-valor também ficou menor do que nos testes sem os arquivos adicionais. Note-se também que, ao contrário dos testes no ambiente anterior, todos os demais métodos de expansão não apresentaram diferenças significativas com relação ao *baseline*, tanto para a métrica *Bpref*, como para MAP.

Aparentemente, DBFIRE foi mais robusto quando exposto a esse ambiente mais

“hostil”, por assim dizer. Talvez isso possa ser explicado pelo fato de os demais métodos de expansão usarem alguma heurística quanto à frequência dos termos na coleção completa (ao contrário de DBFIRE, que se foca apenas no corpus de expansão).

### **7.3.3 Usando menos palavras-chave do usuário na busca expandida**

Imagine-se agora um cenário em que o usuário é mais sucinto na descrição de sua necessidade de informação. Até que ponto os métodos de expansão dependem da quantidade de palavras-chave a serem expandidas? Em tese, estamos ligando uma maior quantidade de termos a uma melhoria da expressão da necessidade da informação, o que pode não ser necessariamente verdade. No entanto, pelas diferenças que obtivemos nos valores das métricas, houve sim um impacto neste sentido.

Simulamos esse cenário considerando como palavras-chave do usuário os literais de cada consulta SQL. Como exemplo, vamos revisitar a consulta apresentada no Capítulo 2: os filmes dirigidos por Martin Scorsese. Enquanto as palavras chave do usuário podem partir de “Martin Scorsese movies” ou algo como “movies directed by Martin Scorsese”, os literais da consulta se resumem a “Martin Scorsese”. Assim, as palavras-chave do usuário tendem a trazer mais dados sobre a necessidade de informação do usuário do que apenas os literais da consulta.

Além de simular uma descrição mais curta da necessidade de informação, esse cenário simula também a execução totalmente automática de DBFIRE, sem interferência do usuário. Esse cenário foi realizado no mesmo ambiente que aquele da seção 7.3.1, i.e., com a coleção de documentos formada pelos arquivos originais INEX-DC.

O comparativo com o *baseline*, SCORE, SEMEX, KLD, DFR e RM encontra-se na Tabela 7.7, enquanto que o respectivo gráfico de precisão X revocação é exibido na Figura 7.7. Por sua vez, os dados da comparação com os sistemas INEX estão na Tabela 7.8, e o gráfico de precisão X revocação deste teste é exibido na Figura 7.8.

#### **Discussão**

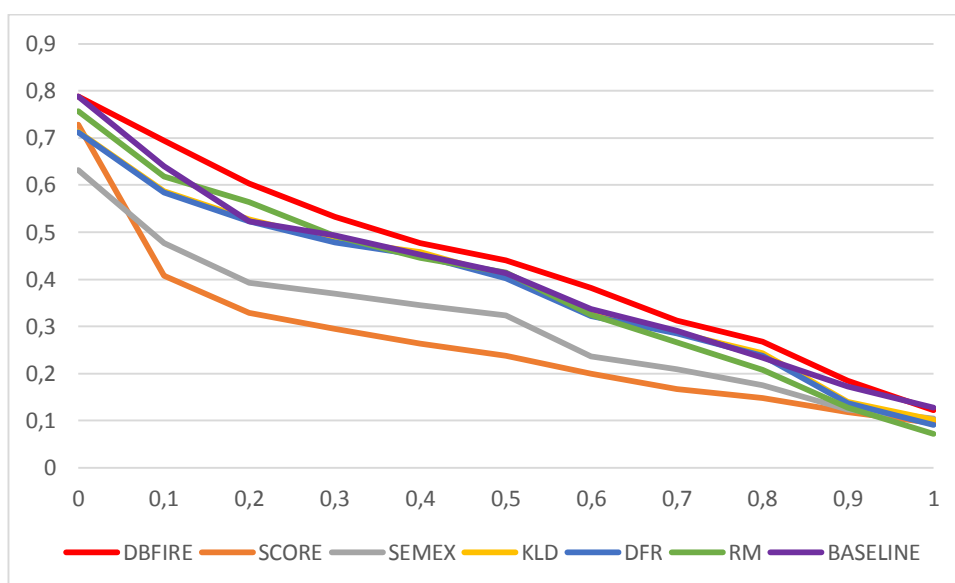
DBFIRE ainda bate todos os sistemas em todas as métricas de qualidade, sendo que as diferenças com relação aos métodos KLD, DFR e RM na métrica *Bpref* passaram a ser significativas. No entanto, no comparativo com os sistemas INEX isso não ocorreu.

Podemos concluir que esse é um cenário que dificulta a expansão, haja vista a diferença no desempenho relativo à primeira configuração de testes (seção 7.3.1): houve perda de desempenho em todos os métodos de expansão. Apesar disso, DBFIRE saiu-se

bem frente a esses métodos.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.4186	0.3790	0.2035	0.2840	0.3677	0.3648	0.3709
<b>Diferença</b>	-	<b>10.4%</b>	<b>&gt;100%</b>	<b>47.4%</b>	<b>13.8%</b>	<b>14.7%</b>	<b>12.8%</b>
<i>p</i> -valor	-	0.03	<0.01	<0.01	<0.01	<0.01	<0.01
<i>Bpref</i>	0.4640	0.4120	0.2578	0.3678	0.4450	0.4443	0.4309
<b>Diferença</b>	-	<b>12.6%</b>	<b>79.9%</b>	<b>26.1%</b>	<b>4.2%</b>	<b>4.4%</b>	<b>5.7%</b>
<i>p</i> -valor	-	0.01	<0.01	<0.01	0.04	0.03	0.03

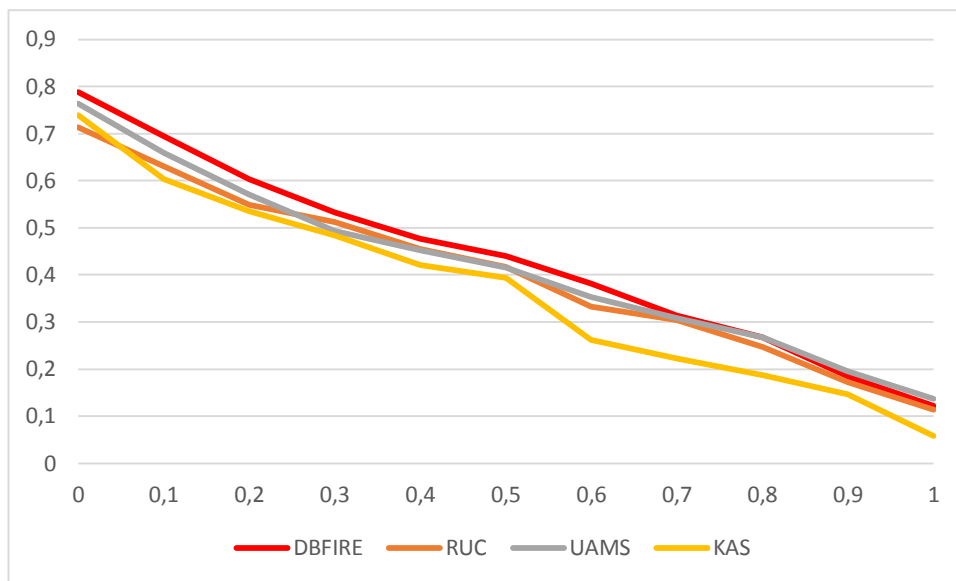
**Tabela 7.7 – Comparativo usando literais da consulta como palavras-chave para expansão**



**Figura 7.7 - Precisão x Revocação com literais da consulta como palavras-chave**

	DBFIRE	KAS	RUC	UAMS
MAP	0.4186	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>20.3%</b>	<b>9.3%</b>	<b>5.4%</b>
<i>p</i> -valor	-	0.09	0.12	0.22
<i>Bpref</i>	0.4640	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>8.2%</b>	<b>13.9%</b>	<b>3.6%</b>
<i>p</i> -valor	-	0.41	0.04	0.09

**Tabela 7.8 - Comparativo com sistemas INEX usando literais da consulta como palavras-chave para expansão**



**Figura 7.8 - Precisão x Revocação com literais da consulta como palavras-chave versus sistemas INEX**

### 7.3.4 Excluindo campos do resultado da consulta

As consultas no BD induzido se utilizaram de todos os campos disponíveis no seu resultado: eram basicamente formadas por um `SELECT *` operando em todas as tabelas. O que aconteceria se alguns desses campos não aparecessem no resultado, notadamente aqueles em que se necessita de uma junção entre tabelas para chegar ao que se deseja? Isso é o que acontece na consulta “Filmes dirigidos por Martin Scorsese”: ao recuperar todos os campos da consulta (`SELECT *`), o resultado conterà também informações sobre Martin Scorsese, mesmo que o desejado fossem somente os seus filmes.

Além disso, outra característica da coleção INEX-DC é a existência de muitos campos do tipo `TEXT`, a exemplo da descrição da trama de um filme (campo *plot*), ou da biografia de uma personalidade (campo *biography*). Em alguns casos, esses campos contêm praticamente um documento por completo. Essa situação não é a mais frequente nas consultas mais corriqueiras dentro de uma organização.

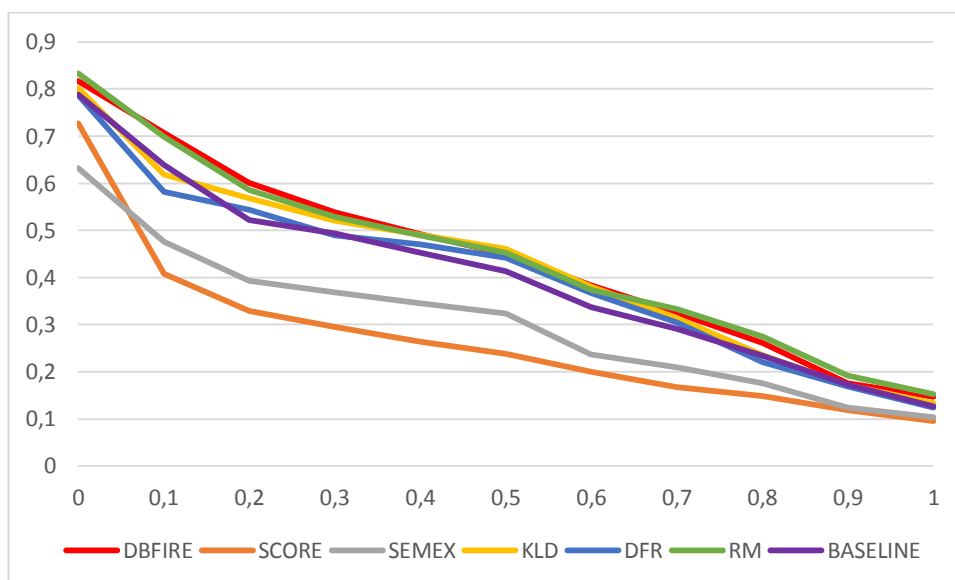
Assim, esse cenário de teste parte da configuração básica apresentada na seção 7.3.1, mas removendo dos resultados das consultas qualquer campo referente a junções: se a consulta pede um filme, apenas os dados do filme são apresentados, assim como quando a necessidade de informação pedir uma pessoa (ator, diretor, etc.). Campos do tipo `TEXT` também são removidos.

Da mesma maneira que nas seções anteriores, os dados estão divididos em duas

tabelas, a Tabela 7.9 para o *baseline*, SEMEX, SCORE, KLD, DFR e RM, e a Tabela 7.10 para os sistemas INEX. Os respectivos gráficos de precisão e revocação também são exibidos na Figura 7.9 e na Figura 7.10.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.4231	0.3790	0.2035	0.2840	0.4048	0.3869	0.4262
<b>Diferença</b>	-	<b>11.6%</b>	<b>&gt;100%</b>	<b>47.4%</b>	<b>4.5%</b>	<b>9.3%</b>	<b>-0.7%</b>
<i>p</i> -valor	-	0.03	<0.01	<0.01	0.04	0.01	0.43
<i>Bpref</i>	0.4816	0.4120	0.2578	0.3678	0.4999	0.4847	0.4910
<b>Diferença</b>	-	<b>16.8%</b>	<b>86.8%</b>	<b>30.9%</b>	<b>-3.6%</b>	<b>-0.6%</b>	<b>-1.9%</b>
<i>p</i> -valor	-	0.01	<0.01	<0.01	0.93	0.68	0.77

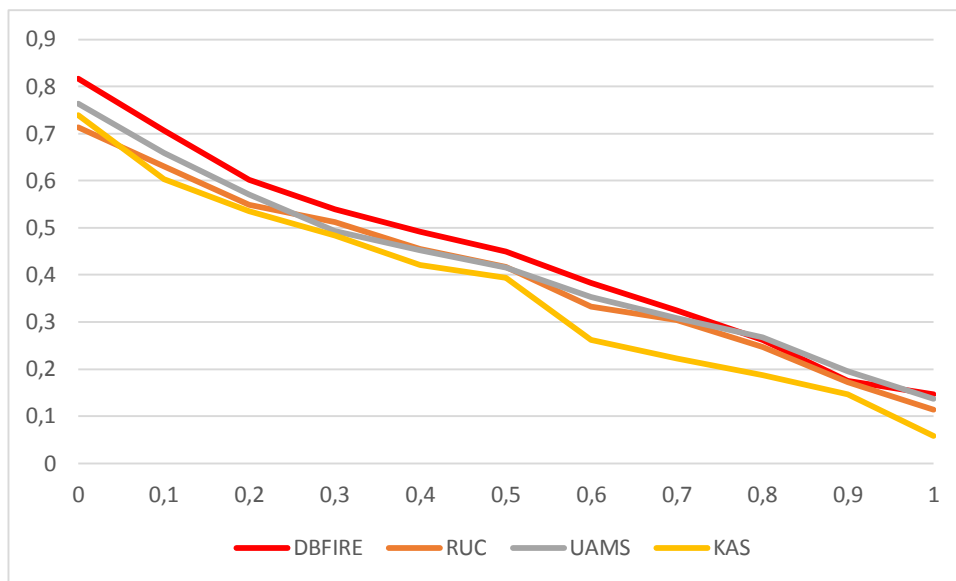
**Tabela 7.9 – Comparativo removendo campos do resultado da consulta**



**Figura 7.9 - Precisão x Revocação removendo campos do resultado da consulta**

	DBFIRE	KAS	RUC	UAMS
MAP	0.4231	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>21.6%</b>	<b>10.5%</b>	<b>6.6%</b>
<i>p</i> -valor	-	0.07	0.8	0.06
<i>Bpref</i>	0.4640	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>12.3%</b>	<b>18.2%</b>	<b>7.5%</b>
<i>p</i> -valor	-	0.43	0.06	0.16

**Tabela 7.10 - Comparativo com sistemas INEX removendo campos do resultado da consulta**



**Figura 7.10 - Precisão x Revocação frente a sistemas INEX (remoção de campos)**

### Discussão

Neste cenário DBFIRE mantém o bom desempenho frente ao *baseline*, a SCORE e SEMEX. Apesar de podermos questionar se há diferenças significativas nas comparações frente aos métodos INEX (pois o nível do p-valor ficou acima de 0.05), o gráfico de precisão X revocação mostra DBFIRE bem posicionado quanto a esses métodos.

No entanto, isso não acontece na comparação frente ao método RM: o gráfico na Figura 7.9 mostra os dois métodos se alternando como o melhor dependendo do nível de revocação. Além disso, pela primeira vez nos experimentos vimos DBFIRE ser batido por algum outro método (RM na comparação por MAP, e todos os de expansão na comparação via *Bpref*). Mesmo assim, as diferenças foram ainda muito pequenas.

Qual a diferença deste cenário para os demais? Com a remoção do conteúdo de alguns campos do resultado da consulta, temos um corpus de expansão menor, com menos elementos por tuplas e com menos termos por elementos. Como esse é o foco principal de DBFIRE, essa pode ter sido a razão de seu desempenho menor frente aos demais métodos de expansão.

## 7.4 Comparativos no Macro-Cenário 2

Os experimentos a seguir detalham os comparativos na mesma coleção INEX-DC como na seção anterior, mas agora recuperando arquivos diretamente via buscas estruturadas. Neste caso, os primeiros  $k$  documentos XML retornados são analisados para seleção de termos para expansão; os  $n$  melhores termos são submetidos ao SRI. Os detalhes de como

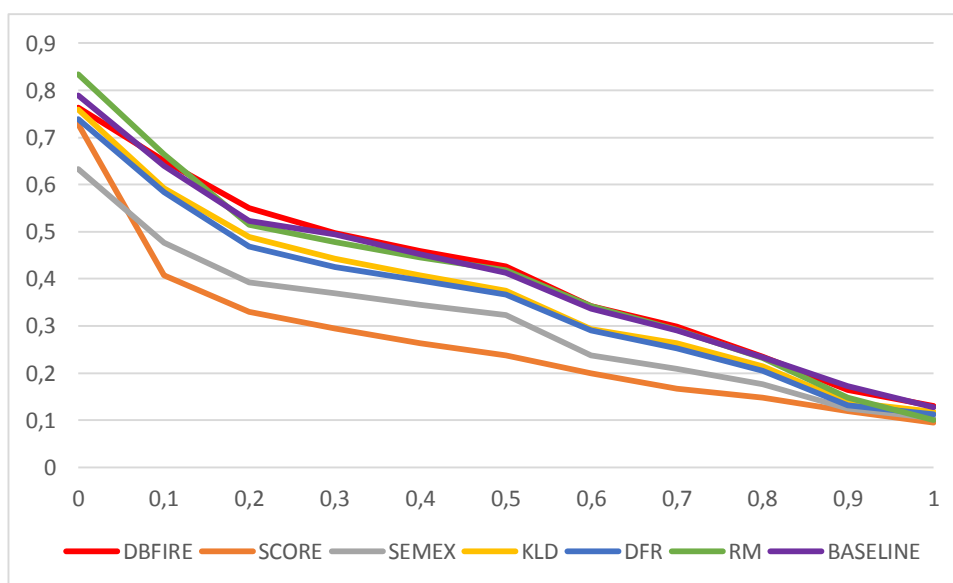
converter os documentos recuperados em tuplas para DBFIRE encontram-se na seção 6.1.2.

### 7.4.1 Comparativos no ambiente padrão de testes

Os comparativos aqui se deram na configuração padrão, i.e., com todos os campos retornados nas buscas estruturadas, e com a expansão a partir das palavras-chave do usuário. Analogamente às seções anteriores, os comparativos foram divididos em duas tabelas (Tabela 7.11 e Tabela 7.12) e dois gráficos de precisão e revocação (Figura 7.11 e Figura 7.12).

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3914	0.3790	0.2035	0.2840	0.3454	0.3363	0.3805
<b>Diferença</b>	-	<b>3.2%</b>	<b>92.3%</b>	<b>37.8%</b>	<b>13.3%</b>	<b>16.3%</b>	<b>2.8%</b>
<i>p</i> -valor	-	0.86	<0.01	<0.01	<0.01	<0.01	0.43
<i>Bpref</i>	0.4377	0.4120	0.2578	0.3678	0.4086	0.4064	0.4394
<b>Diferença</b>	-	<b>6.2%</b>	<b>69.7%</b>	<b>19.0%</b>	<b>7.0%</b>	<b>7.6%</b>	<b>-0.4%</b>
<i>p</i> -valor	-	0.23	<0.01	<0.01	0.25	0.13	1

**Tabela 7.11 – Comparativo padrão no macro-cenário 2**



**Figura 7.11 - Precisão x Revocação (comparativo padrão no macro-cenário 2)**

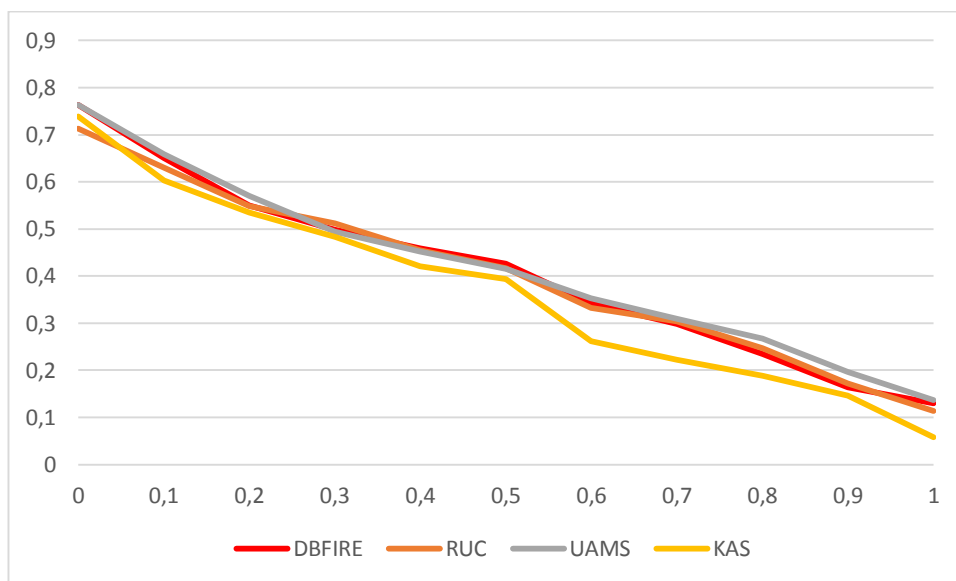
#### Discussão

Esse é o único macro-cenário em que as consultas foram realizadas por terceiros. E como fazem parte de um workshop em que se objetiva criar condições mais difíceis para os participantes, a intenção não era exatamente a de criar consultas exatas. Vejamos o que isso pode acarretar analisando o tópico 2011107, já apresentado no Capítulo 4 (*Tom*

*Hanks biography*).

	DBFIRE	KAS	RUC	UAMS
MAP	0.3914	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>12.5%</b>	<b>2.2%</b>	<b>-1.39%</b>
<i>p</i> -valor	-	0.34	0.67	0.52
<i>Bpref</i>	0.4377	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>2.0%</b>	<b>7.4%</b>	<b>-2.25%</b>
<i>p</i> -valor	-	0.94	0.72	0.89

**Tabela 7.12 - Comparativo padrão com sistemas INEX**



**Figura 7.12 - Precisão x Revocação (comparativo padrão no macro-cenário 2 frente aos sistemas INEX)**

Veja abaixo o texto da versão NEXI da coleção e logo a seguir a versão SQL feita para as consultas ao BD induzido:

```
//person[about(., Tom Hanks)]

SELECT DISTINCT * FROM `imdb_inex`.`person` as P,
`imdb_inex`.`biographies` as B
where P.idperson=B.idperson
and match(B.biography) against ("Tom Hanks" in boolean mode)
```

Na versão NEXI, não há uma restrição à ocorrência exclusiva da frase nominal “Tom Hanks”, e muito menos se esses termos devem aparecer apenas no campo *biography*: em princípio, qualquer ocorrência de “Tom” e/ou “Hanks” em qualquer seção de um documento sobre uma personalidade seria um documento válido. Claro que a



consulta SQL também não é completamente exata (ela tem todos os defeitos da RI, já que se baseia no índice FULL TEXT), mas é bem mais restritiva.

Mesmo tendo isso em mente, DBFIRE ainda consegue bons resultados frente aos demais métodos, ainda que vários sem significância estatística, notadamente frente ao *baseline*. Essa pouca diferença estatística pode ser comprovada no gráfico de precisão X revocação, em que o comportamento de DBFIRE, do *baseline* e do método RM são bastante próximos.

Apesar de já considerarmos este cenário hostil, fizemos mais testes para analisar os efeitos de outros cenários já colocados na seção 7.3. Dos três casos lá tratados, excluimos apenas a concatenação dos documentos TREC, pelo fato de esses documentos não estarem no formato XML como esperado para as consultas automáticas em linguagem NEXI.

#### 7.4.2 Expansão com os literais da consulta

Retomamos aqui o cenário que parte da expansão usando menos palavras-chave do usuário. Os dados são mostrados na Tabela 7.13 e na Tabela 7.14, enquanto os gráficos de precisão X revocação aparecem na Figura 7.13 e na Figura 7.14.

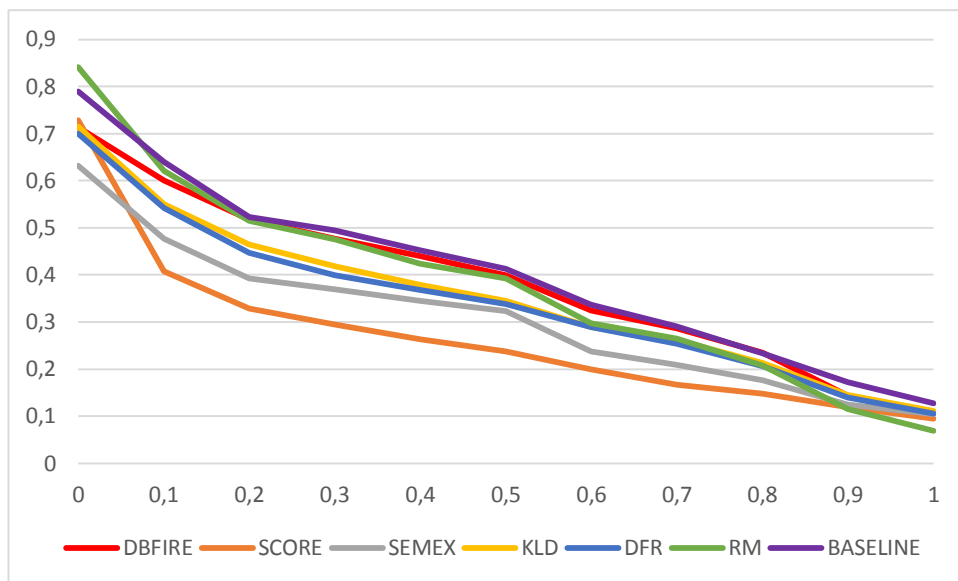
	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3695	0.3790	0.2035	0.2840	0.3306	0.3216	0.3585
<b>Diferença</b>	-	<b>-2.5%</b>	<b>81.5%</b>	<b>30.1%</b>	<b>11.7%</b>	<b>14.8%</b>	<b>3.0%</b>
<i>p</i> -valor	-	0.48	<0.01	<0.01	<0.01	<0.01	0.09
<i>Bpref</i>	0.4158	0.4120	0.2578	0.3678	0.4025	0.4027	0.4246
<b>Diferença</b>	-	<b>0.91%</b>	<b>61.2%</b>	<b>13.0%</b>	<b>3.3%</b>	<b>3.2%</b>	<b>-4.6%</b>
<i>p</i> -valor	-	0.7	<0.01	<0.01	0.22	0.25	0.44

**Tabela 7.13 – Comparativo macro-cenário 2 (expansão com os literais)**

#### Discussão

Não há dúvida de que este cenário foi especialmente danoso para DBFIRE: pela primeira vez ele passa a ter um desempenho abaixo do *baseline*, ainda que a diferença não seja significativa. Vê-se também que o mesmo acontece com os demais métodos de expansão.

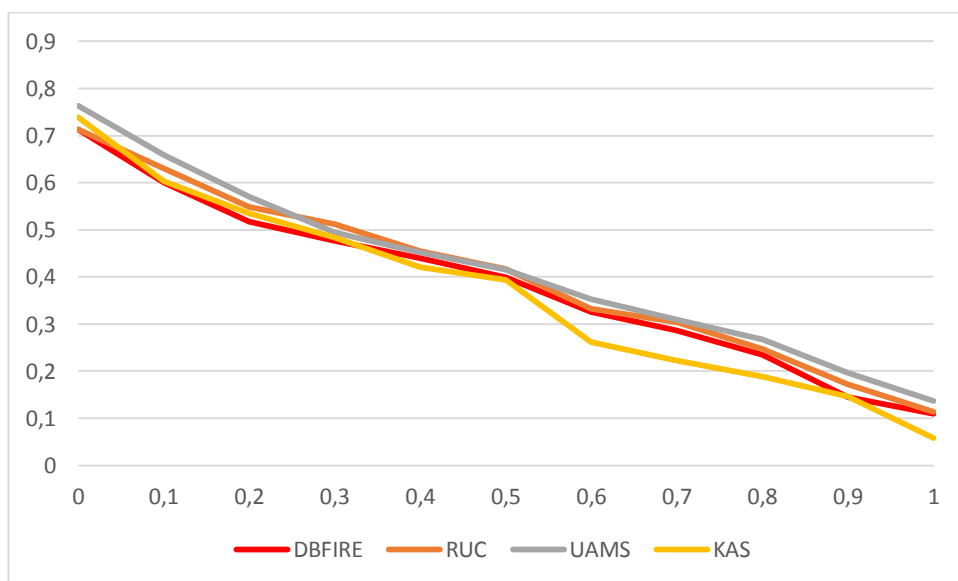
Pelo gráfico de precisão X revocação há uma leve coincidência entre a evolução de DBFIRE, RM e o *baseline*, tendo DBFIRE apresentado a pior posição nos pontos iniciais da curva.



**Figura 7.13 - Precisão x Revocação (expansão com os literais)**

	DBFIRE	KAS	RUC	UAMS
MAP	0.3695	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>6.2%</b>	<b>-3.4%</b>	<b>-6.9%</b>
<i>p</i> -valor	-	0.76	0.26	0.22
<i>Bpref</i>	0.4158	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>-3.0%</b>	<b>2.1%</b>	<b>-7.1%</b>
<i>p</i> -valor	-	0.67	0.83	0.38

**Tabela 7.14 - Comparativo com sistemas INEX no macro-cenário 2 (expansão com os literais)**



**Figura 7.14 - Precisão x Revocação no macro-cenário 2 (expansão com os literais)**

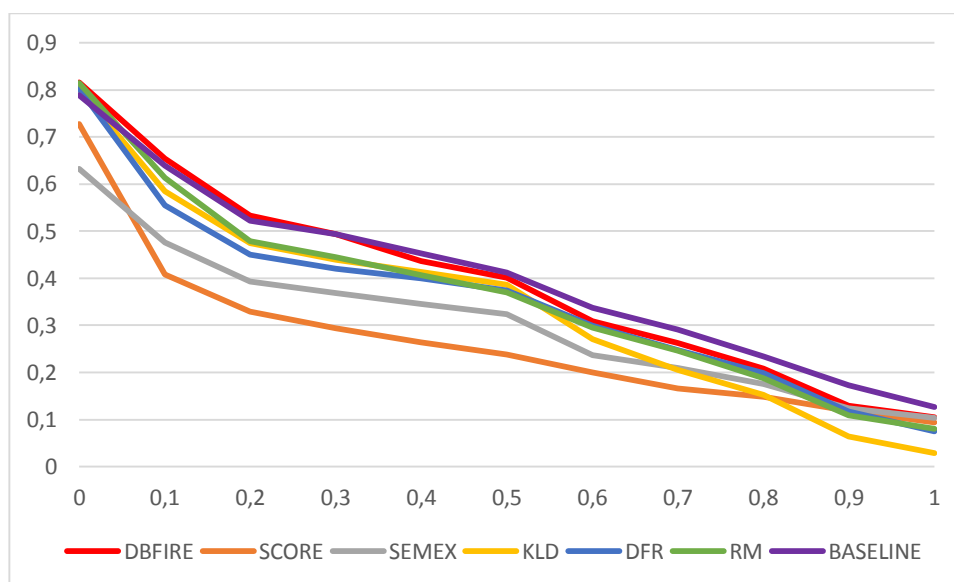
### 7.4.3 Removendo campos dos resultados das consultas

Este cenário simula consultas retornando o conteúdo de menos campos do que aqueles presentes no comparativo padrão: particularmente campos de tabelas adicionais que venham a aparecer devido a junções, e campos do tipo TEXT. No nosso caso, a junção nem precisa ser explicitada na consulta, pois todas as eventuais junções já se encontram disponíveis nos documentos XML recuperados: assim, tudo o que se fez foi desprezar o conteúdo desses campos.

Mais uma vez, os comparativos estão divididos em duas tabelas: Tabela 7.15 e Tabela 7.16. Da mesma forma, os gráficos de precisão e revocação: Figura 7.15 e Figura 7.16.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3668	0.3790	0.2035	0.2840	0.3195	0.3306	0.3351
<b>Diferença</b>	-	<b>-3.2%</b>	<b>80.2%</b>	<b>7.0%</b>	<b>14.8%</b>	<b>10.9%</b>	<b>9.4%</b>
<i>p</i> -valor	-	0.7	<0.01	0.14	0.07	0.02	0.05
<i>Bpref</i>	0.3938	0.4120	0.2578	0.3678	0.3916	0.4082	0.3894
<b>Diferença</b>	-	<b>-4.4%</b>	<b>52.7%</b>	<b>19.0%</b>	<b>0.55%</b>	<b>-3.5%</b>	<b>1.1%</b>
<i>p</i> -valor	-	0.56	<0.01	<0.01	0.74	0.89	0.86

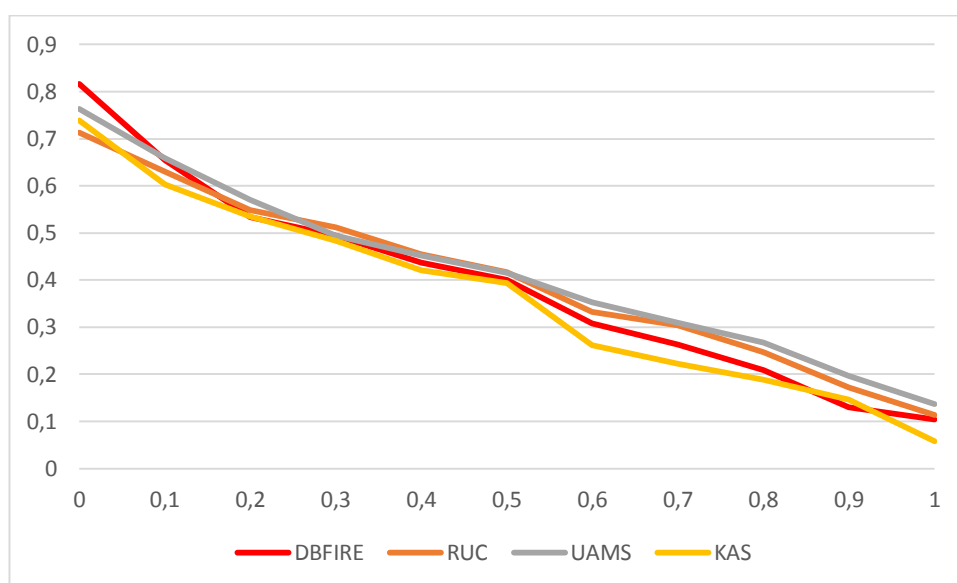
**Tabela 7.15 – Comparativo macro-cenário 2 (removendo campos)**



**Figura 7.15 - Precisão x Revocação macro-cenário 2 (removendo campos)**

	DBFIRE	KAS	RUC	UAMS
MAP	0.3668	0.3478	0.3828	0.3969
<b>Diferença</b>	-	<b>5.4%</b>	<b>-4.1%</b>	<b>-7.57%</b>
<i>p</i> -valor	-	0.53	0.57	0.67
<i>Bpref</i>	0.3938	0.4288	0.4072	0.4478
<b>Diferença</b>	-	<b>-8.1%</b>	<b>-3.3%</b>	<b>-12.0%</b>
<i>p</i> -valor	-	0.49	0.78	0.55

**Tabela 7.16 - Comparativo macro-cenário 2 com sistemas INEX (removendo campos)**



**Figura 7.16 - Precisão x Revocação macro-cenário 2 com sistemas INEX (removendo campos)**

### Discussão

Este cenário é bem mais nocivo aos métodos de expansão. O *baseline* não é superado por nenhum deles e pela primeira vez a diferença de DBFIRE ante SEMEX passa a não ser significativa na métrica MAP. Apesar disso, o gráfico de precisão X revocação sempre mostra DBFIRE acima de SEMEX; com relação ao *baseline*, DBFIRE tem uma tendência equivalente, sendo superado nos últimos pontos de revocação.

No comparativo gráfico ante os métodos INEX, DBFIRE tem um comportamento em que ora aparece à frente dos demais, ora empatado, ora abaixo. Todas as situações se devem à dificuldade inerente do cenário.

## 7.5 Comparativos no Macro-Cenário 3

Aqui serão abordados os testes com a Coleção INEX-LOD. Os testes compreenderam o comparativo básico, semelhante àquele das seções anteriores e outros em dois cenários em condições mais adversas.

Não chegamos a mostrar os dados para o cenário com a inclusão de arquivos da coleção TREC *Robust* pelo fato de não termos encontrado nenhuma mudança significativa nos resultados; isso se explica porque a coleção INEX-LOD já é muito grande: os cerca de 500 mil documentos da coleção TREC *Robust* não teriam mesmo como fazer diferença ante os mais de 10 milhões da coleção INEX-LOD.

### **7.5.1 Comparativo nas configurações padrão**

Neste comparativo, fizemos testes de DBFIRE frente ao *baseline*, aos sistemas SCORE, SEMEX, KLD, DFR e RM, e aos sistemas participantes do workshop INEX.

Os números do comparativo aparecem na Tabela 7.17 e na Tabela 7.18, enquanto que os gráficos de precisão/revocação são exibidos na Figura 7.17 e na Figura 7.18.

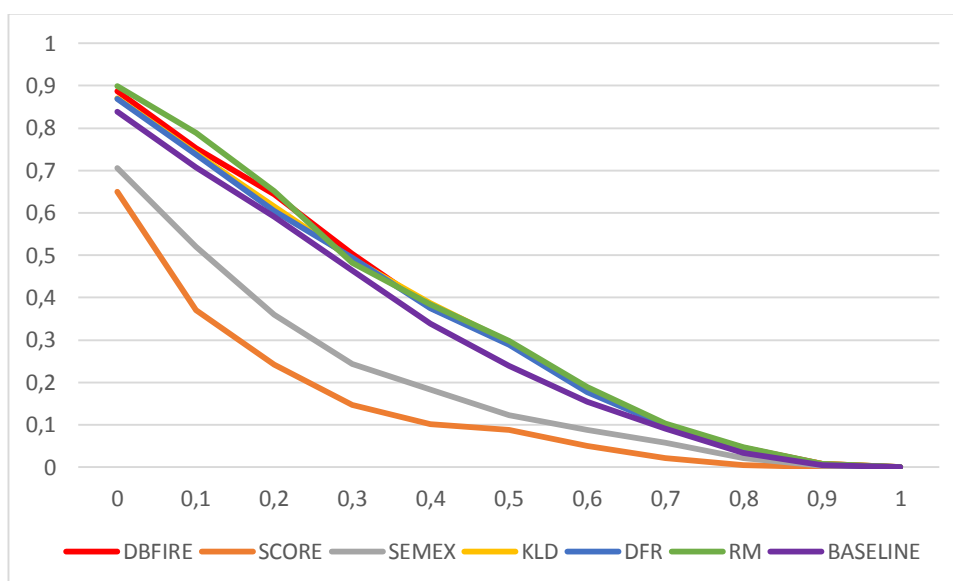
#### **Discussão**

DBFIRE aparece bem frente ao *baseline*, SCORE e SEMEX, mas tem desempenho praticamente idêntico aos demais métodos de expansão. Isso fica bem claro no comparativo gráfico, onde as curvas de todos os métodos de expansão quase que se sobrepõem, vindo a curva do *baseline* logo abaixo.

Quanto aos sistemas INEX vê-se um fato interessante: uma diferença significativa desfavorável a DBFIRE na comparação com a métrica MAP, torna-se significativamente favorável quando comparado via métrica *Bpref*. Isso deve refletir um viés na coleção devido à avaliação incompleta através de *pool* de documentos. Esse efeito foi mais forte aqui do que na coleção INEX-DC provavelmente por haver menos sistemas participantes gerando documentos relevantes para o *pool*. Assim, é provável que muitos documentos retornados por DBFIRE não tenham tido a chance de terem sua relevância avaliada.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3206	0.2923	0.1284	0.1880	0.3148	0.3120	0.3251
<b>Diferença</b>	-	<b>9.6%</b>	<b>&gt;100%</b>	<b>70.5%</b>	<b>1.86%</b>	<b>2.7%</b>	<b>-1.3%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.29	0.23	0.94
<i>Bpref</i>	0.4510	0.4023	0.3067	0.3622	0.4525	0.4511	0.4568
<b>Diferença</b>	-	<b>12.09%</b>	<b>47.0%</b>	<b>24.4%</b>	<b>-0.34%</b>	<b>-0.02%</b>	<b>-1.2%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.42	0.54	0.06

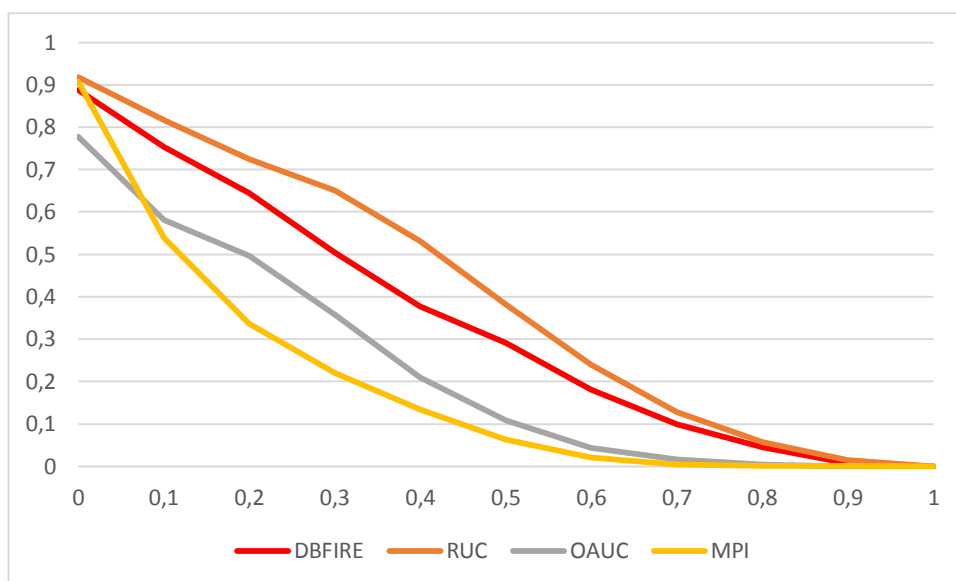
**Tabela 7.17 – Resultados macro-cenário 3 (comparativo padrão)**



**Figura 7.17 – Precisão x Revocação (comparativo padrão)**

	DBFIRE	MPI	OAUC	RUC
MAP	0.3206	0.1183	0.1488	0.3459
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>&gt;100%</b>	<b>-7.3</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01
<i>Bpref</i>	0.4510	0.1281	0.1998	0.4103
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>&gt;100%</b>	<b>9.9%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01

**Tabela 7.18 – Comparativo padrão com sistemas INEX)**



**Figura 7.18 – Precisão x Revocação (comparativo padrão com sistemas INEX)**

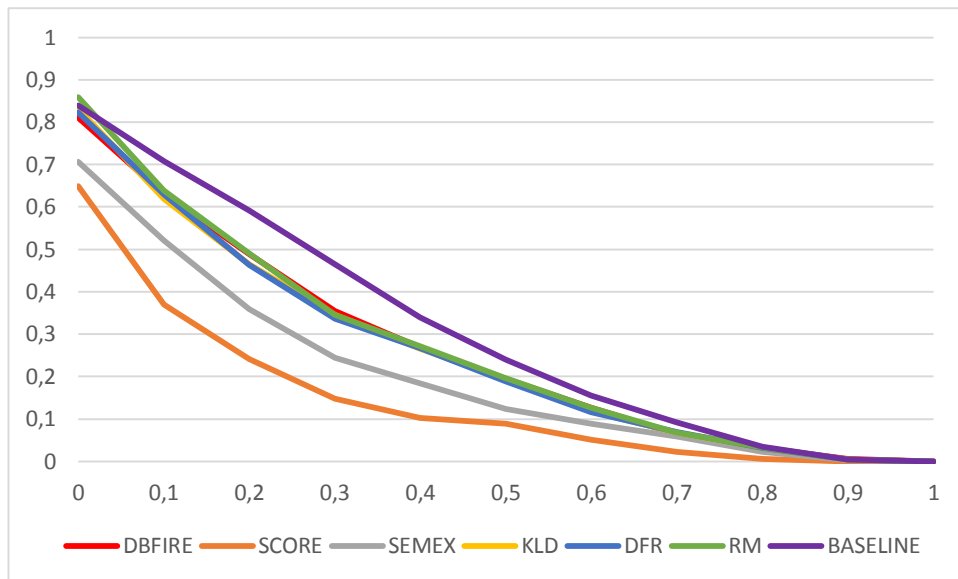
Notar que o gráfico de precisão/revocação mostra uma grande diferença favorável ao método RUC frente a DBFIRE. Não é de se estranhar, já que para o cálculo da precisão interpolada (assim como para MAP) considera-se documentos não-julgados como sendo irrelevantes. Assim, já era esperado que o quadro fortemente desfavorável a DBFIRE observado com a métrica MAP também se apresentasse graficamente.

### 7.5.2 Efetuando a expansão com os literais da consulta SQL

Repetimos a simulação da expansão a partir de menos palavras-chave do usuário, representadas aqui pelos literais das consultas SPARQL. Os dados são apresentados na Tabela 7.19 e na Tabela 7.20, e os comparativos gráficos na Figura 7.19 e na Figura 7.20.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.2458	0.2923	0.1284	0.1880	0.2427	0.2413	0.2501
<b>Diferença</b>	-	<b>-15.9%</b>	<b>91.4%</b>	<b>30.7%</b>	<b>1.27%</b>	<b>1.8%</b>	<b>-1.7%</b>
<i>p</i> -valor	-	0.04	<0.01	<0.01	0.36	0.22	0.37
<i>Bpref</i>	0.4390	0.4023	0.3067	0.3622	0.4355	0.4349	0.4445
<b>Diferença</b>	-	<b>9.1%</b>	<b>43.1%</b>	<b>21.1%</b>	<b>0.80%</b>	<b>0.95%</b>	<b>-1.2%</b>
<i>p</i> -valor	-	0.05	<0.01	<0.01	0.21	0.35	0.03

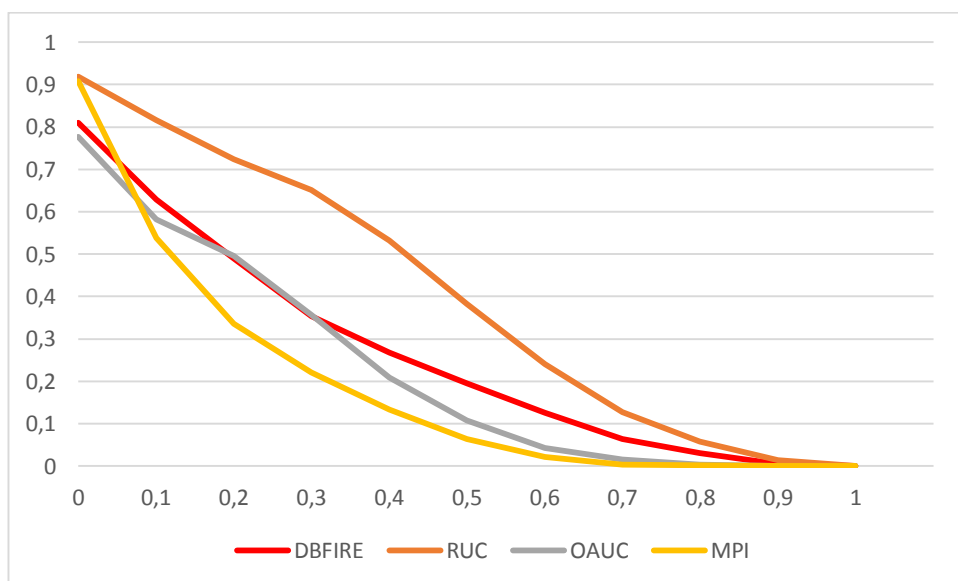
**Tabela 7.19 – Resultados macro-cenário 3 (expandido a partir dos literais)**



**Figura 7.19 – Precisão x Revocação (expandido a partir dos literais)**

	DBFIRE	MPI	OAUC	RUC
MAP	0.2458	0.1183	0.1488	0.3459
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>65.1%</b>	<b>-28.9</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01
<i>Bpref</i>	0.4390	0.1281	0.1998	0.4103
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>&gt;100%</b>	<b>7.0%</b>
<i>p</i> -valor	-	<0.01	<0.01	0.25

**Tabela 7.20 – Comparativo com sistemas INEX expandido a partir dos literais**



**Figura 7.20 – Precisão x Revocação (comparativo padrão com sistemas INEX)**

**Discussão**



Repete-se aqui algo que aconteceu na subseção anterior: no comparativo relativo ao *baseline* diferenças nocivas a DBFIRE quando consideradas pela métrica MAP, tornam-se significativamente favoráveis quando usamos *Bpref*. Além disso, o comparativo gráfico mostra todos os métodos de expansão praticamente coincidentes e bem abaixo do *baseline*.

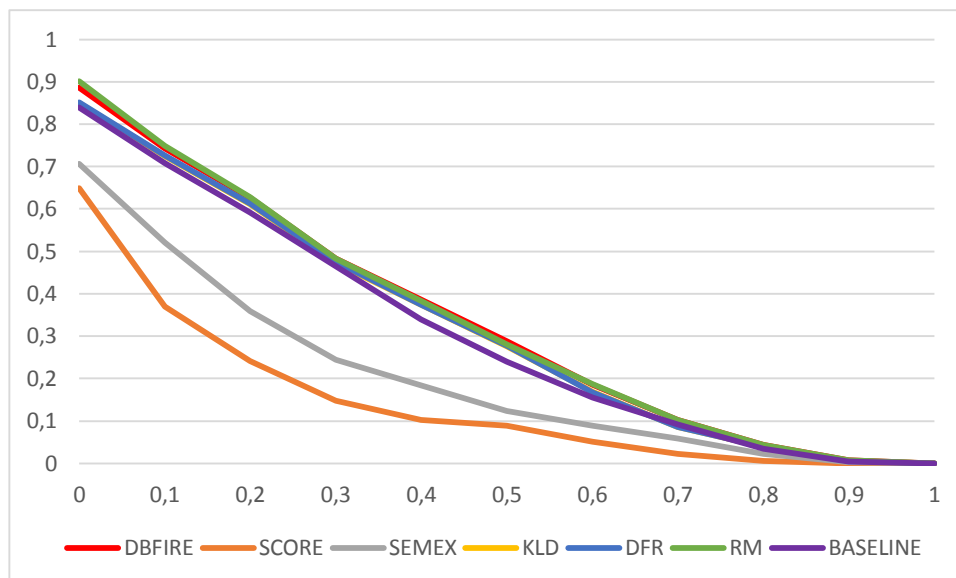
A distância para os métodos INEX também aparece bem marcante no gráfico de precisão X revocação. Novamente, esse efeito já era esperado já que a diferença relativa à métrica MAP também foi acentuada.

### 7.5.3 Removendo campos do resultado da consulta SPARQL

Inicialmente, é necessário lembrar que os resultados das consultas SPARQL são retornados no formato de triplas RDF: <recurso, atributo, valor>. Como esse modelo de dados não nos dá uma tabela de forma explícita como no mundo relacional, optamos por remover todos os valores relativos aos atributos, utilizando como corpus da expansão apenas os valores dos recursos. Por exemplo, na consulta pelos países da América Central, usamos como tuplas apenas os nomes dos países, sem nenhum outro item relativo a seus atributos (como língua, continente, etc.). Os comparativos aparecem na Tabela 7.21 e na Tabela 7.22, enquanto que os gráficos de precisão X revocação vêm na Figura 7.21 e na Figura 7.22.

	DBFIRE	BASELINE	SCORE	SEMEX	KLD	DFR	RM
MAP	0.3180	0.2923	0.1284	0.1880	0.3057	0.3065	0.3184
<b>Diferença</b>	-	<b>8.7%</b>	<b>&gt;100%</b>	<b>69.1%</b>	<b>4.0%</b>	<b>3.7%</b>	<b>-1.2%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.42	0.3	0.67
<i>Bpref</i>	0.4463	0.4023	0.3067	0.3622	0.4345	0.4357	0.4487
<b>Diferença</b>	-	<b>10.9%</b>	<b>45.5%</b>	<b>23.2%</b>	<b>2.7%</b>	<b>2.4%</b>	<b>-0.5%</b>
<i>p</i> -valor	-	<0.01	<0.01	<0.01	0.55	0.94	0.79

**Tabela 7.21 – Resultados macro-cenário 3 (removendo campos da consulta)**



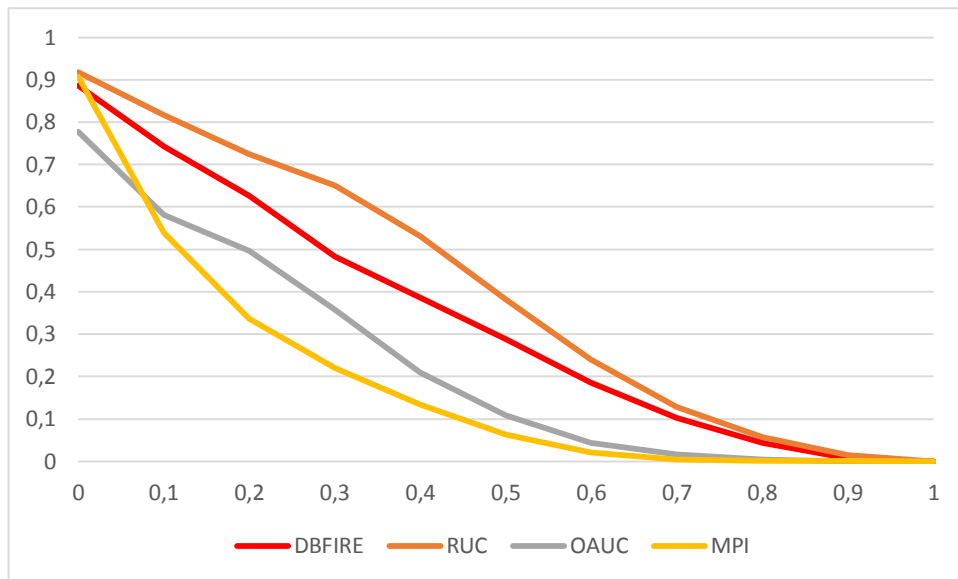
**Figura 7.21 – Precisão x Revocação (removendo campos da consulta)**

### Discussão

Neste cenário tivemos resultados bem similares àqueles do comparativo padrão, com DBFIRE superando o *baseline* nas métricas MAP e *Bpref*, com uma coincidência gráfica entre todos os métodos de expansão e o *baseline*, e resultados desfavoráveis a DBFIRE que passam a favoráveis quando se considera a métrica *Bpref* nos comparativos com os sistemas INEX. É como se os termos advindos dos outros atributos da consulta nada contribuíssem para a expansão nesta coleção.

	DBFIRE	MPI	OAUC	RUC
MAP	0.3180	0.1183	0.1488	0.3459
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>&gt;100%</b>	<b>-8.0</b>
<i>p</i> -valor	-	<0.01	<0.01	0.07
<i>Bpref</i>	0.4463	0.1281	0.1998	0.4103
<b>Diferença</b>	-	<b>&gt;100%</b>	<b>&gt;100%</b>	<b>8.7%</b>
<i>p</i> -valor	-	<0.01	<0.01	0.02

**Tabela 7.22 – Comparativo com sistemas INEX expandido a partir dos literais**



**Figura 7.22 – Precisão x Revocação (comparativo padrão com sistemas INEX)**

De fato, os tópicos que usamos para a coleção INEX-LOD são peculiares. Como seguem o estilo do jogo *Jeopardy* de perguntas e respostas, eles normalmente só possuem uma “resposta” válida, e os termos dessa resposta aparecem com muita frequência nos valores dos atributos da consulta SPARQL; a “resposta” é exatamente o recurso da tripla RDF. Daí a pouca diferença quando da remoção de campos, e a quase coincidência entre os desempenhos de DBFIRE e os demais métodos de expansão.

## 7.6 Comentando os resultados obtidos

Depois de tantos dados e gráficos, o que realmente transparece dos experimentos aqui apresentados?

Em primeiro lugar, voltemos às hipóteses que procuramos comprovar, apresentadas inicialmente no Capítulo 1: gostaríamos que DBFIRE apresentasse resultados melhores que uma busca sem expansão, que fosse superior a outros métodos de integração SGBD/SRI, e que sua proposta para ordenação de termos se mostrasse superior a outras disponíveis na literatura.

Dessas três hipóteses, uma delas foi comprovadamente verificada em todos os experimentos: a melhoria de DBFIRE frente a outros métodos de integração (SCORE e SEMEX, no caso). Com isso podemos dizer: efetuar integração SGBD/SRI via expansão de consultas é uma opção superior.

Já a superioridade do método com relação à busca sem expansão foi verificada na maior parte dos experimentos. Além disso, se considerarmos os bons resultados frente

aos sistemas INEX em alguns cenários, temos essa conclusão reforçada.

É preciso lembrar que esses métodos tiveram algumas vantagens frente a DBFIRE: por ocasião dos workshops, os sistemas participantes poderiam ser “treinados” (via aprendizado de máquina [31]) com dados de workshops anteriores, e eles foram desenhados para aproveitarem ao máximo tais coleções. Por outro lado, DBFIRE também teve seus benefícios: utilizou consultas criadas manualmente (caso dos macro-cenários 1 e 2). Mesmo assim, o fato de DBFIRE estar bem posicionado entre eles, especialmente na coleção INEX-DC, sendo competitivo entre os três melhores métodos dentre os nove participantes, também é positivo.

De qualquer forma, houve situações em que mesmo o *baseline* superou DBFIRE. No entanto, isso aconteceu num ambiente em que as consultas não foram otimizadas, o que não seria o foco pensado para o método. Mesmo os resultados negativos no macro-cenário 3, foram revertidos se analisados pela métrica *Bpref*. Então, acreditamos que conseguimos verificar essa hipótese.

Por fim, ao se comparar DBFIRE com outros métodos de ordenação de termos, tivemos várias situações de aparente equivalência entre eles, mas várias também com evidente superioridade de DBFIRE. Vamos nos ater apenas ao método que obteve melhores resultados no geral: o método RM.

Vejam alguns exemplos das situações de bom e mau desempenho de DBFIRE: o tópico com a maior diferença percentual favorável a DBFIRE e, de forma análoga, o tópico mais desfavorável a DBFIRE. Para simplificar, consideremos essas situações na configuração padrão dos comparativos e em dois macro-cenários apenas. Assim, na Tabela 7.23 temos um detalhamento relativo ao macro-cenário 1, enquanto a Tabela 7.24 apresenta o mesmo quadro, mas agora aplicado ao macro-cenário 3.

As tabelas mostram os tópicos, os 10 melhores termos de cada método (DBFIRE e RM) incluindo seus pesos para o SRI, assim como os valores da métrica MAP para cada método.

Perceba-se que há termos sugeridos por ambos os métodos (mas com pesos diferentes), assim como termos exclusivamente sugeridos por um ou por outro método. Do ponto de vista da métrica MAP, é possível constatar discrepâncias acentuadas entre os dois métodos, ao analisarmos apenas casos isolados.

<b>Tópico</b>	<b>Top-10 Termos de Expansão - DBFIRE (com pesos)</b>	<b>Top-10 Termos de Expansão – RM (com pesos)</b>	<b>MAP (DBFIRE)</b>	<b>MAP (RM)</b>
<b>king kong jack black</b>	0.5000 2005 0.0968 usa 0.0800 imdb 0.0800 1969 0.0800 hermosa 0.0800 www 0.0800 beach 0.0800 california 0.0800 http 0.0800 jacob	0.5000 2005 0.2105 usa 0.1949 title 0.1949 thomas 0.1949 beach 0.1949 www 0.1949 hermosa 0.1949 jacob 0.1949 california 0.1949 august	<b>0.5000</b>	<b>0.2500</b>
<b>vietnam jungle</b>	0.5000 war 0.4500 com 0.4500 www 0.4500 title 0.4500 2009 0.4500 imdb 0.4500 http 0.4375 story 0.3125 journey 0.3000 time	0.5000 helicopter 0.4005 2009 0.2500 wars 0.1445 http 0.1445 imdb 0.1445 com 0.1445 www 0.1445 title 0.1406 anonymous 0.1365 usa	<b>0.3625</b>	<b>0.5014</b>

Tabela 7.23 – DBFIRE x RM (macro-cenário 1)

<b>Tópico</b>	<b>Top-10 Termos de Expansão - DBFIRE (com pesos)</b>	<b>Top-10 Termos de Expansão – RM (com pesos)</b>	<b>MAP (DBFIRE)</b>	<b>MAP (RM)</b>
<b>academy award movie celebrity couples</b>	0.5000 film 0.2525 allen 0.2240 woody 0.0923 american 0.0853 tracy 0.0661 spencer 0.0304 043 0.0287 york 0.0248 actor 0.0213 john	0.5000 allen 0.4709 woody 0.3049 film 0.1797 american 0.1629 043 0.0876 hall 0.0846 winners 0.0833 play 0.0813 jewish 0.0806 york	<b>0.0731</b>	<b>0.0388</b>
<b>greatest guitarist</b>	0.5000 album 0.2910 rock 0.2124 music 0.2029 david 0.1584 guitar 0.1502 guitarists 0.1054 american 0.1054 john 0.0987 gilmour 0.0987 band	0.5000 rock 0.2811 guitarists 0.2784 person 0.2756 american 0.2712 people 0.2542 living 0.2418 blue 0.2313 stephens 0.2313 leigh 0.2313 cheer	<b>0.2719</b>	<b>0.4156</b>

Tabela 7.24 – DBFIRE x RM (macro-cenário 3)

# Capítulo 8 - Conclusões e Trabalhos Futuros

Apresentamos o método DBFIRE voltado à integração entre SGBDs e SRIs. O método permite a recuperação de documentos fortemente ligados a consultas a BDs, aplicando o paradigma de expansão de buscas à integração entre SGBDs e SRIs.

A primeira contribuição do método é a utilização do resultado de uma consulta a BD como fonte de termos de busca ao SRI. Esses termos, associados a palavras-chave fornecidas pelo usuário, constituem sentenças de busca mais aptas a recuperar documentos relevantes no SRI do que outros métodos de integração SGBD/SRI. Temos assim a expansão de buscas orientada a consultas a BD, utilizando resultados de consultas como corpus de expansão.

A segunda contribuição consiste na forma de ordenação dos termos candidatos à expansão, focada na estimativa de sua difusão ao longo do resultado da consulta a BD. A estimativa se baseia nas distribuições das ocorrências dos termos exclusivamente no corpus de expansão. Essa forma de ordenação também se mostrou eficaz em comparação com outros métodos de ordenação de termos.

Em conjunto, essas contribuições proporcionam um arcabouço bastante útil para a recuperação de documentos relevantes à necessidade de informação expressa na consulta a BD. Pode ser facilmente implementado, seja como recurso embutido no próprio SGBD, seja mesmo no nível da aplicação.

Apesar de comprovarmos suas virtudes nos vários experimentos realizados, vemos alguns aspectos em que o método ainda pode evoluir, os quais são discutidos a seguir.

## 8.1 Incorporando outros fatores aos pesos dos termos para

## expansão

Para DBFIRE, a utilidade de um termo depende de suas probabilidades de ocorrência exclusivamente dentro do corpus de expansão. Como apresentado na Equação 4.3, são apenas duas: a probabilidade  $P_s$  de ocorrer em todo o resultado da consulta a BD em conjunto com a probabilidade  $P_e$  de ocorrer em algum elemento de tupla deste resultado.

No entanto, é possível que mais de um termo tenha os mesmos valores para essas probabilidades: por exemplo, imagine-se uma consulta em que todos os termos apareçam uma única vez; pela formulação atual de DBFIRE, todos serão igualmente importantes para a expansão, o que pode ser contestável. Como efetuar o desempate nestes casos?

Talvez aqui possamos usar fatores que são o foco de outros métodos de ordenação para diferenciar importância desses termos como, por exemplo, sua frequência na coleção completa ou sua ocorrência conjunta com as palavras-chave do usuário.

## 8.2 Atribuição de pesos diferentes para cada componente da fórmula de ordenação de termos

É possível que os componentes da fórmula de ordenação de termos (as probabilidades na sequência –  $P_s$  – e nos elementos –  $P_e$ ) possam ter pesos diferentes dependendo da consulta a BD; uma tentativa neste sentido já foi inclusive apresentada em [25], mas valendo para *todas* as consultas. Apesar de apresentar melhorias em alguns cenários, o método não é significativamente melhor que aquele apresentado aqui.

Uma possibilidade seria fazer com que a sintonia fina relativa a qual componente deva ter mais importância possa ser feita consulta a consulta, dependendo da razão entre os componentes  $P_e$  e  $P_s$ : quanto maior a proporção de um em função do outro, maior sua importância na ordenação.

## 8.3 Selecionando as tuplas com maior potencial para expansão

Nem todas as tuplas de uma consulta terão o mesmo número de termos. Mas a rigor, no mundo BD todas são igualmente relevantes.

Os resultados dos experimentos do Capítulo 7 mostraram que a exclusão de alguns campos da expansão determinou uma queda na qualidade da recuperação de documentos. Será que, de forma análoga, as tuplas com maior quantidade de termos poderão fornecer

termos mais úteis para expansão? Seria o caso de se investigar essa possibilidade.

Outra forma de determinar a utilidade de uma tupla seria a partir da maior ocorrência das palavras-chave do usuário, num mecanismo semelhante à ordenação de documentos nativamente provida por um SRI. Assim, o corpus de expansão pode ser formado apenas pelas tuplas com melhor potencial, excluindo possíveis fontes de ruídos.

## **8.4 Integração SGBDs/SRI a partir de consultas a BDs via palavras-chave**

O processamento de consultas a BD através de palavras-chave é uma área de intensa pesquisa (ver Capítulo 2). O objetivo é recuperar as tuplas mais associadas às palavras-chave fornecidas sem a necessidade de formalismos SQL, ou conhecimento de esquemas de BDs.

Uma vez recuperadas as tuplas, seria possível usá-las como corpus de expansão dentro do arcabouço de DBFIRE. Isso evitaria a necessidade da existência de uma consulta convencional ao BD, normalmente desenvolvida por especialistas, abrindo a possibilidade de o usuário final (leigo, por suposição) beneficiar-se do arcabouço para expansão proposto aqui.

A pergunta que se faz é: esse corpus seria tão útil como aquele para o qual DBFIRE foi originalmente pensado (consultas estruturadas a BDs)? Ou ainda: esse corpus traria melhores resultados do que a expansão de buscas tradicional via PRF?

## **8.5 Variando o peso máximo dos termos para expansão dependendo da consulta**

Como foi visto no Capítulo 7, dentre os parâmetros usados por DBFIRE, o valor do peso máximo atribuído a um termo para expansão (parâmetro  $\beta$ ) é o que tem maior impacto nos resultados. Sugeriu-se utilizar um valor que imaginamos seja adequado à maioria das situações, mas é possível que ele possa ser ajustado automaticamente, consulta a consulta. A questão é: que fator deve ser usado para este ajuste?

Podem-se usar diversas variáveis dentre as já citadas aqui para outras finalidades: quantidade de palavras-chave do usuário difundidas ao longo do resultado da busca, ou mesmo a relação entre as probabilidades  $P_s$  e  $P_e$ , por exemplo. Quaisquer que sejam os fatores a ser utilizados, é possível efetuar treinamento via métodos de aprendizado de



máquina [31] em cima das coleções de teste disponíveis para se estabelecer uma correlação entre essas variáveis e o aumento/diminuição do valor de  $\beta$  com relação à melhoria da qualidade na recuperação de documentos.

## **8.6 Explorar recursos da própria linguagem de consulta estruturada**

É possível que a consulta estruturada apresente recursos que não foram explorados até o momento, como ordenação do resultado da consulta, por exemplo (através de cláusulas do tipo ORDER BY). Esta é uma informação importante: subentende-se que as primeiras tuplas têm maior importância para o usuário. Neste caso, pode-se dar maior peso aos termos nas primeiras das tuplas resultantes, pois em tese eles também teriam mais importância para o usuário.

## **8.7 Reconhecimento de frases nominais**

É fato que a recuperação através de frases nominais apresenta uma maior precisão do que se usarmos os termos isoladamente. Por exemplo, o resultado de uma busca por *social network* tem resultados bem diferentes do que por “*social network*”. No entanto, a qualidade vai depender do que o usuário deseja. Assim, o uso de frases nominais favorece os resultados quando se sabe qual a frase nominal que se deseja, ou seja, quando é o usuário determinando as frases nominais.

Fazer com que DBFIRE utilize frases nominais como átomos de expansão pode melhorar os resultados, mas isso depende do reconhecimento correto das frases nominais dentro do texto. Já há trabalhos nessa linha, mas voltados à RI convencional [87]. Como será que a ideia funcionaria dentro de DBFIRE?

## **8.8 Novos experimentos**

Os resultados não-significativos verificados em alguns cenários de testes no Capítulo 7, evidenciaram a necessidade de testes adicionais para se confirmar os benefícios de DBFIRE frente a outros métodos concorrentes, notadamente os diferentes métodos de ordenação de termos. Além disso, pode-se mesmo questionar até que ponto as coleções de testes usadas aqui seriam representativas das situações reais de uso de DBFIRE. Ou se a interferência direta (como na construção manual das consultas, por exemplo) não afetaram os resultados obtidos.

Uma possibilidade para se relativizar o possível viés devido à criação manual das consultas, seria passar a tarefa para terceiros, por exemplo, alunos da disciplina de Bancos de Dados do curso de Ciência da Computação. Eles seriam os encarregados de construir as consultas tanto para o macro-cenário 1 (no qual as consultas são feitas em SQL) como para o macro-cenário 3 (com consultas em SPARQL).

Uma outra opção seria montar um ambiente de teste numa organização real, a partir de seus documentos e BDs. Essa alternativa oferecería uma conclusão mais precisa sobre o desempenho de DBFIRE, utilizando consultas reais (sem o viés da criação manual), possivelmente extraídas dos logs do próprio SGBD organizacional.

Mesmo a etapa de avaliação de relevância dos documentos recuperados (fase mais custosa de todo o processo) poderia se valer de alternativas para a avaliação de baixo custo [1, 21]. Assim, é possível fazê-lo com um pequeno grupo de juízes em pouco tempo e com alta confiança. Com isso, se teria uma dimensão mais apropriada dos benefícios de DBFIRE numa situação real de uso.

## **Anexo A - Consultas SQL para a Coleção INEX-DC**

Topic: 2011101

Literals: social network

```
SELECT DISTINCT * FROM movie as M
where match(M.title, M.plot, M.tagline) against ("social
network" in boolean mode)
```

Topic: 2011102

Literals: best movie award James Cameron

```
SELECT DISTINCT * FROM movie as M, person as P, person_movie as
PM, trivias as T
where M.idmovie=PM.idmovie and P.idperson=PM.idperson and
(T.idmovie=M.idmovie or T.idperson=P.idperson) and PM.role=2
and match(P.name) against ('+James +Cameron' in boolean mode)
and match(T.trivia) against ('+best +movie +award' in boolean
mode)
```

Topic: 2011103

Literals: James Bond

```
SELECT DISTINCT * FROM person as P, person_movie as PM
where P.idperson=PM.idperson and PM.role=1
and match(P.name) against ("James Bond" in boolean mode)
```

Topic: 2011104

Literals: Ellen Page thriller

```
SELECT DISTINCT * FROM movie as M, person as P, person_movie as
PM, genres as G
where M.idmovie=PM.idmovie and P.idperson=PM.idperson and
G.idmovie=M.idmovie and PM.role=1
and P.name='Ellen Page' and G.genre='Thriller'
```

Topic: 2011105

Literals: king kong jack black

```
SELECT DISTINCT * FROM movie as M, person as P, person_movie as
PM
where M.idmovie=PM.idmovie and P.idperson=PM.idperson and
PM.role=1
and match(P.name) against ('+Jack +Black' in boolean mode) and
match(M.title) against ("King Kong" in boolean mode)
```

Topic: 2011106

Literals: Terry Gilliam Benicio del Toro gonzo

```
SELECT DISTINCT * FROM movie as M, person as P1, person as P2,  
person_movie as PM1, person_movie as PM2  
where M.idmovie=PM1.idmovie and M.idmovie=PM2.idmovie and  
P1.idperson=PM1.idperson and P2.idperson=PM2.idperson and  
PM1.role=1 and PM2.role=2  
and match(P2.name) against ('+Terry +Gilliam' in boolean mode)  
and  
match(P1.name) against ('+Benicio +del +Toro' in boolean mode)  
and  
match(PM1.charac) against ('+Dr +gonzo' in boolean mode)
```

Topic: 2011107

Literals: Tom Hanks

```
SELECT DISTINCT * FROM person as P, biographies as B  
where P.idperson=B.idperson  
and match(B.biography) against ('"Tom Hanks"' in boolean mode)
```

Topic: 2011108

Literals: artificial intelligent Haley Joel Osment

```
SELECT DISTINCT * FROM person as P2, movie as M, person as P1,  
person_movie as PM1, person_movie as PM2  
where M.idmovie=PM1.idmovie and M.idmovie=PM2.idmovie and  
P1.idperson=PM1.idperson and P2.idperson=PM2.idperson and  
PM1.role=1 and PM2.role=2  
and match(P1.name) against ('+Haley +Joel +Osment' in boolean  
mode) and  
match(M.title) against ('artificial intelligent' in boolean  
mode)
```

Topic: 2011109

Literals: Don Quixote

```
SELECT DISTINCT * FROM movie as M, trivias as T  
where M.idmovie=T.idmovie  
and match(T.trivia) against ('"Don Quixote"' in boolean mode)
```

Topic: 2011111

Literals: french france

```
SELECT DISTINCT * FROM movie as M, countries as C, languages as  
L  
where C.idmovie=M.idmovie and L.idmovie=M.idmovie  
and L.language='French' and C.country='France' and M.year>=1990  
order by year
```

Topic: 2011112

Literals: food usa

```
SELECT DISTINCT * FROM movie as M, countries as C  
where M.idmovie=C.idmovie  
and C.country='USA' and match(M.plot) against('food' in boolean  
mode)
```

Topic: 2011113

Literals: Paul Verhoeven Arnold Schwarzenegger

```
SELECT DISTINCT * FROM movie as M, person as P1, person as P2,  
person_movie as PM1, person_movie as PM2  
where M.idmovie=PM1.idmovie and M.idmovie=PM2.idmovie and  
P1.idperson=PM1.idperson and P2.idperson=PM2.idperson and  
PM2.role=2  
and match(P2.name) against ('+Paul +Verhoeven' in boolean mode)  
and  
match(P1.name) against ('+Arnold +Schwarzenegger' in boolean  
mode)
```

Topic: 2011114

Literals: computer animation

```
SELECT DISTINCT * FROM movie as M, genres as G  
where M.idmovie=G.idmovie  
and G.genre='Animation' and match(M.plot) against ('+computer  
+animation' in boolean mode)
```

Topic: 2011115

Literals: aliens usa

```
SELECT DISTINCT * FROM movie as M  
where match(M.plot) against ('+aliens +usa' in boolean mode)
```

Topic: 2011116

Literals: action biker

```
SELECT DISTINCT * FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Action' and  
match(M.plot) against ('+biker' in boolean mode)
```

Topic: 2011117

Literals: animation fairy-tale

```
SELECT DISTINCT * FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Animation' and  
match(M.plot) against ('"fairy-tale"' in boolean mode)
```

Topic: 2011118

Literals: musical webber

```
SELECT DISTINCT * FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Musical' and  
match(M.plot) against ('+webber' in boolean mode)
```

Topic: 2011119

Literals: baseball usa

```
SELECT DISTINCT * FROM movie as M  
where match(M.plot) against ('+baseball +usa' in boolean mode)
```

Topic: 2011120

Literals: Vietnam war true story

```
SELECT DISTINCT * FROM movie as M  
where match(M.plot) against ('+Vietnam +war +true +story' in  
boolean mode)
```

Topic: 2011122  
Literals: Chernobyl  
SELECT DISTINCT \* FROM movie as M, locations as L  
where L.idmovie=M.idmovie  
and match (L.location) against ('+Chernobyl' in boolean mode)  
and  
match(M.plot) against ('+Chernobyl' in boolean mode)

Topic: 2011124  
Literals: survive desert island  
SELECT DISTINCT \* FROM movie as M  
where match(M.title, M.plot, M.tagline) against ('+survive  
+desert +island' in boolean mode)

Topic: 2011127  
Literals: Maureen Lipman mother  
SELECT DISTINCT \* FROM movie as M, person as P, person\_movie as  
PM  
where M.idmovie=PM.idmovie and P.idperson=PM.idperson and  
PM.role=1  
and match(PM.charac) against ('+mother' in boolean mode) and  
match(P.name) against ('+Maureen +Lipman' in boolean mode)

Topic: 2011128  
Literals: lovers arctic Spanish  
SELECT DISTINCT \* FROM movie as M, aliases as A, languages as L  
where M.idmovie=A.idmovie and L.idmovie=M.idmovie  
and L.language='Spanish' and  
match (A.alias) against ('+arctic +lovers' in boolean mode )

Topic: 2011129  
Literals: Alien  
SELECT DISTINCT \* FROM movie as M  
Where M.year<=1970 and match(M.plot) against ('+alien' in  
boolean mode)

Topic: 2011130

Literals: Dogme 95

```
SELECT DISTINCT * FROM movie as M, countries as C
where M.idmovie=C.idmovie
and match(M.plot) against('+dogme +95' in boolean mode)
and C.country <> 'Albania' and C.country <> 'Andorra' and
C.country <> 'Armenia' and C.country <> 'Austria' and C.country
<> 'Belarus' and C.country <> 'Belgium'
and C.country <> 'Bosnia Herzegovina' and C.country <>
'Bulgaria' and C.country <> 'Croatia' and C.country <> 'Cyprus'
and C.country <> 'Czech Republic' and C.country <>
'Czechoslovakia'
and C.country <> 'Denmark' and C.country <> 'East Germany' and
C.country <> 'Estonia' and C.country <> 'Federal Republic of
Yugoslavia' and C.country <> 'Finland' and C.country <> 'France'
and C.country <> 'Georgia' and C.country <> 'Germany' and
C.country <> 'Gibraltar' and C.country <> 'Greece' and C.country
<> 'Hungary' and C.country <> 'Iceland'
and C.country <> 'Ireland' and C.country <> 'Italy' and
C.country <> 'Liechtenstein' and C.country <> 'Lithuania' and
C.country <> 'Luxembourg' and C.country <> 'Malta'
and C.country <> 'Moldova' and C.country <> 'Monaco' and
C.country <> 'Montenegro' and C.country <> 'Netherlands' and
C.country <> 'Norway' and C.country <> 'Poland'
and C.country <> 'Portugal' and C.country <> 'Republic of
Macedonia' and C.country <> 'Romania' and C.country <> 'Russia'
and C.country <> 'San Marino' and C.country <> 'Serbia'
and C.country <> 'Serbia Montenegro' and C.country <> 'Slovakia'
and C.country <> 'Slovenia' and C.country <> 'Soviet Union' and
C.country <> 'Spain' and C.country <> 'Sweden'
and C.country <> 'Switzerland' and C.country <> 'UK' and
C.country <> 'Ukraine' and C.country <> 'West Germany' and
C.country <> 'Yugoslavia'
```

Topic: 2011131

Literals: Fellowship Ring Return King Towers

```
SELECT DISTINCT * FROM person as P, movie as M1, movie as M2,
movie as M3, person_movie as PM1, person_movie as PM2,
person_movie as PM3
where M1.idmovie=PM1.idmovie and P.idperson=PM1.idperson and
PM1.role=1 and PM1.idmovie>0 and PM1.idperson >0 and
M2.idmovie=PM2.idmovie and P.idperson=PM2.idperson and
PM2.role=1 and PM2.idmovie>0 and PM2.idperson >0 and
M3.idmovie=PM3.idmovie and P.idperson=PM3.idperson and
PM3.role=1 and PM3.idmovie>0 and PM3.idperson >0
and match(M1.title) against('"The Fellowship of the Ring"' in
boolean mode) and
match(M2.title) against('"The Return of the King"' in boolean
mode) and
match(M3.title) against('"The Two Towers"' in boolean mode)
```



Topic: 2011133  
Literals: titleFriends  
Friends tv-series  
SELECT DISTINCT \* FROM person as P, movie as M, person\_movie as PM  
PM  
where P.idperson=PM.idperson and M.idmovie=PM.idmovie  
and M.year=1994 and PM.role=1 and  
match(M.url) against ('"Title?Friends "' in boolean mode)

Topic: 2011135  
Literals: Cannes documentary  
SELECT DISTINCT \* FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Documentary' and  
match(M.plot) against ('+Cannes' in boolean mode)

Topic: 2011136  
Literals: trained dolphin whale  
SELECT DISTINCT \* FROM movie as M  
where match(M.plot) against ('+trained +dolphin' in boolean  
mode) or match(M.plot) against ('+trained +whale' in boolean  
mode)

Topic: 2011137  
Literals: romance Leonardo DiCaprio Tom Cruise  
SELECT DISTINCT \* FROM movie as M, genres as G, person as P,  
person\_movie as PM  
where G.idmovie=M.idmovie and P.idperson=PM.idperson and  
M.idmovie=PM.idmovie and PM.role=1  
and G.genre='Romance' and (match(P.name) against ('+Leonardo  
+DiCaprio' in boolean mode) or (match(P.name) against ('+Tom  
+Cruise' in boolean mode) )

Topic: 2011138  
Literals: ancient Egyptian  
SELECT DISTINCT \* FROM movie as M  
where match(M.plot) against ('+ancient +Egypt' in boolean mode)

Topic: 2011140  
Literals: food industry documentary  
SELECT DISTINCT \* FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Documentary' and  
match(M.plot) against ('+food +industry' in boolean mode)

Topic: 2011141  
Literals: Vietnam jungle  
SELECT DISTINCT \* FROM movie as M, locations as L  
where L.idmovie=M.idmovie  
and match(L.location) against ('+Vietnam' in boolean mode) and  
match(M.plot) against ('jungle' in boolean mode)

Topic: 2011142  
Literals: Vietnam  
SELECT DISTINCT \* FROM person as P  
where match(P.birth\_date) against ('Vietnam' in boolean mode)

Topic: 2011143  
Literals: Documentary usa President  
Documentaries US Presidents  
SELECT DISTINCT \* FROM movie as M, genres as G  
where G.idmovie=M.idmovie  
and G.genre='Documentary' and  
match(M.plot) against ('+usa +President' in boolean mode)

Topic: 2011144  
Literals: Cannes jury  
SELECT DISTINCT \* FROM person as P, trivias as T  
where P.idperson=T.idperson  
and match(T.trivia) against ('+Cannes +jury' in boolean mode)

Topic: 2011145  
Literals: Palme winner  
SELECT DISTINCT \* FROM movie as M, trivias as T  
where M.idmovie=T.idmovie  
and match(T.trivia) against ('+Palme +winner' in boolean mode)

## **Anexo B - Consultas SPARQL para a Coleção INEX-LOD**

```
Topic: 2013302
Literals: river north dakota
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/River> .
?subject <http://dbpedia.org/ontology/region>
<http://dbpedia.org/resource/North_Dakota> .
<http://dbpedia.org/resource/North_Dakota>
<http://dbpedia.org/property/lowestpoint> ?point .
filter regex(xsd:string(?point),
replace(replace(xsd:string(?subject), "_", " "),
"http://dbpedia.org/resource/", ""), "i")) .
}
```

```
Topic: 2013303
Literals: president indonesia 1921
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/President> .
?subject <http://dbpedia.org/ontology/deathPlace> ?deathPlace .
?subject <http://dbpedia.org/ontology/birthDate> ?birthDate .
filter(regex(?deathPlace, "Indonesia", "i")) .
FILTER (year(?birthDate) = 1921) .
}
```

```
Topic: 2013304
Literals: Millard Fillmore 1850 7
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/ontology/deathDate> ?death .
?subject <http://dbpedia.org/ontology/successor>
<http://dbpedia.org/resource/Millard_Fillmore> .
FILTER (year(?death) = 1850) .
FILTER (month(?death) = 7) .
}
```

```
Topic: 2013305
Literals: Malay speaking countries and territories Island
chinese population
SELECT DISTINCT ?subject2 ?property2 ?object2 WHERE {
?subject1 ?property1 ?object1 .
?subject2 ?property2 ?object2 .
?subject1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/Malay-
speakingCountriesAndTerritories> .
?subject1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/IslandCountries> .
?subject2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/Malay-
speakingCountriesAndTerritories> .
?subject2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/IslandCountries> .
}
```

```
FILTER contains(lcase(xsd:string(?object1)), "chinese" ) .
FILTER contains(lcase(xsd:string(?object1)), "population" ) .
FILTER (?subject1=?subject2 ) .
}
```

Topic: 2013306

Literals: just like starting over lennon yoko

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/property/title> ?o .
?subject <http://dbpedia.org/property/artist> ?a .
FILTER contains (lcase(xsd:string(?o)), "just") .
FILTER contains (lcase(xsd:string(?o)), "like") .
FILTER contains (lcase(xsd:string(?o)), "starting") .
FILTER contains (lcase(xsd:string(?o)), "over") .
FILTER contains (lcase(xsd:string(?a)), "yoko") .
FILTER contains (lcase(xsd:string(?a)), "lennon") .
}
```

Topic: 2013307

Literals: 8 2 eight half director

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?film <http://dbpedia.org/ontology/director> ?subject .
FILTER regex (?film, "8") .
FILTER regex (?film, "2") .
?film ?property2 ?object2 .
FILTER regex(?object2, "eight", "i") .
FILTER regex(?object2, "half", "i") .
FILTER regex(?object2, "director", "i") .
}
```

Topic: 2013308

Literals: john turturro coen brothers 1991

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/ontology/starring>
<http://dbpedia.org/resource/John_Turturro> .
?subject <http://dbpedia.org/ontology/director>
<http://dbpedia.org/resource/Coen_brothers> .
?subject ?p ?o .
FILTER regex(?o, "1991") .
}
```

Topic: 2013309

Literals: lucky jim

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/property/author> ?subject .
FILTER contains (lcase(xsd:string(?x)), "lucky") .
FILTER contains (lcase(xsd:string(?x)), "jim") .
}
```

Topic: 2013310

Literals: baguio manila Quezon city

```
SELECT DISTINCT ?subject ?property ?object WHERE {
```

```
?subject ?property ?object .
?x <http://dbpedia.org/ontology/country> ?subject .
FILTER regex(ucase(xsd:string(?x)),
"\bbaguio\b|\bmanila\b|\bquenzon\b.*\bcity\b") .
}
```

Topic: 2013311

Literals: central america

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/property/data> ?city .
?city <http://dbpedia.org/ontology/country> ?subject .
FILTER regex(?x, "Central", "i") .
FILTER regex(?x, "America", "i") .
}
```

Topic: 2013312

Literals: sons and lovers

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/ontology/author> ?subject .
FILTER regex(?x, "Sons", "i") .
FILTER regex(?x, "and", "i") .
FILTER regex(?x, "Lovers", "i") .
}
```

Topic: 2013313

Literals: glen glenda bride monster plan outer space

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/ontology/director> ?subject .
?y <http://dbpedia.org/ontology/director> ?subject .
?z <http://dbpedia.org/ontology/director> ?subject .
FILTER regex (?x, "Glen", "i") .
FILTER regex (?x, "Glenda", "i") .
FILTER regex (?y, "Bride", "i") .
FILTER regex (?y, "Monster", "i") .
FILTER regex (?z, "Plan", "i") .
FILTER regex (?z, "Outer", "i") .
FILTER regex (?z, "Space", "i") .
}
```

Topic: 2013314

Literals: Short story collections by Alice Munro open

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject ?property1
<http://dbpedia.org/resource/Category:Short_story_collections_by_Alice_Munro> .
?subject ?property2 ?object2 .
FILTER regex(?object2, "Open") .
}
```

Topic: 2013315

Literals: Pickwick Papers boz

```
SELECT DISTINCT ?subject ?property ?object WHERE {
```

```
?subject ?property ?object .
?title <http://dbpedia.org/ontology/author> ?subject .
FILTER regex (?title, "Pickwick","i") .
FILTER regex (?title, "Papers","i") .
?title ?y ?z .
FILTER regex (?z, "boz","i") .
}
```

Topic: 2013316

```
Literals: City-states Sir Stamford Raffles asian port
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:City-states> .
?subject ?founded ?founder .
filter regex(?founder, "Sir", "i") .
filter regex(?founder, "Stamford", "i") .
filter regex(?founder, "Raffles", "i") .
?subject ?y ?z .
filter regex(?z, "asian", "i") .
filter regex(?z, "port", "i") .
}
```

Topic: 2013317

```
Literals: Mona Lisa Paris
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?piece <http://dbpedia.org/property/museum> ?subject .
?subject ?property1 ?location .
FILTER regex(?piece, "Mona", "i") .
FILTER regex(?piece, "Lisa", "i") .
FILTER regex(?location, "paris", "i") .
}
```

Topic: 2013318

```
Literals: executive mansion james blaine
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/ontology/location> ?location .
?state <http://dbpedia.org/ontology/capital> ?location .
?subject ?p1 ?o1 .
filter regex(?o1, "executive.*mansion","i") .
filter regex(?o1, "james.*blaine","i") .
}
```

Topic: 2013319

```
Literals: james blaine studied law
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/ontology/successor> ?subject .
FILTER regex (?x, "James.*Blaine", "i") .
?subject ?p1 ?o1 .
filter regex(?o1, "studied.*law","i") .
}
```

Topic: 2013320

Literals: mindanao Muslim

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?z <http://dbpedia.org/ontology/country> ?subject .
FILTER regex (?z, "Mindanao", "i") .
?subject ?p ?o .
FILTER regex (?o, "Muslim", "i") .
}
```

Topic: 2013321

Literals: beloved nobel laureates literature african american

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?book <http://dbpedia.org/property/author> ?subject .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/NobelLaureatesInLiterature> .
FILTER regex (?book , "beloved", "i") .
?subject ?prop ?obj .
FILTER regex (?obj , "african-american", "i") .
}
```

Topic: 2013322

Literals: capital native name wien

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/CapitalsInEurope> .
?subject <http://dbpedia.org/property/nativeName> ?name .
FILTER regex (?name, "wien", "i") .
}
```

Topic: 2013323

Literals: Iceland

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?country <http://dbpedia.org/ontology/currency> ?subject .
FILTER regex (?country, "Iceland", "i") .
}
```

Topic: 2013324

Literals: active member kneset Ambassadors Israel United States

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/AmbassadorsOfIsraelToTheUnitedSta
tes> .
?subject ?property1 ?object1
filter regex(?object1, "active.*member.*kneset","i") .
}
```

Topic: 2013325

Literals: Seoul River

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/River> .
}
```



```
?subject ?x <http://dbpedia.org/resource/Seoul>
}
```

Topic: 2013326

```
Literals: longest unbroken term office Prime Ministers Canada
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/PrimeMinistersOfCanada> .
?subject ?y ?z .
FILTER regex (?z, "longest.*unbroken.*term.*office", "i") .
}ontolo
```

Topic: 2013327

```
Literals: An American Tragedy tragic america
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
<http://dbpedia.org/resource/An_American_Tragedy>
<http://dbpedia.org/ontology/author> ?subject .
?subject ?y ?z .
filter regex(?z, "tragic.*america", "i") .
}
```

Topic: 2013328

```
Literals: Presidents United States atomic weapons against Japan
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/PresidentsOfTheUnitedStates> .
?subject ?x ?y .
FILTER regex (?y, "atomic.*weapons.*against.*Japan", "i") .
}
```

Topic: 2013329

```
Literals: 1906 territory Papua australia
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Country> .
?subject ?y ?z .
FILTER regex (?z, "1906.*.*territory.*Papua", "i") .
?subject ?y1 ?z1 .
FILTER regex (?z1, "australia", "i") .
}
```

Topic: 2013330

```
Literals: baylor University tornado 1953
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x1 <http://dbpedia.org/ontology/city> ?subject .
?subject ?y ?x2 .
FILTER regex (?x1, "Baylor.*University", "i") .
FILTER regex (?x2, "tornado.*1953|1953.*tornado", "i") .
}
```

Topic: 2013331

Literals: DePaul University 1955 1976 mayor

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/property/almaMater>
<http://dbpedia.org/resource/DePaul_University> .
?subject <http://dbpedia.org/property/termStart> ?start .
?subject <http://dbpedia.org/property/title> ?title .
?subject <http://dbpedia.org/property/termEnd> ?end.
FILTER regex(xsd:string(?start),"1955") .
FILTER regex(xsd:string(?end),"1976") .
FILTER regex(xsd:string(?title),"mayor","i") .
}
```

Topic: 2013332

Literals: Grosny

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/property/admCtrName>
<http://dbpedia.org/resource/Grosny> .
}
```

Topic: 2013333

Literals: 1997 Houston airport Buildings and monuments honoring American Presidents

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Airport> .
?subject <http://dbpedia.org/ontology/location>
<http://dbpedia.org/resource/Houston> .
?subject ?p1
<http://dbpedia.org/resource/Category:Buildings_and_monuments_ho
noring_American_Presidents> .
?subject ?p2 ?o2 .
filter regex (?o2,"1997") .
}
```

Topic: 2013334

Literals: university cathedral Notre Dame

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?uni <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/University> .
?uni <http://dbpedia.org/ontology/city> ?subject .
?uni ?p1 ?o1 .
filter regex(?o1, "notre.*dame", "i") .
filter regex(?o1, "catedral", "i") .
}
```

Topic: 2013335

Literals: The Heart of a Woman autobiography

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/ontology/author> ?s .
?x <http://dbpedia.org/ontology/author> ?s .
}
```

```

FILTER regex (?x, "The.*Heart.*of.*a.*Woman", "i") .
?subject ?p1 ?o1 .
FILTER regex (?o1, "autobiography", "i") .
}

```

Topic: 2013337

```

Literals: Kennedy assassination governors of Texas
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/GovernorsOfTexas> .
?subject ?p1 ?o1 .
FILTER regex (?o1, "Kennedy.*assassination", "i") .
}

```

Topic: 2013338

```

Literals: Undershaw The Hound of the Baskervilles
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?x <http://dbpedia.org/ontology/author> ?subject .
?x ?p ?o .
FILTER regex (?x, "The.*Hound.*of.*the.*Baskervilles", "i") .
<http://dbpedia.org/resource/Undershaw> ?p1 ?o1 .
FILTER regex (?o1, replace(replace(xsd:string(?subject), "_", "" ), "http://dbpedia.org/resource/", ""), "i") .
}

```

Topic: 2013339

```

Literals: Florida county Dade
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?county <http://dbpedia.org/ontology/countySeat> ?subject .
?county <http://dbpedia.org/ontology/state>
<http://dbpedia.org/resource/Florida> .
?county ?y ?z .
filter regex(?z, "dade?county", "i") .
}

```

Topic: 2013340

```

Literals: Stadsholmen Riddarholmen
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
<http://dbpedia.org/resource/Stadsholmen> ?p1 ?s .
<http://dbpedia.org/resource/Riddarholmen> ?p2 ?s .
?s <http://www.w3.org/2000/01/rdf-schema#label> ?label .
?x <http://dbpedia.org/ontology/capital> ?subject .
filter regex(xsd:string(?label), concat("\\b",
replace(replace(xsd:string(?subject), "_", "" ),
"http://dbpedia.org/resource/", ""), "\\b"), "i") .
}

```

Topic: 2013341

```

Literals: Alexander Nevsky Cathedral
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?o <http://dbpedia.org/property/province> ?subject .
}

```

```
FILTER regex (?o, "Alexander.*Nevsky.*Cathedral", "i") .
}
```

Topic: 2013342

Literals: Summa Theologica

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/property/notableWorks> ?work.
FILTER regex (?work, "Summa.*Theologica", "i") .
}
```

Topic: 2013343

Literals: Indian Cuisine rice dal vegetables roti papad

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject ?p
<http://dbpedia.org/resource/Category:Indian_cuisine> .
?subject ?p1 ?o .
FILTER regex (?o, "rice" , "i") .
FILTER regex (?o, "dal", "i") .
FILTER regex (?o, "vegetables", "i") .
FILTER regex (?o, "roti", "i") .
FILTER regex (?o, "papad", "i") .
}
```

Topic: 2013344

Literals: One Day Cricket Indian Test Captain Best Crickets scholar

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/IndianTestCaptains> .
?subject ?p ?o .
FILTER regex (?o, "one.*day.*Cricket", "i") .
?subject ?p1 ?o1 .
FILTER regex (?o1, "best.*Crickets", "i") .
?subject ?p2 ?o2 .
FILTER regex (?o2, "scholar", "i") .
}
```

Topic: 2013345

Literals: country predominant German language

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Country> .
?subject <http://dbpedia.org/ontology/language>
<http://dbpedia.org/resource/German_language> .
?subject ?p ?o .
filter regex(?o, "predominant", "i") .
}
```

Topic: 2013346

Literals: greatest guitarist of all time

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
}
```

```
?subject <http://dbpedia.org/property/instrument> ?instrument .
?subject ?p ?o .
filter regex (?instrument, "\\bguitar\\b", "i") .
FILTER regex (?o, "greatest guitarist.*of all time", "i") .
}
```

Topic: 2013347

```
Literals: England national football player highest paid
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/ontology/team>
<http://dbpedia.org/resource/England_national_football_team> .
?subject ?p ?o .
FILTER regex (?o, "highest-paid", "i") .
}
```

Topic: 2013348

```
Literals: prima ballerina Bolshoi 1960
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/SpanishBalletDancers> .
?subject ?p ?o .
FILTER regex (?o, "prima.*ballerina.*Bolshoi.*1960", "i") .
}
```

Topic: 2013350

```
Literals: Academy award won winn
SELECT DISTINCT ?subject1 ?subject2 ?object WHERE {
?subject1 <http://dbpedia.org/property/partner> ?subject2 .
?subject1 <http://dbpedia.org/property/occupation> ?occl .
?subject2 <http://dbpedia.org/property/occupation> ?occ2 .
?m <http://dbpedia.org/ontology/starring> ?subject1 .
?m <http://dbpedia.org/ontology/starring> ?subject2.
?m ?p ?o2
FILTER regex(?o2, "won.*Academy.*Award|Academy.*Award.*winn",
"i") .
FILTER regex(?occl, "actor", "i") .
FILTER regex(?occ2, "actress", "i") .
?subject1 ?property1 ?object .
?subject2 ?property2 ?object .
}
```

Topic: 2013351

```
Literals: award singer actor actress
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://dbpedia.org/property/occupation> ?occl.
?subject <http://dbpedia.org/property/occupation> ?occ2.
filter regex(?occl,"actor|actress","i") .
filter regex(?occ2,"singer","i") .
?subject ?p ?o .
filter regex(?o,"award","i") .
}
```

Topic: 2013352

Literals: rock festival Australia Beenie Man Odd Future  
controversy

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/class/yago/RockFestivalsInAustralia> .
OPTIONAL {
?subject ?p ?o .
FILTER regex (?o, "Odd.*Future", "i") .
FILTER regex (?o, "Beenie.*Man", "i") .
FILTER regex (?o, "controversy", "i") .
}
}
```

Topic: 2013353

Literals: twin people from united states professional tennis  
double players

```
SELECT DISTINCT ?subject ?property ?object WHERE {
?subject ?property ?object .
?subject <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Twin_people_from_the_United_States> .
?subject ?p ?o .
FILTER regex (?o, "professional", "i") .
FILTER regex (?o, "tennis.*double.*players", "i") .
}
```

## **Anexo C – Versão SQL das Consultas SPARQL da Coleção INEX-LOD**

```
2013302
river north dakota
river north dakota
SELECT M.lowestpoint
FROM M AS M
WHERE M.type='River' and M.region='North Dakota'
```

```
2013303
president indonesia 1921
president indonesia 1921
SELECT M.name
FROM M AS M
WHERE M.type='President' and M.deathPlace='Indonesia' and
M.birthDate=1921
```

```
2013304
Millard Fillmore 1850 7
Millard Fillmore 1850 7
SELECT *
FROM M AS M
WHERE M.deathDate=1850 AND M.predecessor='Millard Fillmore'
```

```
2013305
Malay speaking countries and territories Island chinese
population
Malay speaking countries and territories Island chinese
population
SELECT *
FROM M AS M
WHERE M.type='Malay speaking Countries & Territories' and
M.type='Island Countries' and M.language='chinese'
```

```
2013306
just like starting over lennon yoko
just like starting over lennon yoko
SELECT M.album
FROM M AS M
WHERE M.title='just like starting over' and M.artist='yoko' and
M.artist='lennon'
```

```
2013307
8 2 eight half director
8 2 eight half director
SELECT *
FROM M AS M
WHERE M.director='eight half'
```



```
2013308
john turturro coen brothers 1991
john turturro coen brothers 1991
SELECT *
FROM M AS M
WHERE M.starring='John Turturro' AND M.director='Coen brothers'
AND M.releaseDate=1991
```

```
2013309
lucky jim
lucky jim
SELECT M.author
FROM M AS M
WHERE M.title='lucky jim'
```

```
2013310
baguio manila Quezon city
baguio manila Quezon city
SELECT M.country
FROM M AS M
WHERE M.city='baguio' and M.city='manila' and M.city='quezon
city'
```

```
2013311
central america
central america
SELECT M.country
FROM M AS M
WHERE M.continent='Central America'
```

```
2013312
sons and lovers
sons and lovers
SELECT M.author
FROM M AS M
WHERE M.title='Sons & Lovers'
```

```
2013313
glen glenda bride monster plan outer space
glen glenda bride monster plan outer space
SELECT M.director
FROM M AS M
WHERE M.title='Glen Glenda' AND M.title='bride monster' AND
M.title='plan outer space'
```

```
2013314
Short story collections by Alice Munro open
Short story collections by Alice Munro open
SELECT *
FROM M AS M
WHERE M.Category='Short story collections by Alice Munro' AND
M.title='Open'
```

```
2013315
Pickwick Papers boz
Pickwick Papers boz
SELECT M.author
from M AS M
WHERE M.title='Pickwick Papers' and M.alternativeNames='boz'
```

```
2013316
City-states Sir Stamford Raffles asian port
City-states Sir Stamford Raffles asian port
SELECT *
from M AS M
WHERE M.Category='City states' M.founder='Sir Stamford Raffles'
and M.location='asia' and M.type='port'
```

```
2013317
Mona Lisa Paris
Mona Lisa Paris
SELECT M.museum
from M AS M
WHERE M.painting='Mona Lisa' and M.location='paris'
```

```
2013318
executive mansion james blaine
executive mansion james blaine
SELECT *
from M AS M
WHERE M.type='executive mansion' and M.type='capital' and
M.resident='james blaine'
```

```
2013319
james blaine studied law
james blaine studied law
SELECT *
FROM M AS M
WHERE M.predecessor = 'James Blaine' and M.course='law'
```

```
2013320
mindanao Muslim
mindanao Muslim
SELECT M.country
FROM M AS M
WHERE M.island='Mindanao' and M.religion='Muslim' and
M.religion='Roman Catholic'
```

```
2013321
beloved nobel laureates literature african american
beloved nobel laureates literature african american
SELECT M.author
FROM M AS M
WHERE M.title='beloved' and M.type='Nobel Laureates in
Literature' and M.origin='african american'
```

```
2013322
capital native name wien
capital native name wien
SELECT *
FROM M AS M
WHERE M.type = 'Capitals In Europe' and M.nativeName='wien'
```

```
2013323
Iceland
Iceland
SELECT M.currency
FROM M AS M
WHERE M.country='Iceland' and M.country='sweden'
```

```
2013324
active member kneset Ambassadors Israel United States
active member kneset Ambassadors Israel United States
SELECT *
FROM M AS M
WHERE M.type='Ambassadors Of Israel To The United States' and
M.memberof='kneset'
```

```
2013325
Seoul River
Seoul River
SELECT M.name
FROM M AS M
WHERE M.type='River' and M.city='Seoul'
```

2013326

```
longest unbroken term office Prime Ministers Canada
longest unbroken term office Prime Ministers Canada
SELECT *
FROM M AS M
WHERE M.type='Prime Ministers Of Canada' and M.title='longest
unbroken term office'
```

2013327

```
An American Tragedy tragic america
An American Tragedy tragic america
SELECT M.author
FROM M AS M
WHERE M.title='An American Tragedy' and M.title ='tragic
america'
```

2013328

```
Presidents United States atomic weapons against Japan
Presidents United States atomic weapons against Japan
SELECT *
FROM M AS M
WHERE M.type='Presidents Of The United States' and
M.bombed='Japan'
```

2013329

```
1906 territory Papua australia
1906 territory Papua australia
SELECT *
FROM M AS M
WHERE M.type='Country' and M.territoryOf='australia' and
M.foundationdate=1906
```

2013330

```
baylor University tornado 1953
baylor University tornado 1953
SELECT M.city
FROM M AS M
WHERE M.university='baylor university' and M.event='tornado
1953'
```

2013331

```
DePaul University 1955 1976 mayor
DePaul University 1955 1976 mayor
SELECT M.name
FROM M AS M
WHERE M.almaMater='DePaul University' and M.termStart=1955 and
M.termEnd=1976 and M.title='mayor'
```

```
2013332
Grosny
Grosny
SELECT M.name
FROM M AS M
WHERE M.capital='Grosny'
```

```
2013333
1997 Houston airport Buildings and monuments honoring American
Presidents
1997 Houston airport Buildings and monuments honoring American
Presidents
SELECT *
from M AS M
WHERE M.type='Airport' and M.location='Houston' and
M.Category='Buildings & monuments honoring American Presidents'
M.renamedIn=1997
```

```
2013334
university cathedral Notre Dame
university cathedral Notre Dame
SELECT M.city
from M AS M
WHERE M.type='University' and M.location='notre dame catedral'
```

```
2013335
The Heart of a Woman autobiography
The Heart of a Woman autobiography
SELECT M.author
from M AS M
WHERE M.type='autobiography' and M.title='The Heart of a Woman'
```

```
2013337
Kennedy assassination governors of Texas
Kennedy assassination governors of Texas
SELECT M.name
from M AS M
WHERE M.type='Governors Of Texas' and M.event='Kennedy
assassination'
```

```
2013338
Undershaw The Hound of the Baskervilles
Undershaw The Hound of the Baskervilles
SELECT M.author
from M AS M
WHERE M.title='The Hound of the Baskervilles' and
M.location='Undershaw'
```

```
2013339
Florida county Dade
Florida county Dade
SELECT M.countySeat
from M AS M
WHERE M.location='florida' M.knownas='dade'
```

```
2013340
Stadsholmen Riddarholmen
Stadsholmen Riddarholmen
SELECT *
from M AS M
WHERE M.island='Stadsholmen' and M.island='Riddarholmen'
M.type='capital'
```

```
2013341
Alexander Nevsky Cathedral
Alexander Nevsky Cathedral
SELECT M.name
FROM M AS M
WHERE M.type='city' and M.country='bulgaria' and
M.building='Alexander Nevsky Cathedral'
```

```
2013342
Summa Theologica
Summa Theologica
SELECT *
from M AS M
WHERE M.type='saint' and M.notableWorks='Summa Theologica'
```

```
2013343
Indian Cuisine rice dal vegetables roti papad
Indian Cuisine rice dal vegetables roti papad
SELECT *
FROM M AS M
WHERE M.Category='Indian cuisine' and M.ingredients='rice' and
M.ingredients='dal' and M.ingredients='vegetables' and
M.ingredients='roti' and M.ingredients='papad'
```

```
2013344
One Day Cricket Indian Test Captain Best Crickets scholar
One Day Cricket Indian Test Captain Best Crickets scholar
SELECT *
from M AS M
WHERE M.type='Indian Test Captains' and M.sport='Cricket' and
M.knownas='India Best Schoolboy Cricketer'
```

```
2013345
country predominant German language
country predominant German language
SELECT *
from M AS M
WHERE M.type='Country' and M.language = 'German'
```

```
2013346
greatest guitarist of all time
greatest guitarist of all time
SELECT *
from M AS M
WHERE M.instrument='guitar' and M.knownas='greatest guitarist of
all time'
```

```
2013347
England national football player highest paid
England national football player highest paid
SELECT *
from M AS M
WHERE M.team='England national football team' and
M.knownas='highest paid football player'
```

```
2013348
prima ballerina Bolshoi 1960
prima ballerina Bolshoi 1960
SELECT *
FROM M AS M
WHERE M.type='Spanish Ballet Dancers' and M.title='prima
ballerina' and M.date=1960 and M.ballet='Bolshoi'
```

```
2013350
Academy award won winn
Academy award won winn
SELECT M.title
from M AS M
WHERE M.occupation='actress' and M.occupation='actor' and
M.award='Academy Award'
```

```
2013351
award singer actor actress
award singer actor actress
SELECT *
from M AS M
WHERE M.occupation='actor' or M.occupation='actress' and
M.occupation='singer' and M.prize='award'
```

```
2013352
rock festival Australia Beenie Man Odd Future controversy
rock festival Australia Beenie Man Odd Future controversy
SELECT *
from M AS M
WHERE M.type='Rock Festivals In Australia' and
M.participant='Odd Future' and M.participant='Beenie Man'
```

```
2013353
twin people from united states professional tennis double
players
twin people from united states professional tennis double
players
SELECT *
from M AS M
WHERE M.category='Twin people from the UnitedStates' and
M.profession='tennis double players'
```



## Referências

- [1] Allan, J. et al. 2006. Minimal Test Collections for Retrieval Evaluation. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR'06* (2006), 268–275.
- [2] Amati, G. and Rijsbergen, C.J. Van 2002. Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Transactions on Information Systems*. 20, 4 (2002), 357–389.
- [3] Amer-yahia, S. et al. 2008. DB & IR Integration : Report on the Dagstuhl Seminar “ Ranked XML Querying ” 1. *ACM SIGMOD Record*. 42, 2 (2008), 84–89.
- [4] Amer-Yahia, S. et al. 2005. Report on the DB/IR panel at SIGMOD 2005. *ACM SIGMOD Record*. 34, 4 (Dec. 2005), 71–74.
- [5] Amer-yahia, S. and Lalmas, M. 2006. XML Search : Languages , INEX and Scoring. *ACM SIGMOD Record*. 35, 4 (2006), 16–23.
- [6] Arcoverde, J.M.A. et al. 2006. Using Noun Phrases for Local Analysis in Automatic Query Expansion. *CLEF (Working Notes) 2006* (Alicante, Spain, 2006).
- [7] Aslam, J.A. and Yilmaz, E. 2005. A geometric interpretation and analysis of R-precision. *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM'05* (New York, NY, USA, 2005), 664–671.
- [8] Baeza-yates, R. and Lalmas, M. 2006. SIGIR 2006 Tutorial: XML Information Retrieval. <http://www.dcs.gla.ac.uk/~mounia/XMLIR.pdf> (2006), 1–44.
- [9] Bergamaschi, S. et al. 2013. Keyword Search and Evaluation over Relational Databases: an Outlook to the Future. *Proceedings of the 7th International Workshop on Ranking in Databases - DBRank '13* (2013).
- [10] Bhalotia, G. et al. 2002. Keyword Searching and Browsing in Databases using BANKS. *Proceedings of the 18th International Conference on Data Engineering - ICDE'02* (2002), 431–440.

- [11] Billerbeck, B. and Zobel, J. 2003. When Query Expansion Fails. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03* (Toronto, Canada, 2003), 387–388.
- [12] Booch, G. et al. 2005. *The Unified Modeling Language User Guide*. Addison-Wesley Professional.
- [13] Buckley, C. and Voorhees, E.M. 2000. Evaluating Evaluation Measure Stability. *Proceedings of the 23rd annual international conference on Research and development in information retrieval - SIGIR '00* (2000), 33–40.
- [14] Buckley, C. and Voorhees, E.M. 2004. Retrieval Evaluation with Incomplete Information. *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04* (New York, New York, USA, 2004), 25–32.
- [15] Cafarella, M.J. et al. 2007. Structured Querying of Web Text: A Technical Challenge. *3rd Biennial Conference on Innovative Data Systems Research - CIDR 2007* (Asilomar, USA, 2007), 225–234.
- [16] Cao, G. et al. 2008. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08* (Singapore, 2008), 243–250.
- [17] Carpineto, C. et al. 2001. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*. 19, 1 (Jan. 2001), 1–27.
- [18] Carpineto, C. et al. 2002. Improving Retrieval Feedback with Multiple Term-Ranking Function Combination. *ACM Transactions on Information Systems*. 20, 3 (2002), 259–290.
- [19] Carpineto, C. and Romano, G. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Computing Surveys*. 44, 1 (Jan. 2012), 1–50.
- [20] Carpineto, C. and Romano, G. 1999. Towards more effective techniques for automatic query expansion. *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries, ECDL '99* (1999), 1–16.
- [21] Carterette, B. 2007. Robust Test Collections for Retrieval Evaluation. *SIGIR 07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (Amsterdam, The Netherlands, 2007), 55–62.
- [22] Cartright, M. et al. 2010. Fast Query Expansion Using Approximations of Relevance Models. *the 19th ACM conference on Conference on information and knowledge management CIKM 10* (2010), 1573–1576.

- [23] Catão, V.S. et al. DBFIRE: Bridging the Gap Between Documents and Databases. *Journal of Information Science* (Em Revisão).
- [24] Catão, V.S. et al. 2014. Information Retrieval from Database Queries. In *Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA ' 2014)* (Doha, Qatar, 2014).
- [25] Catão, V.S. et al. 2015. Retrieving Documents related to Database Queries. *Proceedings of the 41st International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 15)* (Pec pod Sněžkou, Czech Republic, 2015).
- [26] Chaudhuri, S. et al. 2005. Integrating DB and IR Technologies : What is the Sound of One Hand Clapping? *Second Biennial Conference on Innovative Data Systems Research-CIDR '05* (Asilomar, USA, 2005), 1–12.
- [27] Coffman, J. and Weaver, A.C. 2010. A Framework for Evaluating Database Keyword Search Strategies. *Proceedings of the 19th ACM conference on Information and knowledge management, CIKM '10* (2010), 729–738.
- [28] Coffman, J. and Weaver, A.C. 2014. An Empirical Performance Evaluation of Relational Keyword Search Techniques. *IEEE Transactions on Knowledge and Data Engineering*. 26, 1 (2014), 30–42.
- [29] Coffman, J. and Weaver, A.C. 2011. Learning to Rank Results in Relational Keyword Search. *CIKM '11, Proceedings of the 20th ACM international conference on Information and knowledge management* (Glasgow, UK, 2011), 1689–1698.
- [30] Demartini, G. 2007. Leveraging Semantic Technologies for Enterprise Search. *Proceedings of the ACM first Ph.D. workshop in CIKM-PIKM '07* (Lisbon, Portugal, 2007), 25–31.
- [31] Domingos, P. 2012. A few useful things to know about machine learning. *Communications of the ACM*. 55, 10 (2012), 78–87.
- [32] Fox, C. 1992. Lexical analysis and stoplists. *Information retrieval: data structures and algorithms*. William B. Frakes and R. Baeza-Yates, eds. Prentice Hall. 102–130.
- [33] Frans Adriaans, Jaap Kamps, and M.K. 2012. The Importance of Document Ranking and User-Generated Content for Faceted Search and Book Suggestions. *Focused Retrieval of Content and Structure - Lecture Notes in Computer Science, Vol. 7424*. Springer-Verlag. 30–44.
- [34] Fuhr, N. and Lalmas, M. 2005. Introduction to the Special Issue on INEX. *Information Retrieval*. 8, 4 (2005), 515–519.
- [35] Furnas, G.W. et al. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*. 30, 11 (1987), 964–971.

- [36] Gupta, P. and Gupta, V. 2012. A Survey of Text Question Answering Techniques. *International Journal of Computer Applications*. 53, 4 (2012).
- [37] Gurajada, S. et al. 2013. Overview of the INEX 2013 Linked Data Track. *CLEF (Online Working Notes/Labs/Workshop)* (2013).
- [38] Halevy, A. et al. 2006. Principles of dataspace systems. *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems - PODS '06* (Chicago, USA, 2006), 1–9.
- [39] He, H. et al. 2007. BLINKS : Ranked Keyword Searches on Graphs. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data-SIGMOD '07* (2007), 305–316.
- [40] Hedeler, C. et al. 2009. Dimensions of Dataspaces. *Proceedings of the 26th British National Conference on Databases: Dataspace: The Final Frontier* (2009).
- [41] Hoffart, J. et al. 2013. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*. 194, (2013), 28–61.
- [42] Hristidis, V. et al. 2003. Efficient IR-Style Keyword Search over Relational Databases. *Proceedings of the 29th international conference on Very large data bases-VLDB '03* (2003), 850 – 861.
- [43] Hristidis, V. and Papakonstantinou, Y. 2002. DISCOVER : Keyword Search in Relational Databases. *Proceedings of the 28th international conference on Very Large Data Bases-VLDB '02* (2002), 670 – 681.
- [44] Indri: A language-model based search engine for complex queries (extended version): 2005. <http://ciir.cs.umass.edu/pubfiles/ir-407.pdf>. Accessed: 2014-12-15.
- [45] Jain, A. et al. 2008. Optimizing SQL Queries over Text Databases. *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering-ICDE '08* (2008), 636–645.
- [46] Jain, A. et al. 2007. SQL Queries Over Unstructured Text Databases. *2007 IEEE 23rd International Conference on Data Engineering* (Istanbul, Turkey, 2007), 1255–1257.
- [47] Jain, R.K. 1991. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley.
- [48] Jones, K.S. 1981. *Information Retrieval Experiment*. Butterworths.
- [49] Jones, K.S. and C J Van Rijsbergen 1975. *Report on the need for and provision of an ideal information retrieval test collection*. Computing Laboratory, University of Cambridge.

- [50] Kastrati, F. et al. 2011. Enabling Structured Queries over Unstructured Documents. *Proceedings of the 2011 IEEE 12th International Conference on Mobile Data Management-MDM '11* (2011), 80–85.
- [51] Kelly, D. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval*. 3, 1-2 (2009), 1–224.
- [52] Lalmas, M. and Tombros, A. 2007. INEX 2002 - 2006 : Understanding XML Retrieval Evaluation. *Proceedings of the 1st international conference on Digital libraries: research and development-DELOS'07* (Pisa, Italy, 2007), 187–196.
- [53] Lavrenko, V. and Croft, W.B. 2001. Relevance-Based Language Models. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01* (New Orleans, USA, 2001), 120–127.
- [54] Lehmann, J. et al. 2014. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*. (2014).
- [55] Li, G. et al. 2008. EASE : An Effective 3-in-1 Keyword Search Method for Unstructured , Semi-structured and Structured Data. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data-SIGMOD '08* (Vancouver, Canada, 2008), 903–914.
- [56] Liu, F. et al. 2006. Effective Keyword Search in Relational Databases. *Proceedings of the 2006 ACM SIGMOD international conference on Management of data-SIGMOD '06* (2006), 563–574.
- [57] Liu, J. et al. 2006. Answering Structured Queries on Unstructured Data. *9th International Workshop on the Web and Databases (WebDB 2006)* (Chicago, USA, 2006), 25–30.
- [58] Luk, R.W.P. et al. 2002. A survey in indexing and searching XML documents. *Journal of the American Society for Information Science and Technology*. 53, 6 (2002), 415–437.
- [59] Luo, Y. et al. 2007. Spark: top-k keyword query in relational databases. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data-SIGMOD '07* (2007), 115–126.
- [60] Macdonald, C. et al. 2005. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. *Proceedings of the The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)* (2005).
- [61] Mangold, C. et al. 2006. Symbiosis in the Intranet : How Document Retrieval Benefits from Database Information. *International Conference on Management of Data - COMAD 2006* (2006), 186–189.

- [62] Manning, C.D. et al. 2008. *An Introduction to Information Retrieval*. Cambridge University Press.
- [63] Mirza, H.T. et al. 2010. Practicability of Dataspace Systems. *International Journal of Digital Content Technology and its Applications*. 4, 3 (Jun. 2010), 233–243.
- [64] Mitra, M. et al. 1998. Improving Automatic Query Expansion. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'98* (Melbourne, Australia, 1998), 206 – 214.
- [65] Mohania, M. et al. 2009. Context Oriented Information Integration. *Transactions on Large-Scale Data And Knowledge-Centeres Systems I, Lecture Notes In Computer Science, Volume 5740*. A. Hameurlain et al., eds. Springer Berlin Heidelberg. 289–326.
- [66] Mohania, M. and Bhide, M. 2008. New Trends in Information Integration. *Proceedings of the 2nd international conference on Ubiquitous information management and communication-ICUIMC '08* (2008), 74–81.
- [67] Muezzinoglu, T. and Badia, A. 2008. Lightweight Database Wrapper for Unstructured Data. *11th International Workshop on the Web and Databases (WebDB 2008)* (2008).
- [68] Pérez-Aguera, J.R. and Araujo, L. 2008. Comparing and Combining Methods for Automatic Query Expansion. *Advances in Natural Language Processing and Applications. Research in Computing Science*. 33, (2008), 177–188.
- [69] Pham, K.C. et al. 2010. Object Search : Supporting Structured Queries in Web Search Engines. *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search* (Stroudsburg, USA, 2010), 44–52.
- [70] Plachouras, V. et al. 2004. University of Glasgow at TREC 2004: Experiments in Web, Robust and Terabyte tracks with Terrier. *Proceedings of the 13th Text REtrieval Conference TREC 2004* (2004).
- [71] Qin, L. et al. 2009. Keyword Search in Databases : The Power of RDBMS. *Proceedings of the 35th SIGMOD international conference on Management of data-SIGMOD '09* (Providence, USA, 2009), 681–694.
- [72] Real-time Query Expansion in Relevance Models: 2006. .
- [73] Robertson, S. 2004. Understanding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation*. 60, 5 (2004), 503–520.
- [74] Rocchio, J.J. 1971. Relevance feedback in information retrieval. *SMART Retrieval System - Experiments in Automatic Document Processing*. G. Salton, ed. Prentice Hall. 313–323.

- [75] Roy, P. et al. 2005. Towards Automatic Association of Relevant Unstructured Content with Structured Query Results. *Proceedings of the fourteenth ACM conference on information and knowledge management - CIKM '05* (Bremen, Germany, 2005), 405–412.
- [76] Roy, P. and Mohania, M. 2007. SCORE: symbiotic context oriented information retrieval. *Proceedings of the joint 9th Asia-Pacific web and 8th international conference on web-age information management conference on Advances in data and web - APWeb/WAIM'07 management* (2007), 30–38.
- [77] Ruthven, I. 2003. Re-examining the Potential Effectiveness of Interactive Query Expansion. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR'03* (2003), 213–220.
- [78] Salton, G. and Buckley, C. 1990. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*. 41, 4 (1990), 288–297.
- [79] Sanderson, M. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*. 4, 4 (2010), 247–375.
- [80] Sarawagi, S. 2008. Information Extraction. *Foundations and Trends in Databases*. 1, 3 (2008), 261–377.
- [81] Schlieder, T. and Meuss, H. 2002. Querying and ranking XML documents. *Journal of the American Society for Information Science and Technology*. 53, 6 (2002), 489–503.
- [82] Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*. 34, 1 (2002), 1–47.
- [83] Thom, J.A. and Scholer, F. 2007. A Comparison of Evaluation Measures Given How Users Perform on Search Tasks. *Proceedings of the 12th Australasian Document Computing Symposium* (Melbourne, Australia, 2007), 56–63.
- [84] Trivedi, K.S. 2001. *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. Wiley-Interscience.
- [85] Trotman, A. and Sigurbjörnsson, B. 2004. Narrowed Extended XPath I (NEXI). *INEX'04 Proceedings of the Third international conference on Initiative for the Evaluation of XML Retrieval* (2004), 16–40.
- [86] Trotman, A. and Sigurbjörnsson, B. 2004. NEXI, Now and Next. *INEX'04 Proceedings of the Third international conference on Initiative for the Evaluation of XML Retrieval* (2004), 41–53.
- [87] Vechtomova, O. 2006. Noun phrases in interactive query expansion and document ranking. *Information Retrieval*. 9, 4 (2006), 399–420.

- [88] Voorhees, E. and Harman, D. 2008. TREC Experiment and Evaluation in Information Retrieval. *Information Retrieval*. 11, 5 (Jun. 2008), 473–475.
- [89] Voorhees, E.M. 2002. The Philosophy of Information Retrieval Evaluation. *CLEF 01 Revised Papers from the Second Workshop of the CrossLanguage Evaluation Forum on Evaluation of CrossLanguage Information Retrieval Systems* (2002), 355–370.
- [90] Voorhees, E.M. 2005. The TREC Robust Retrieval Track. *SIGIR Forum*. 39, 1 (2005), 11–20.
- [91] Voorhees, E.M. 2005. TREC: Improving Information Access through Evaluation. *Bulletin of the American Society for Information Science and Technology*. 32, 1 (2005), 16–21.
- [92] Voorhees, E.M. and Harman, D.K. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press.
- [93] Wang, Q. et al. 2011. Overview of the INEX 2011 Data-Centric Track. *Focused Retrieval of Content and Structure - Lecture Notes in Computer Science, Vol. 7424*. S. Geva et al., eds. Springer-Verlag. 118–137.
- [94] Wang, Q. et al. 2012. RUC @ INEX 2011 Data-Centric Track. *Focused Retrieval of Content and Structure - Lecture Notes in Computer Science, Vol. 7424*. 167–179.
- [95] Webber, W. 2010. Evaluating the Effectiveness of Keyword Search. *IEEE Data Engineering Bulletin*. 33, 1 (2010), 54–59.
- [96] Weikum, G. 2007. DB & IR: Both Sides Now. *Proceedings of the 2007 ACM SIGMOD international conference on management of data-SIGMOD '07* (Beijing, China, 2007), 25–30.
- [97] Wichaiwong, T. and Jaruskulchai, C. 2012. MEXIR at INEX-2011. *Focused Retrieval of Content and Structure - Lecture Notes in Computer Science, Vol. 7424*. 180–187.
- [98] Zobel, J. 1998. How reliable are the results of large-scale information retrieval experiments? *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 98* (1998), 307–314.