

Consulta Baseada em Palavras-chave em Documentos XML

Rafael Odon de Alencar

odon@dcc.ufmg.br

20/05/2009

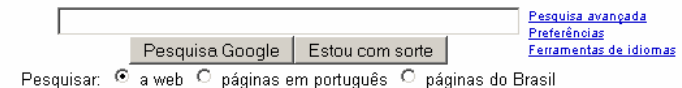
Estrutura da apresentação

1. Consulta por palavra-chave
2. Técnica aplicada ao XML
3. Abordagens discutidas:
 - *Balmin* e outros
 - *XRANK*
 - *XSearch*
4. Extensão do tema
5. Conclusões
6. Referências

1. Consulta por palavra-chave

1. Consulta por palavra-chave

- Por que utilizar?
 - Simplicidade;
 - Usuário não precisa aprender uma linguagem de consulta complexa;
 - Usuário não precisa conhecer a estrutura da base de dados consultada;
 - Permite uma interface de consulta amigável.

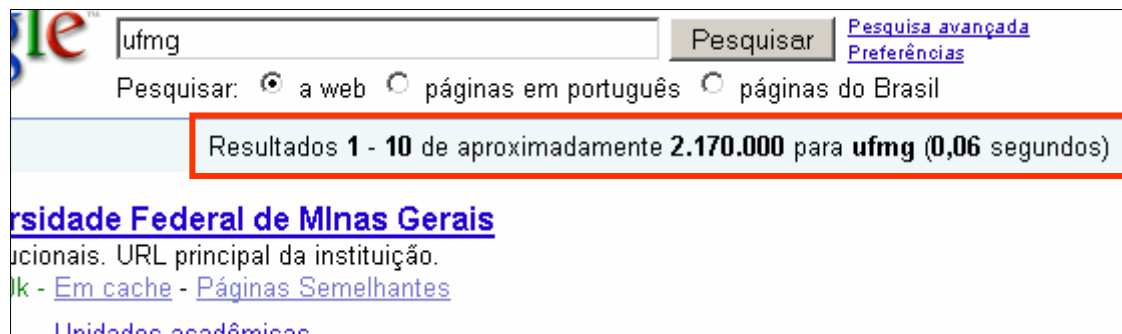


[Soluções de publicidade](#) - [Soluções empresariais](#) - [Tudo sobre o Google](#) - [Google.com in English](#)

©2009 - [Privacidade](#)

1. Consulta por palavra-chave

- As consultas não são precisas:
 - A entrada é muito flexível;
 - Pode-se retornar uma imensidão de resultados.
- Necessidade de fazer *ranking* para exibir resultados relevantes primeiro.



1. Consulta por palavra-chave

- Um paradigma efetivo na busca por documentos HTML:
 - O HTML é uma linguagem de apresentação, não dá suporte semântico à busca;
 - Consultas retornam referências para documentos completos, e não para fragmentos relevantes;
 - *Ranking* baseado em ocorrência e distância de termos e conexões entre documentos.

Canário - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

Terra dos Pássaros

Mais de 1000 produtos para pássaros
Gaiolas, Sementes e muito mais!
www.terradospassaros.com

Negócio Próprio

Ganhe R\$3000,00/mês trabalhando em casa. Trabalho parcial ou Integral.
www.negocioglobal.com.br

Tudo em Aves - Em Oferta

Encontre Variedade de Passarinhos, CDs de Canto, Ovos, Gaiolas e Mais.
www.MercadoLivre.com.br

Anúncios Google

Bicharada.net | Versão 3G | Notícias | Tópicos | Classificados | Clínicas | Lojas | Canis | Hotéis | Zoos | Parques | Contactos | Ficha | Login

Pesquisar

Web Bicharada

Pesquisa Google

Animais

☒ Animais de quinta

☒ Animais selvagens

☒ Artrópodes

☒ Aves

• Abelharuco

• Abetarda-comum

• Abutre-do-Egipto

• Abutre-preto

• Açor

• Agapornis

• Águia-das-Filipinas

• Águia-de-asas-redonda

• Águia-imperial

• Águia-real

• Albatroz-das-Galápagos

• Albatroz-de-laysan

• Albatroz-de-pés-negros

• Albatroz-viajeiro

• Alcatraz-pardo

• Andorinha

• Andorinha do mar escura

• Andorinha roxa

• Anara

• Avestruz do Norte de África

• Bico de Lacre

• Cacatua

• Canário

• Caturra

• Cisne-branco

Aves

Canário

Nome científico: *Serinus canaria*

Origem

Esta ave tem origem nas ilhas Espanholas das Canárias, na costa ocidental do continente africano, sendo por esse motivo bafejadas com os ventos quentes que vêm do Saara. O tempo agradável que durante todo o ano abençoa este território pode ter tido influência para a sua sobrevivência neste arquipélago.

Quando os primeiros navegadores trouxeram este animal para a Europa continental, estavam muito longe de saber o que iria acontecer cinco séculos depois, quando o Canário se tornou uma ave de companhia em muitas habitações, principalmente no sul da Europa, nos países mediterrânicos, e no Brasil.

O belo canto do macho depressa cativou a atenção destes homens do mar, que prontamente capturaram os primeiros espécimes, para trazer na volta para casa.

No entanto, a ave que, em geral, conhecemos com esse nome já é uma mutação dos canários originais.

A gaiola

Nesta espécie, o macho canta, julga-se que para arranjar companhia. Por esse motivo, quando não é época de criação, é normal vermos os machos separados das fêmeas, para cantarem com mais frequência. A fêmea limita-se a piar, e não com muita insistência.

Canário

Foto original (1200x957)

Wallpaper | Postal







Publicidade

Megazoo - Toda a linha

Papagaios Calopsitas Periquitos Os melhores preços, confira
www.supermercadodospets.com.br

Eduque seu CÃO -15min/dia

Método divertido de adestramento você mesmo faz em casa com seu cão
www.AdestramentoDeCoes.com

Classificados de Animais

Venda e compra animais no site de anúncios gratuitos OLX
www.olx.com.br

Anúncios Google

ANÚNCIOS

Animais

☒ Pet-Sitting (Semana, FDS e Fér (PORTO, PORTUGAL)

Gatos

☒ Sphynx ou Esfinge (Lisboa, Portugal)

Animais

☒ Doctorvet Clínica Veterinária, (Curitiba, Brasil)

Aves

☒ Seminário sobre Papagaios

NOTÍCIAS



Grã-Bretanha

☒ Preso por ter cão - gordo

Global

☒ Mande uma baleia para o Japão

Austrália

☒ ...

Concluído

1. Consulta por palavra-chave

- As consultas por palavras-chave em geral podem ser:
 - **Conjuntivas:** elementos retornados devem conter todas as palavras consultadas;
 - **Disjuntivas:** elementos retornados devem conter pelo menos uma das palavras consultadas;
- Abordagem conjuntiva retorna resultados mais interessantes.

2. Técnica aplicada ao XML

2. Técnica aplicada ao XML

- Resultados são fragmentos do XML (elementos aninhados):
 - Possibilidade de retornar trechos semanticamente relevantes, ao invés de documentos inteiros;

<pássaro>

<nome> Canário </nome>

<origem> Esta ave tem origem nas ilhas Espanholas ... </origem>

<gaiola> Nesta espécie, o macho canta, julga-se que ... </gaiola>

<alimentação> A alimentação para os canários é... </alimentação >

</pássaro>

2. Técnica aplicada ao XML

- Necessário considerar granularidade dos elementos, e não do documento como um todo;
- *Ranking* também passa a ser baseado em elementos;
- Deve-se levar em conta relação entre elementos pais e filhos (bi-direcional).
- Tratar conexões do tipo IDREF (entre elementos) e XLinks (entre documentos),

2. Técnica aplicada ao XML

- Nós mais profundos apresentam mais informação do contexto que nós mais genéricos ou abrangentes.

– O que é mais representativo?

Retornar uma conferência onde cada palavra chave aparece em algum dos milhares artigos de alguma das dezenas de edições.

X

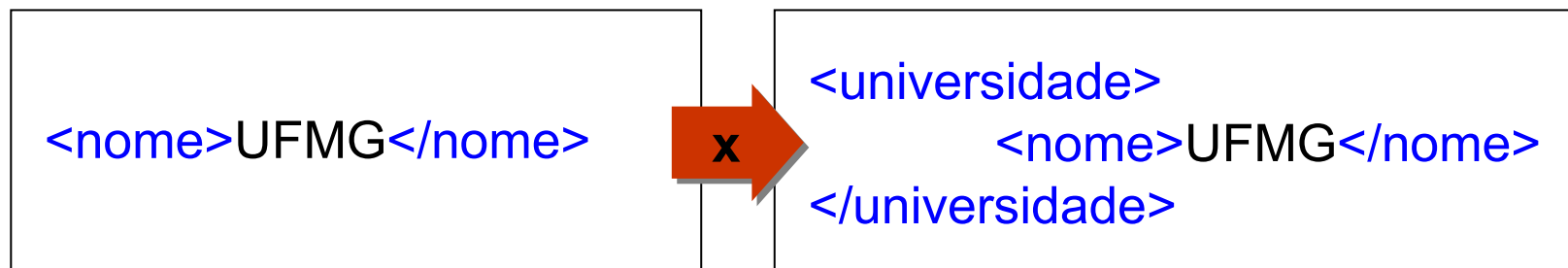
Retornar um artigo cujas palavras chaves ocorrem no título.

2. Técnica aplicada ao XML

- Proximidade de palavras-chave pode não representar muito:
 - Palavras chaves localizadas pouco distantes entre si no documento, podem pertencer a sub-elementos distintos e pouco correlatos no grafo XML.
- Métrica de proximidade de termos pode considerar **altura** e **largura** do fragmento avaliado.

2. Técnica aplicada ao XML

- Apresentação de resultados precisa significar algo para o usuário.
- Retornar apenas os nós que contém as palavras chaves não fornece informação contextual.



2. Técnica aplicada ao XML

- Pode-se permitir a navegação nos resultados para entender o contexto.
 - Necessário implementar mecanismo de navegação.
- Pode-se definir “nós de resposta”, que são fragmentos mínimos a serem retornados, contendo informações significativas:
 - Requer conhecimento do domínio e do esquema.

2. Técnica aplicada ao XML

- Deve-se avaliar a apresentação de informações redundantes:
 - Não incluir todos os ancestrais de uma resposta;
 - Avaliar tais ancestrais, pois eles podem apresentar palavras-chaves em profundidades diferentes;
 - Referências a outros elementos podem levar a caminhos redundantes e repetitivos.

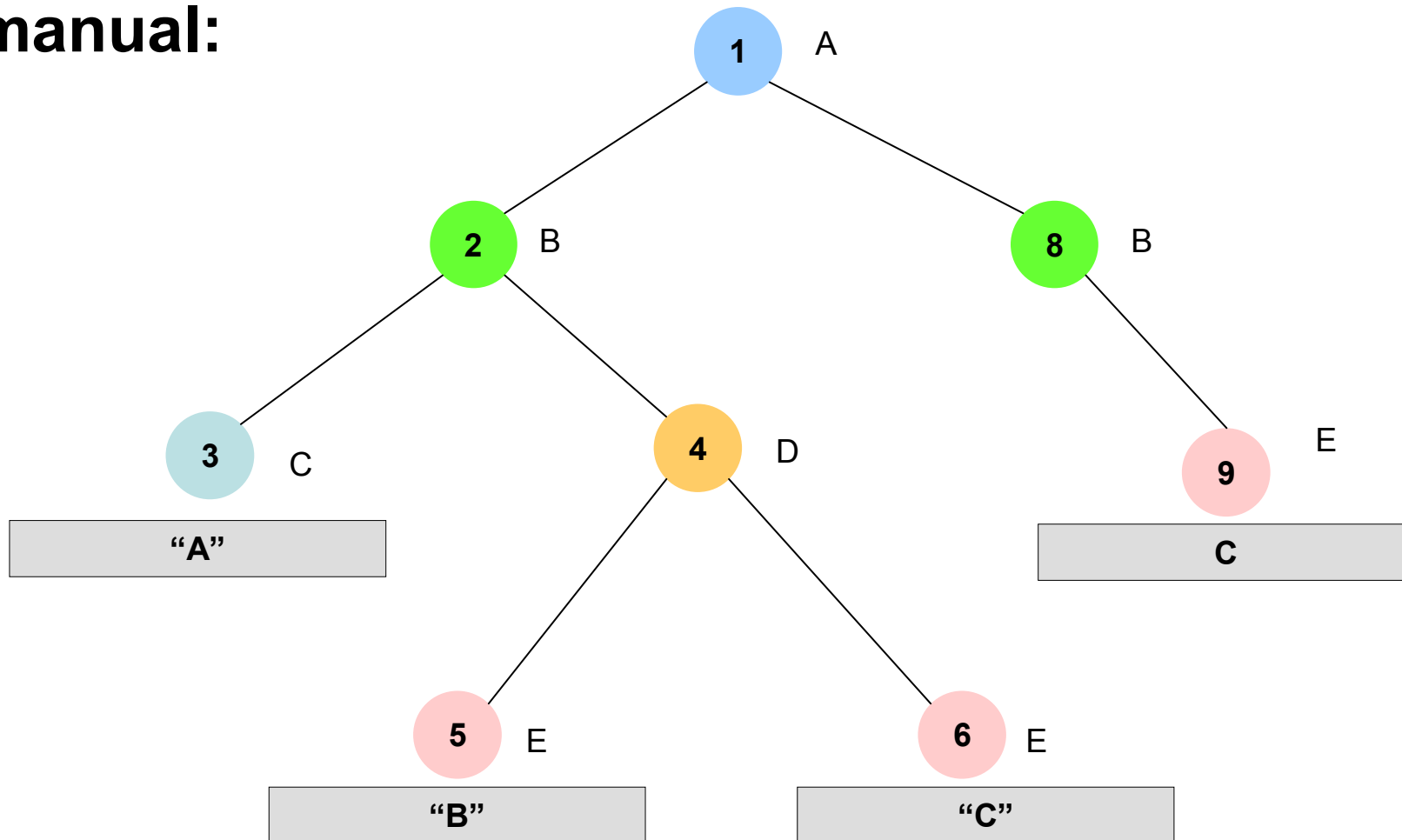
2. Técnica aplicada ao XML

- Procurar por palavras-chaves nos rótulos e nos valores dos nós:
 - Existem diferenças?
 - No que isso vai influenciar nos resultados?
- Ambigüidade de palavras-chave:
 - *Tom Cruise*:
 - Nome de uma **ator**.
 - Turista **Tom** que está num **Cruzeiro**.

2. Técnica aplicada ao XML

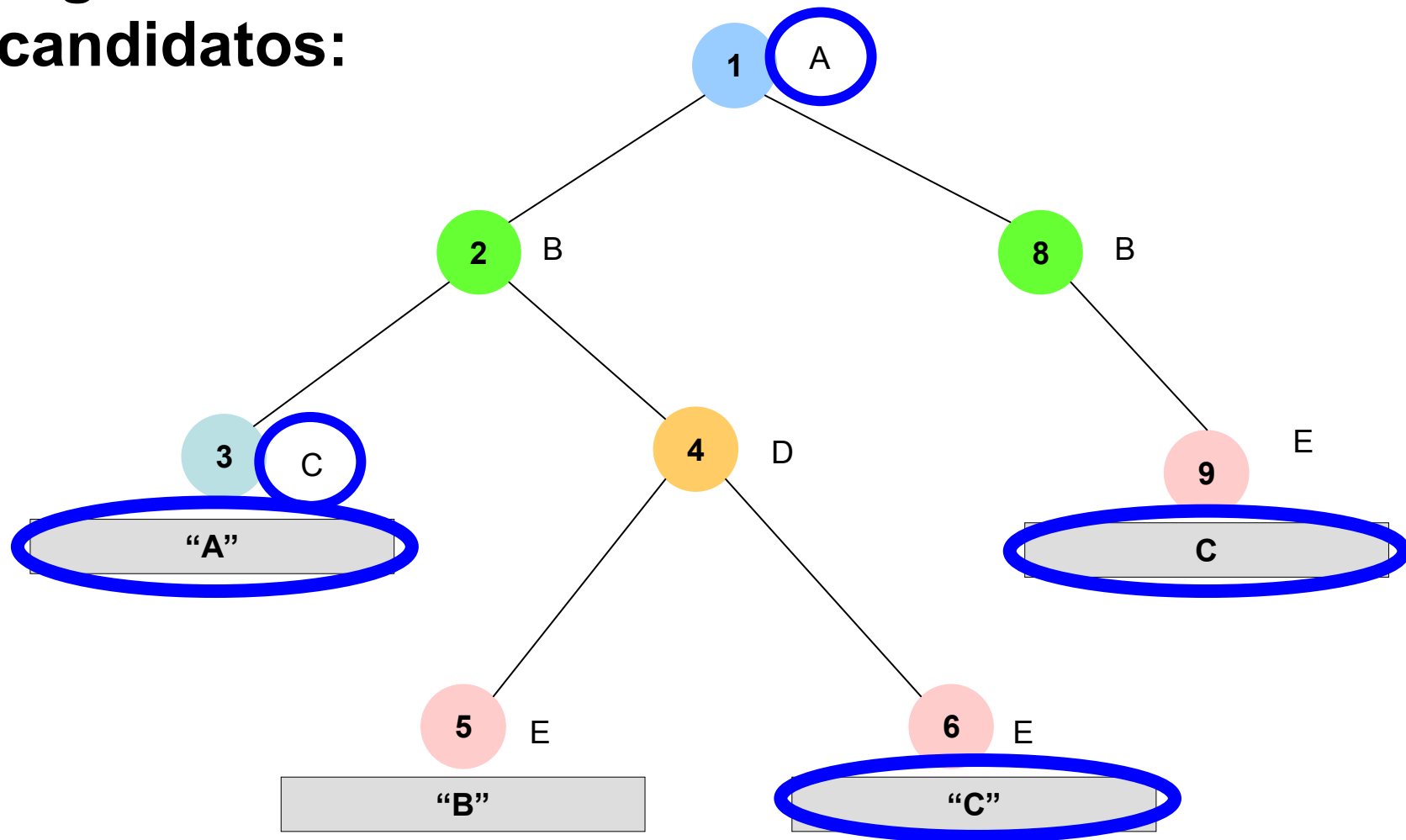
- Por que não usar XQuery para buscar por palavras-chave?
 - Sintaxe complexa;
 - Necessário conhecimento do documento;
 - Não apresenta mecanismos de *ranking*;

Uma consulta manual:



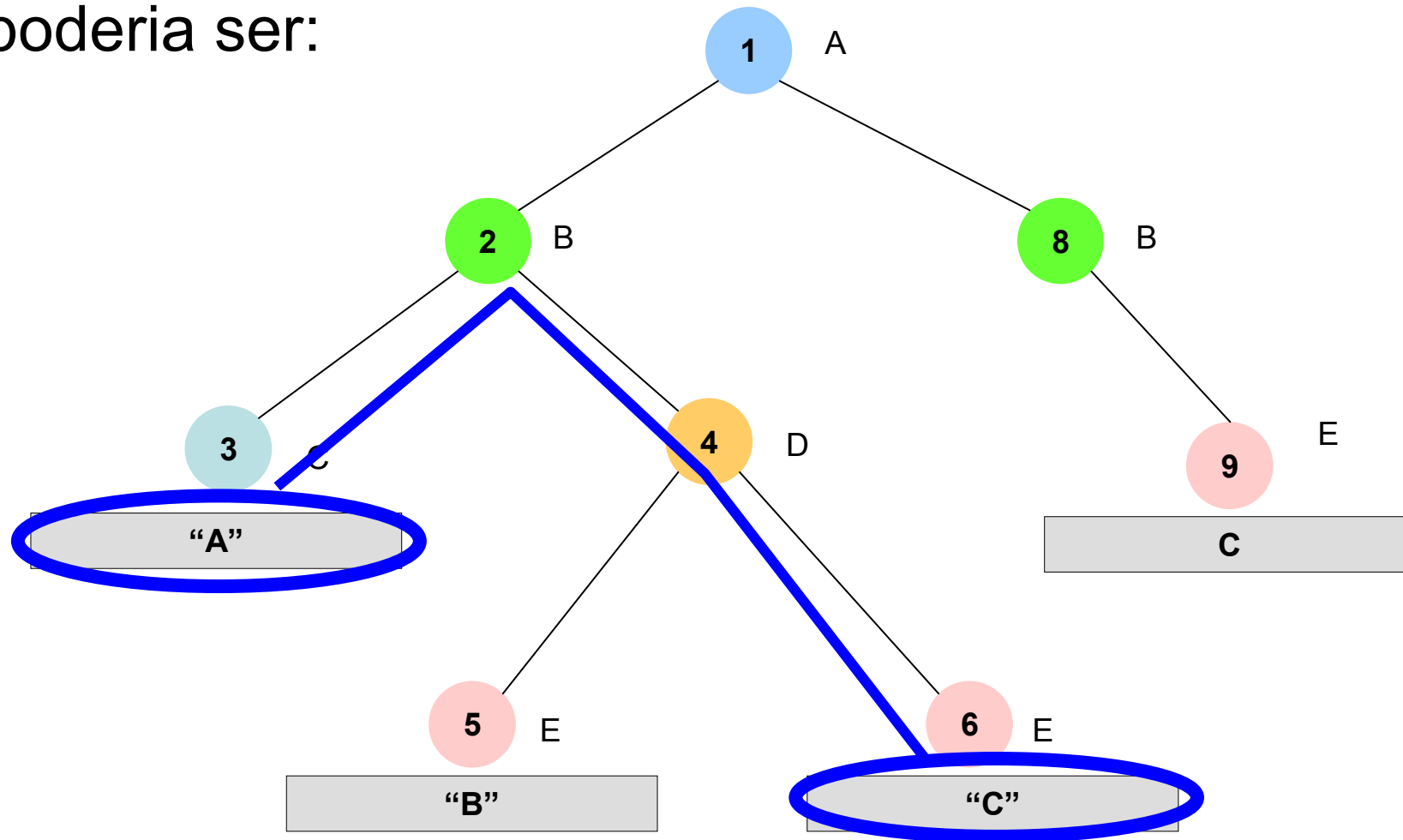
Consultar por A, C

**Alguns
candidatos:**



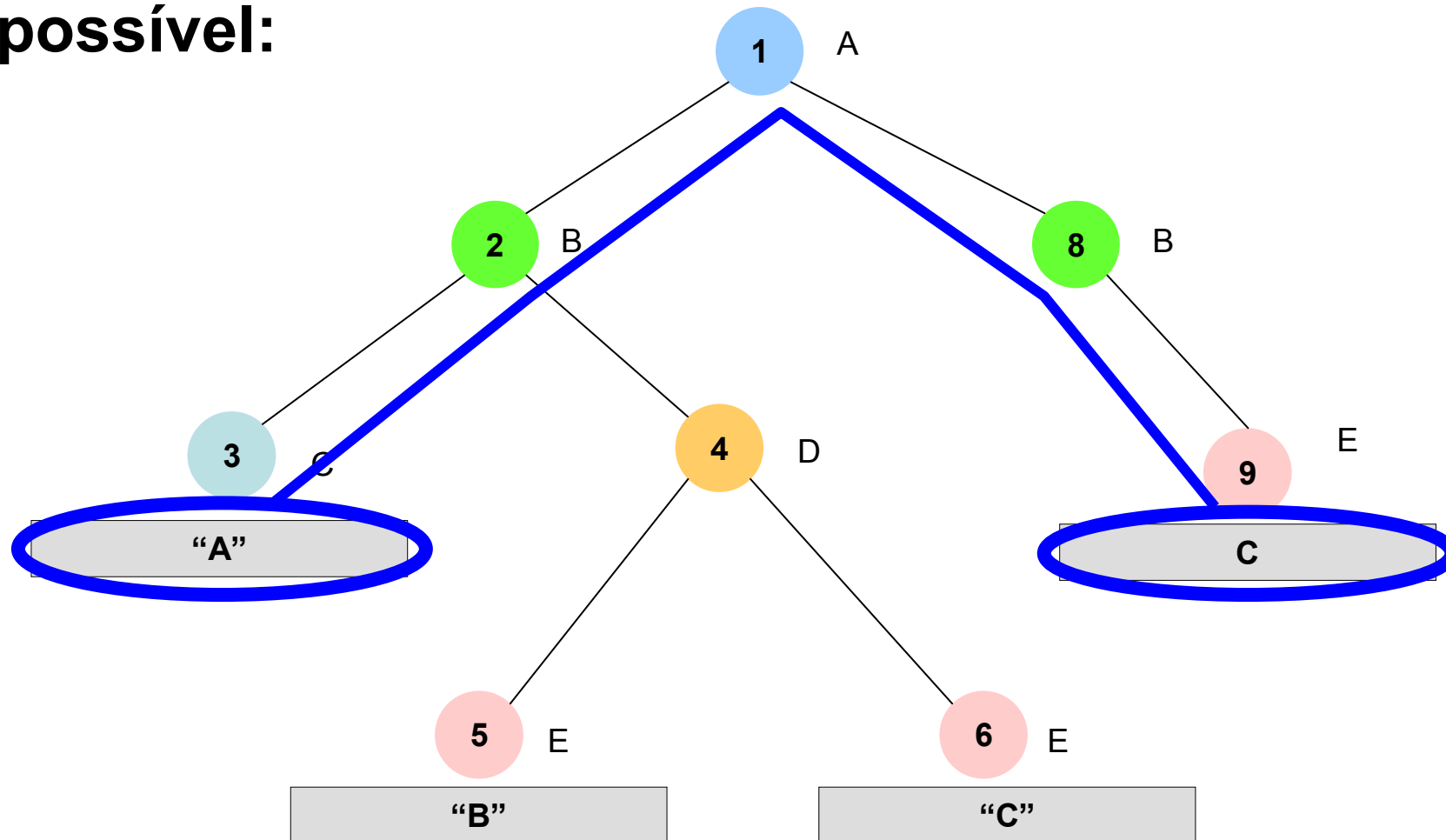
Consultar por A, C

Um resultado
poderia ser:



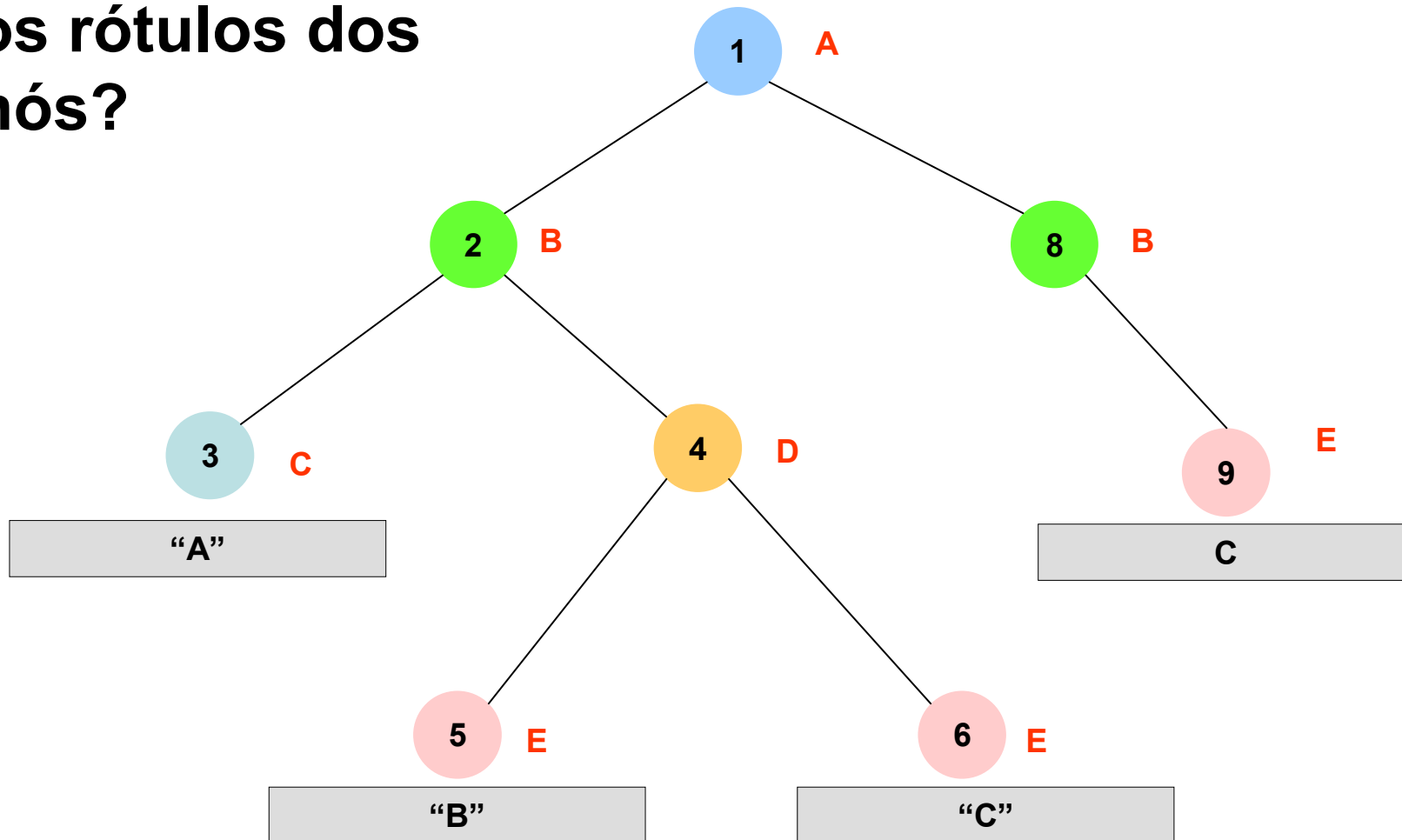
Consultar por A, C

Outro resultado possível:



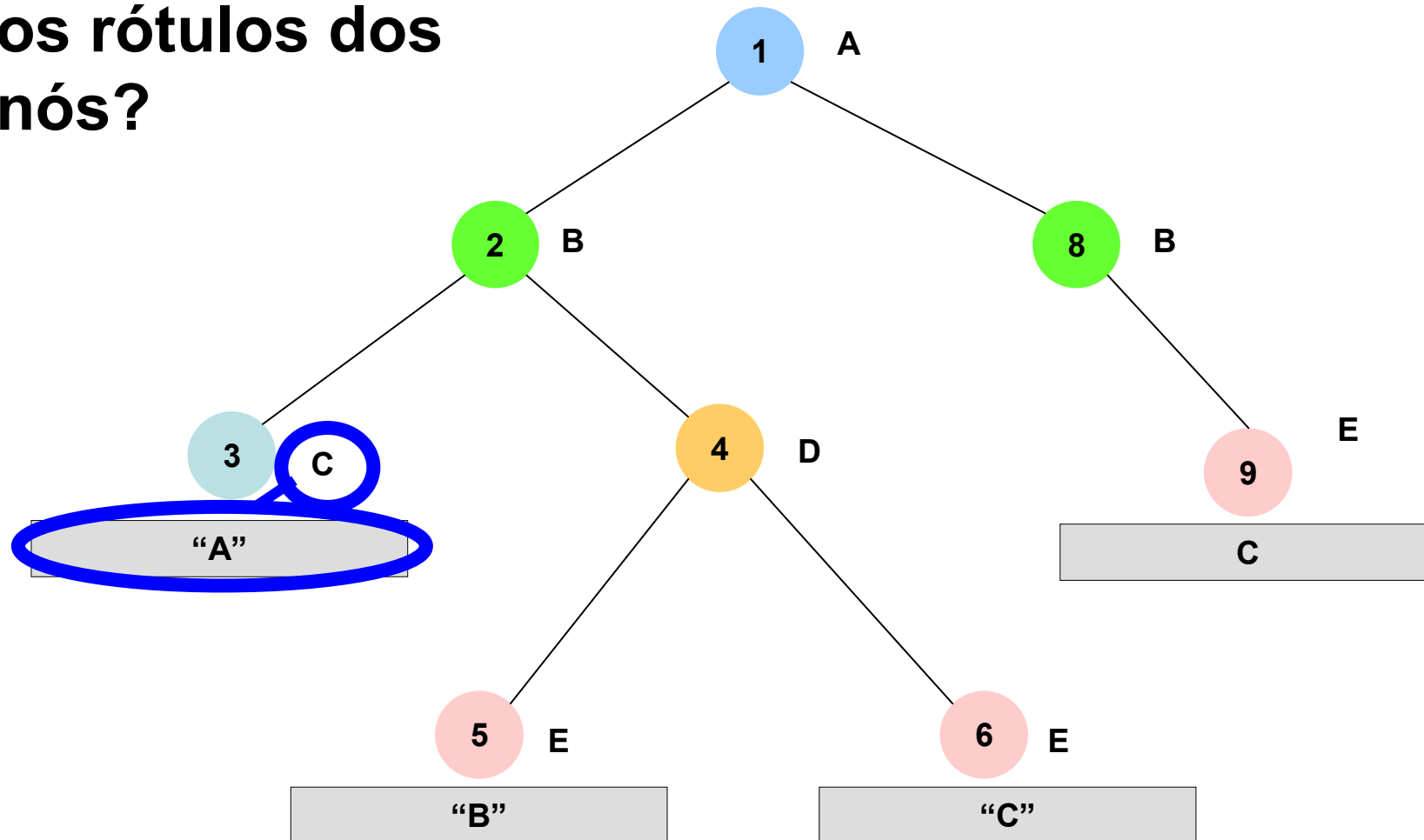
Consultar por A, C

**E se consideramos
os rótulos dos
nós?**



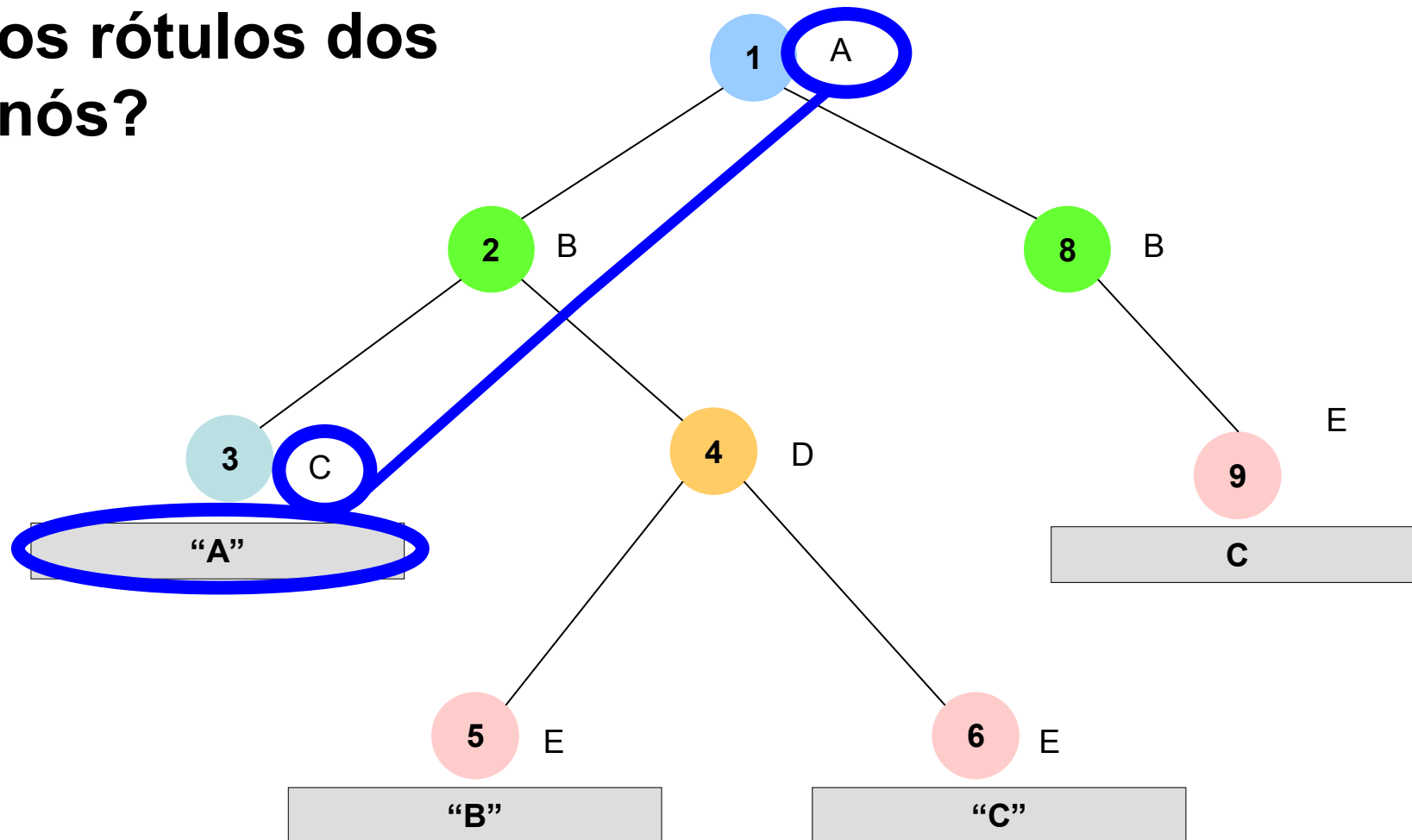
Consultar por A, C

**E se consideramos
os rótulos dos
nós?**



Consultar por A, C

**E se consideramos
os rótulos dos
nós?**



Consultar por A, C

3. Abordagens discutidas:

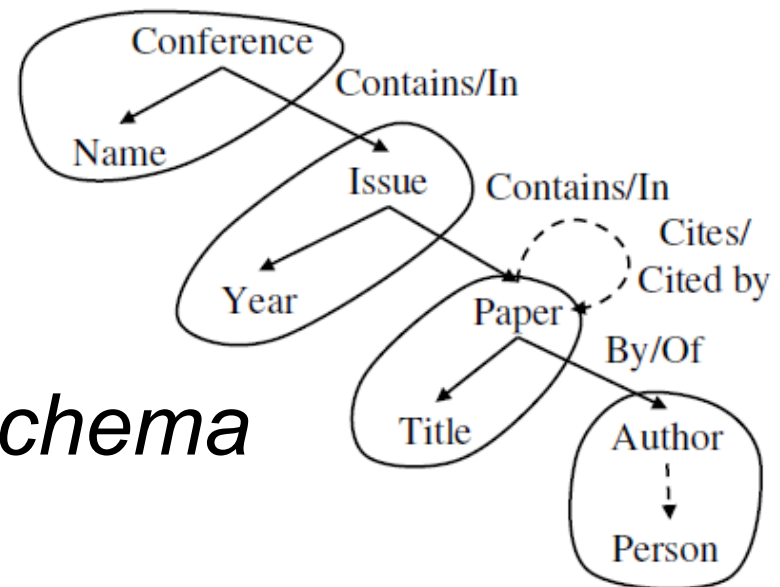
3.1. Balmin e outros ^[1]

3.1. *Balmin* e outros

- Os resultados escolhidos são árvores mínimas de fragmentos XML que contêm todas as palavras-chave (conjuntiva).
- Árvores de menor tamanho denotam maior associação entre as palavras e são consideradas mais relevantes no *rank*.

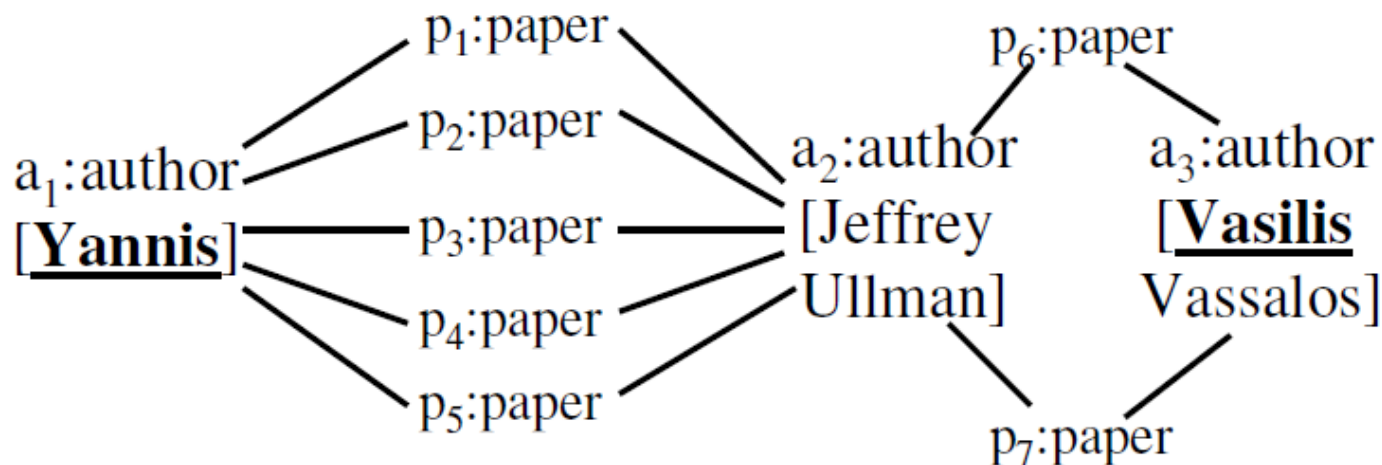
3.1. Balmin e outros

- Busca tirar vantagem do conhecimento do esquema para facilitar a apresentação de resultados.
- Requer que o DBA divida o esquema em “nós de resposta” chamados de *Target Schema Segments* (TSS).



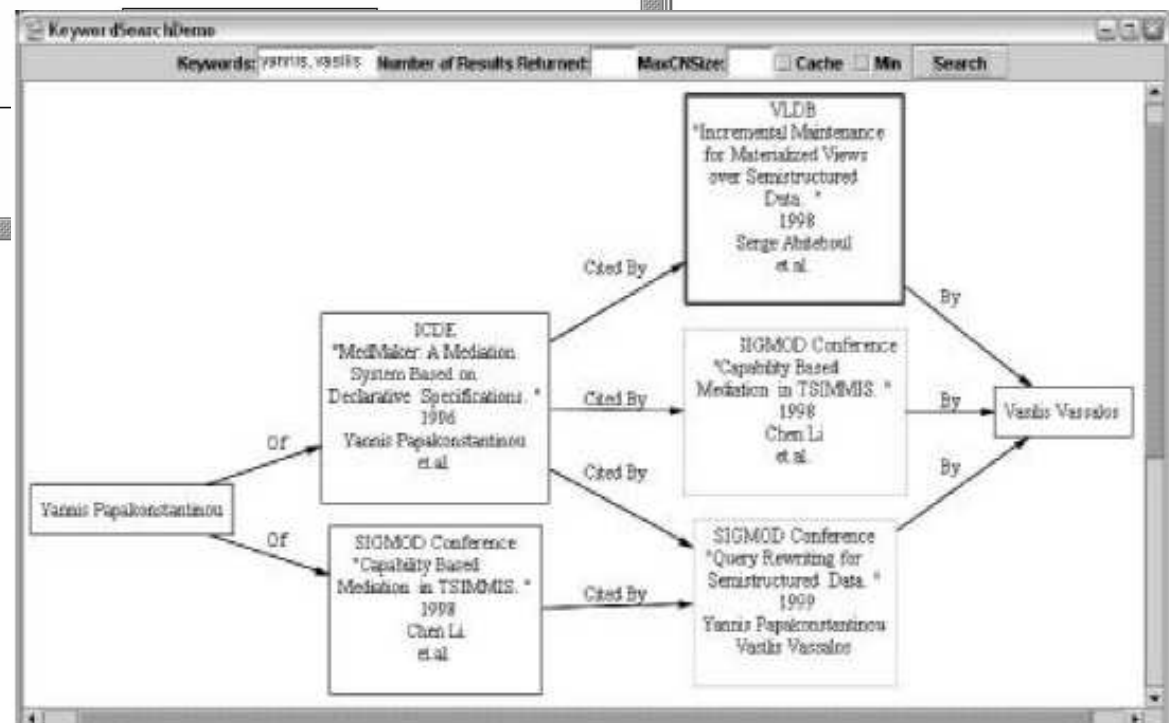
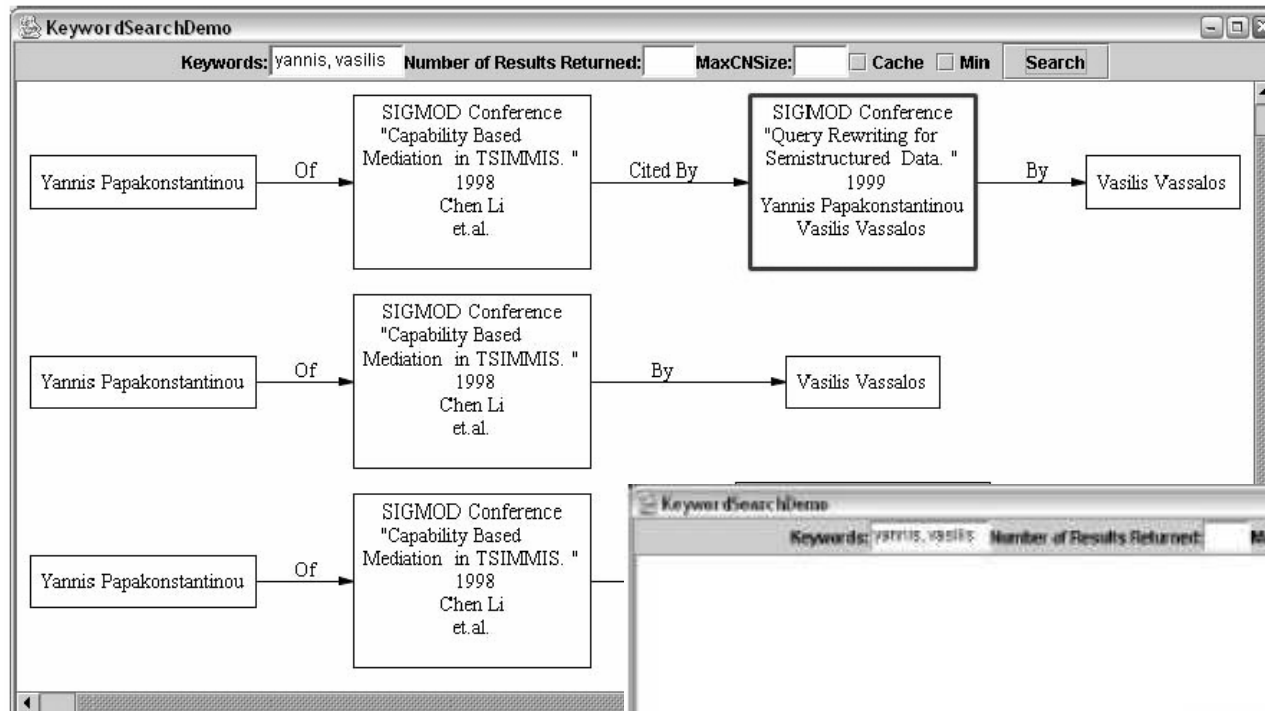
3.1. Balmin e outros

- O algoritmo evita o problema da apresentação de resultados duplicados:
 - Consultas que produzem um grande número de árvores com fragmentos de informações repetidos.



3.1. *Balmin* e outros

- Exibe os resultados através de um grafo de apresentação.
- Permite que o usuário expanda e navegue pelo grafo:
 - Emprega uma técnica de indexação no grafo XML para permitir a navegação rápida e a busca por conexões entre elementos que contém respostas.



3.1. *Balmin* e outros

- Um índice invertido é criado durante a fase de pré-processamento, contendo cada palavra-chave e sua respectiva lista de elementos.
- Conexões entre elementos (IDREFS) são armazenadas para permitir a navegação rápida no grafo.

3.1. *Balmin* e outros

- Recupera do índice os elementos com alguma das palavras-chave e gera árvores mínimas contendo todas as palavras chave.
- Apresenta apenas os resultados do topo e de forma compactada. A expansão é feita sobre demanda, realizando novas consultas mais focadas.

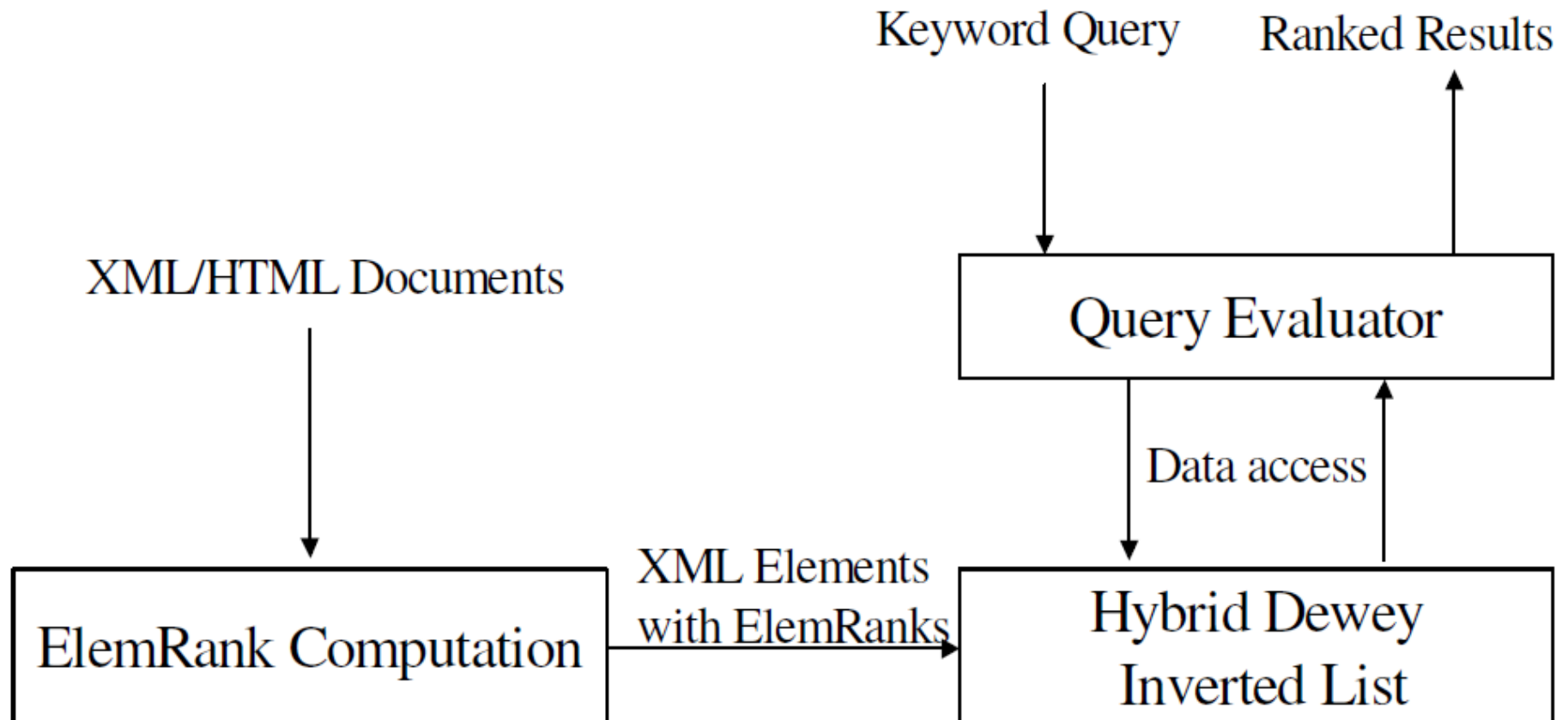
3. Abordagens discutidas:

3.2. XRANK [2]

3.2. XRANK

- Abordagem conjuntiva.
- Retorna os resultados mais específicos que contém todas as palavras chaves.
- Permite tanto que o usuário navegue pelos ancestrais da resposta, quanto a utilização dos “nós de reposta” se disponíveis.

3.2. XRANK



3.2. XRANK

- Considera que a função de *ranking* deve cobrir 3 propriedades:
 - **Especificidade de resultado:** resultados mais específicos devem prevalecer aos mais genéricos. (exclusiva ao XML)
 - **Proximidade de palavras-chave:** deve-se considerar distância entre termos.
 - **Conhecimento das conexões:** elementos muito referenciados devem ser bem posicionados.

3.2. XRANK

- O ***rank*** de um elemento para uma palavra-chave k equivale ao rank do seu descendente que contém k , multiplicado por uma constante de decaimento proporcional a distância entre o elemento e o descendente.
- Elementos que estão próximos da palavra-chave em altura são pouco decaídos (garante especificidade).
- Permite que múltiplas ocorrências da palavra-chave no elemento possam influenciar no *rank*.

3.2. XRANK

- O ***rank* geral** de um elemento, considerando todas as palavras-chave, é dado pela soma do *rank* do elemento para cada palavra-chave, multiplicado por uma medida de proximidade entre elas.
- Permite a utilização de quaisquer funções de *rank* e constantes que se adequem à proposta.
- Permite dar peso às palavras-chave.

3.2. XRANK

- Mecanismo de *ranking* é um refinamento do *PageRank* do *Google*, funcionando assim para HTML e para XML:
 - Se o XML possui altura 2, a consulta ocorre como se fosse em documentos HTML.
 - Permite assim uma passagem suave da *Web* atual para a *Web* semântica.

3.2. XRANK

- Trata conexões do tipo *idref* diferentemente de conexões entre pai e filho.
- Elementos que contém vários filhos com *rank* alto prevalecem sobre elementos que possuem poucos filhos com *rank* alto.

3.2. XRANK

- Evolui a estratégia de indexação:

Abordagem Nãive



Lista Invertida de *Dewey*



Lista Invertida de *Dewey Rankeada*



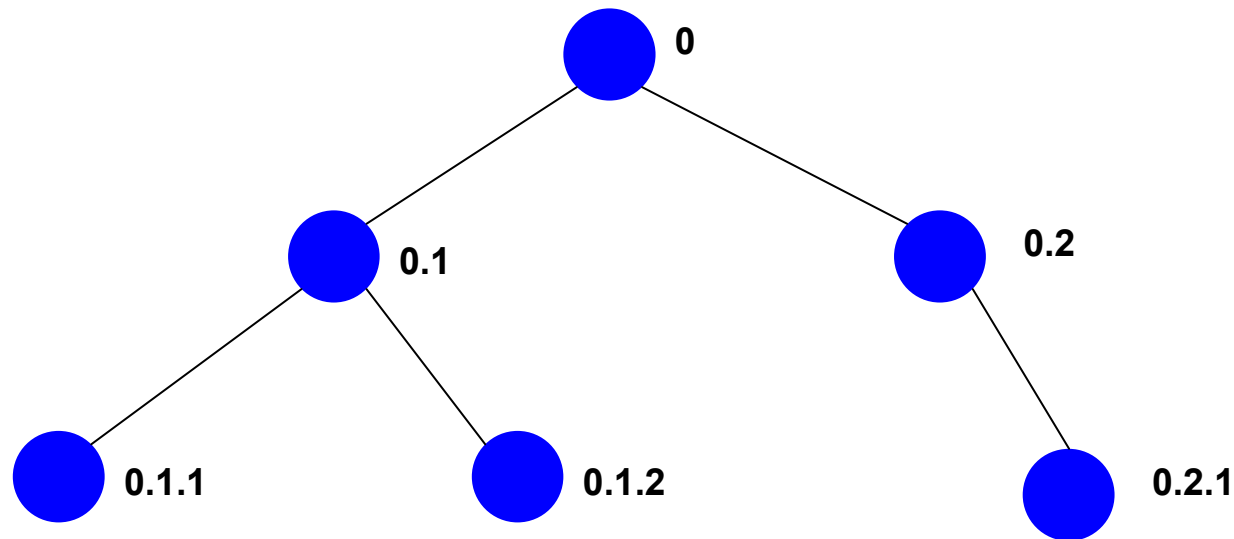
Lista Invertida de *Dewey Híbrida*

3.2. XRANK

- **Abordagem Näive:**
 - Índice invertido onde cada palavra-chave possui uma lista de elementos que a contém;
 - Gera um *overhead* de espaço: se uma palavra-chave ocorre em um elemento, seus ancestrais serão apontados também;
 - Esse *overhead* pode gerar resultados confusos;
 - O índice por si só não garante a especificidade de resultado.

3.2. XRANK

- **Lista Invertida de *Dewey* (DIL):**
 - O ID do ancestral é um prefixo do ID dos descendentes.



3.2. XRANK

- **Lista Invertida de *Dewey* (DIL):**
 - Relações “pai e filho” são implicitamente capturadas;
 - Só armazena os elementos que contém diretamente a palavra-chave (menos espaço);
 - Fica fácil computar o Maior Prefixo Comum (LCP) entre duas palavras-chave.
 - O ID do ancestral mais profundo estará no prefixo do LCP.

3.2. XRANK

- **Lista Invertida de *Dewey Rankeada* (*RDIL*):**
 - Se a lista de Dewey comum for muito longa, é custoso processá-la por completo, sabendo-se que serão exibidos apenas os primeiros resultados;
 - Solução:
 - ordenar a lista pelo *rank* do elemento, ao invés de ordená-la pelo índice do elemento.

3.2. XRANK

- **Lista Invertida de *Dewey Rankeada* (*RDIL*):**
 - Em cada lista de elementos do índice invertido há uma *B+Tree* que é utilizada para determinar o LCP da palavra-chave com as demais;
 - Um algoritmo de limite (TA) é utilizado, e quando atinge-se *m* resultados, o algoritmo para de buscar nas listas.

3.2. XRANK

- **Lista Invertida de *Dewey Rankeada* (*RDIL*):**
 - Não funciona muito bem para palavras-chave pouco correlatas, pois assim o algoritmo terá que percorrer toda a lista invertida;
 - Nesse caso o DIL realiza a mesma tarefa, com menos custo.

3.2. XRANK

- **Lista Invertida de *Dewey* Híbrida (*HDIL*):**
 - Solução proposta pelo XRANK;
 1. Executa o RDIL monitorando sua *performance*.
 2. Calcula o tempo estimado para concluir o RDIL.
 3. Se o tempo for maior que o tempo do DIL, troca de abordagem.
 - Tempo do DIL pode ser calculado baseado apenas número de palavras-chave e no tamanho das listas do índice invertido.

3.2. XRANK

- **Atualização das listas invertidas:**
 - Não dá suporte para inserção/remoção de novos elementos;
 - Dá suporte para exclusão de documentos:
 - Por utilizar ID Dewey, basta excluir os elementos com prefixo do documento excluído.

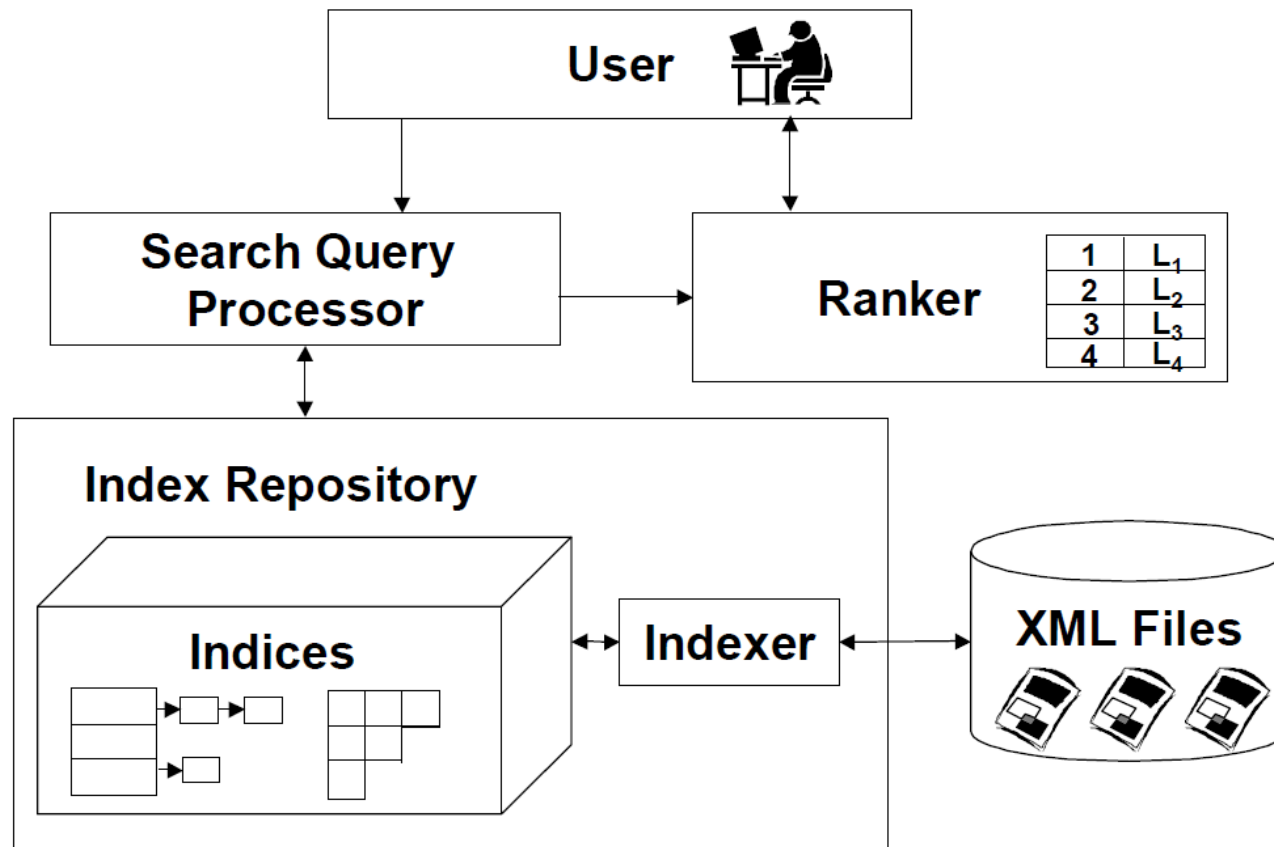
3. Abordagens discutidas:

3.3. XSearch [3]

3.3. XSearch

- Possibilita especificar relação entre TAGs e palavras-chave;
- Só retorna resultados significativamente relacionados;
- *Ranking* combina fatores diversos;
- *Utiliza mais de uma estrutura de índice;*
- É extensível, atuando como um *framework*.

3.3. XSearch

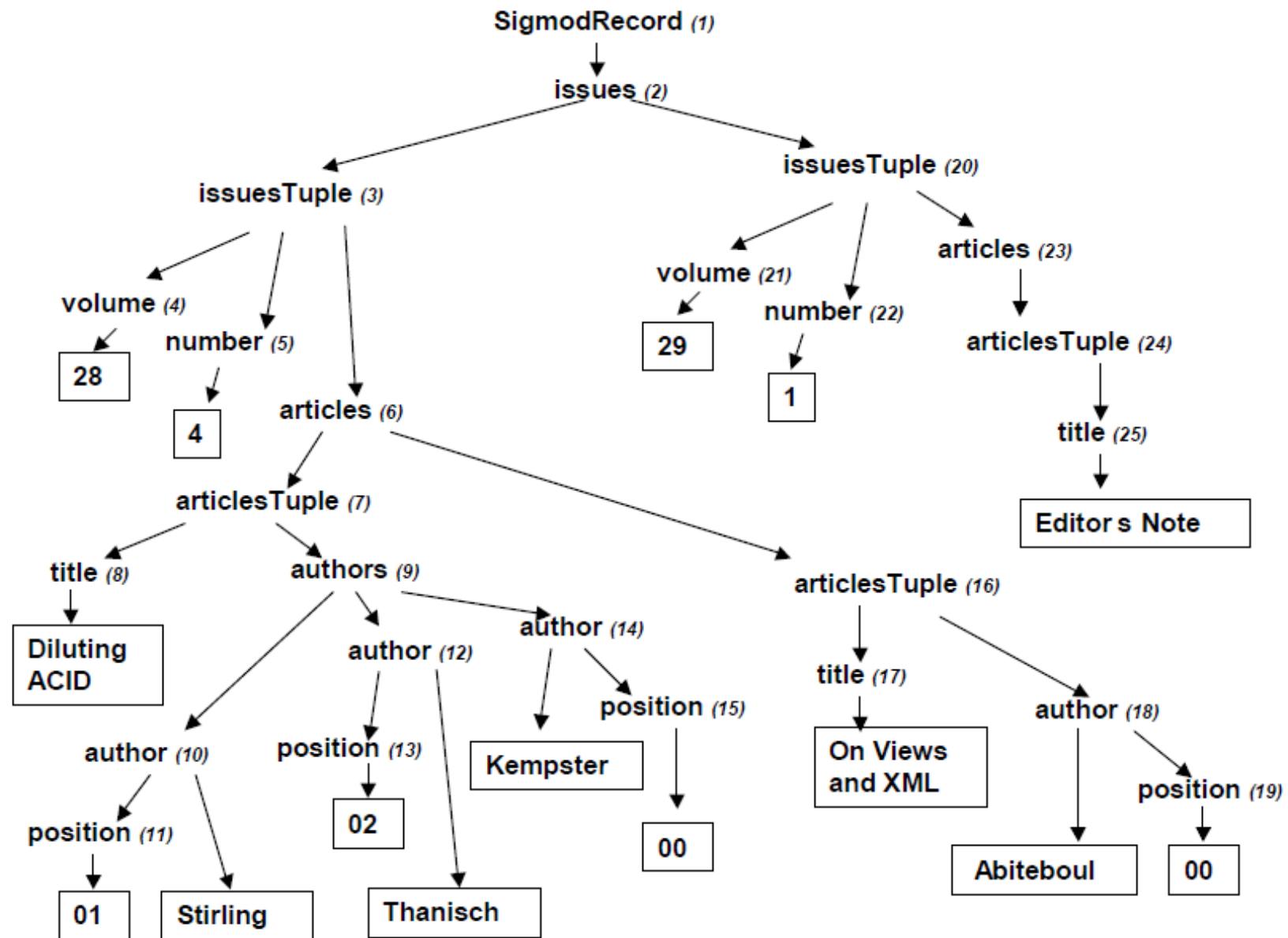


3.3. XSearch

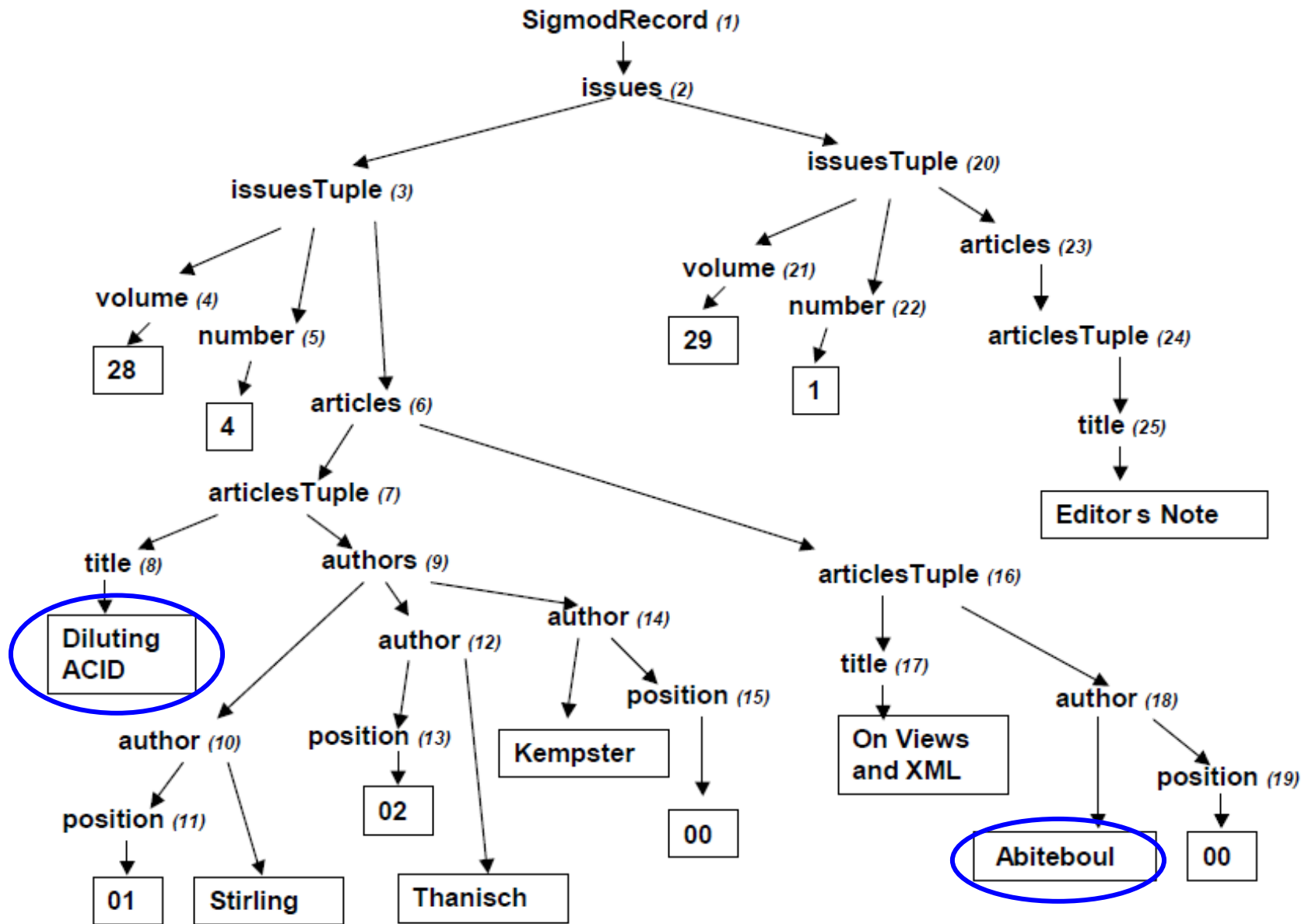
- Formato da consulta é uma lista de rótulos e palavras chave, da forma:
l:k ou l: ou :k
- Pode-se especificar termos opcionais ou requeridos usando o **+** (conjuntivo e disjuntivo);
- Exemplos:
titulo:database, volume:14
+nome:Rafael, cidade:
+author:Norvig, +author:Russel, +book:

3.3. XSearch

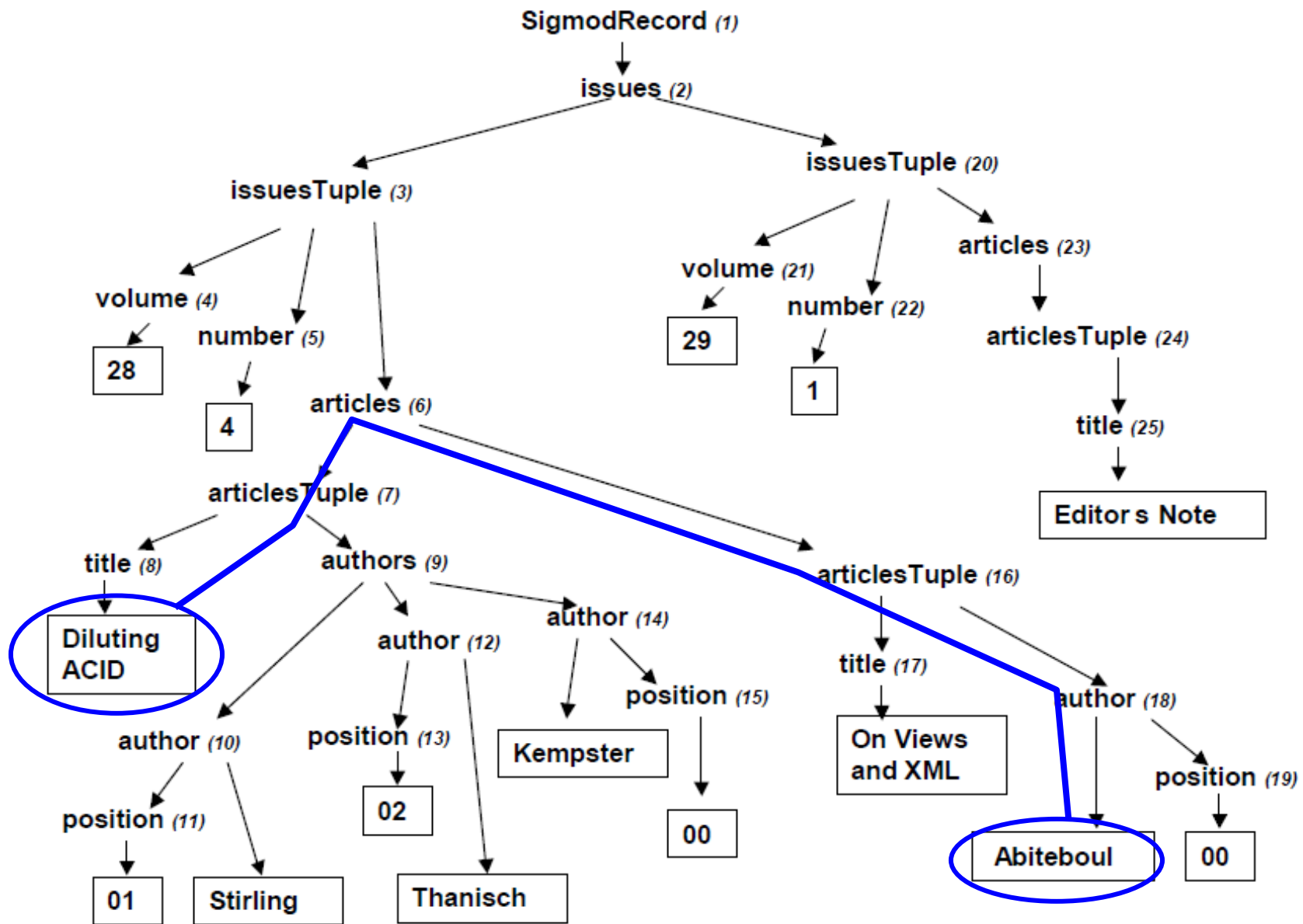
- Considera que os elementos de um resultado devem ser significativamente relacionados:
 - Os elementos da resposta não podem ter ancestrais distintos de mesmo rótulo;
 - O mecanismo não generaliza o “fragmento” de resposta indefinidamente até abranger todas as palavras-chave.
 - Propriedade chamada de **interconexão**.



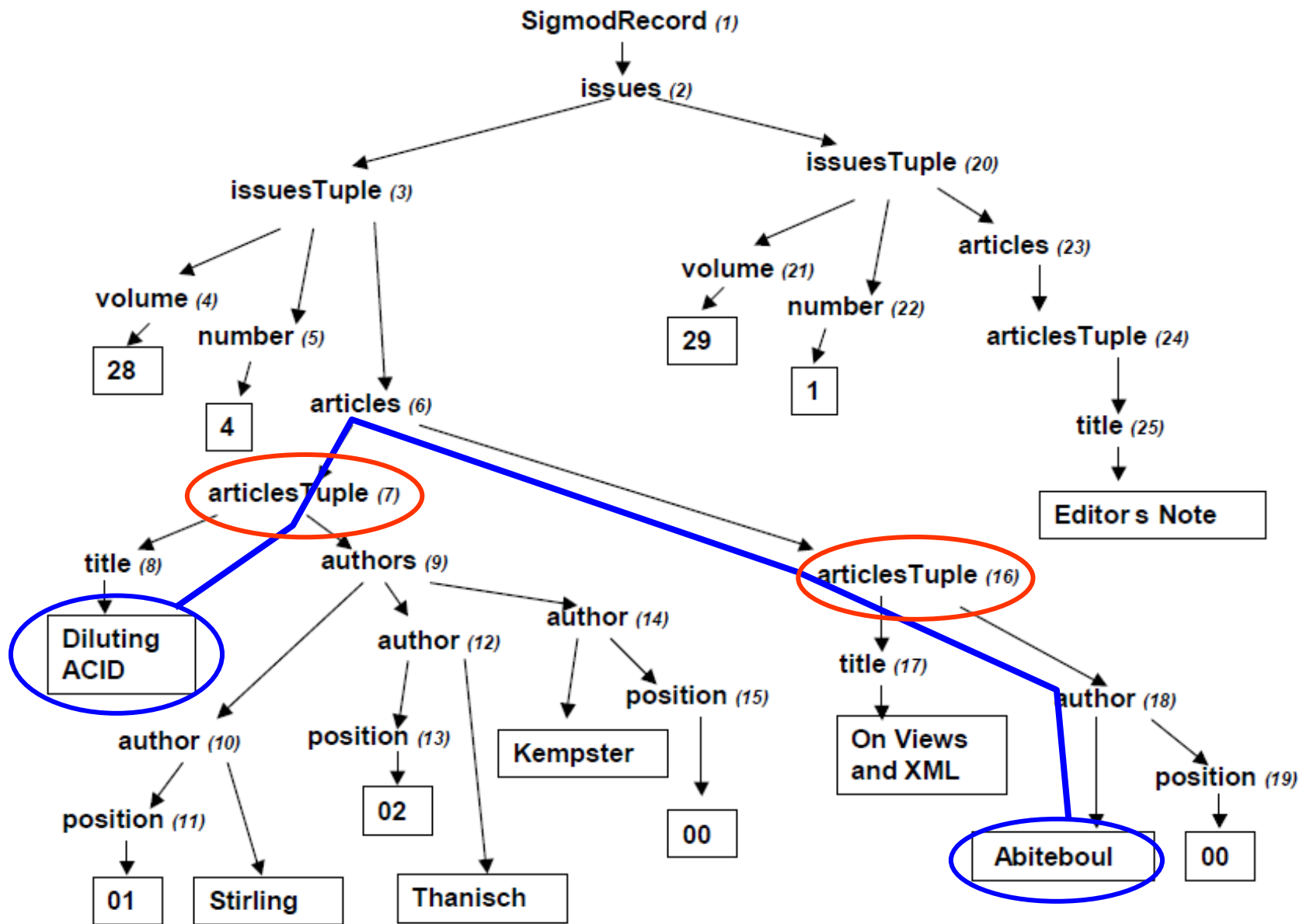
+:Abiteboul, +title:ACID



+:Abiteboul, +title:ACID



+ :Abiteboul, + :title:ACID



+:Abiteboul, +title:ACID

3.3. XSearch

- Mecanismo de *ranking* mais sofisticado;
- Utiliza uma variação do tf-idf (frequência do termo, frequência inversa do documento) para dar peso às palavras chave.
- tf-ilf: (frequência do termo, frequência inversa nas folhas).
 - Considera em quantas folhas o termo também ocorre (relevância).

3.3. XSearch

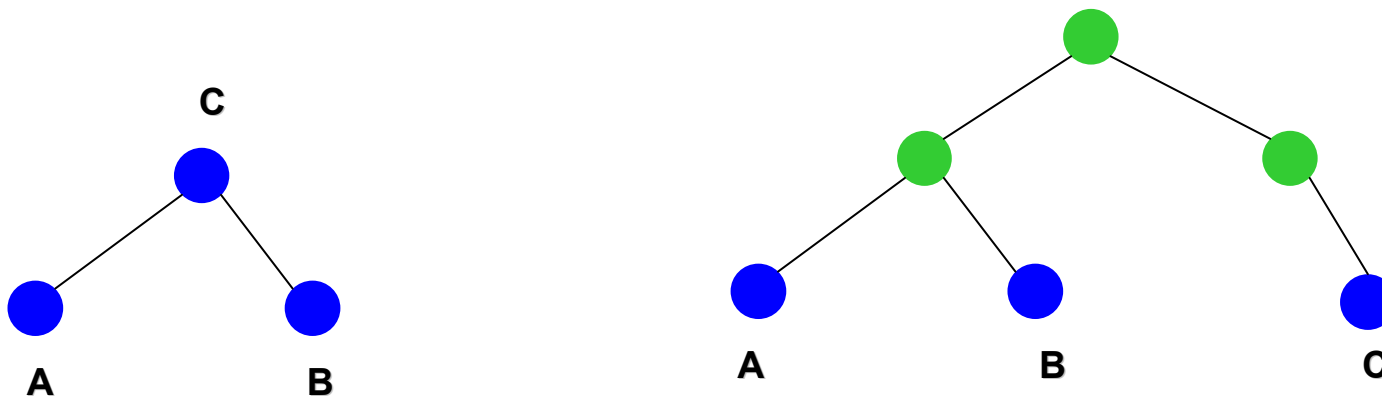
- Os pesos dos termos são armazenados em um índice;
- Permite também que sejam gerados pesos para os rótulos;

3.3. XSearch

- Utiliza o modelo de espaço vetorial para dizer o quanto uma resposta satisfaz a consulta:
 - Mede a similaridade $sim(Q, N)$ entre uma consulta Q e uma resposta N através da distância entre os vetores que compõem Q e N .

3.3. XSearch

- Considera também o tamanho da árvore da resposta $tsize(N)$ (quantidade de nós).
 - Árvores menores denotam respostas mais significativas;



3.3. XSearch

- Considera também a quantidade de elementos que participam em uma relação do tipo ancestral-descendente *anc-des(N)*
 - *Alega que tal conexão representa uma forte conectividade entre dois elementos.*

3.3. XSearch

- O valor do *rank* final para um resultado é uma combinação dos fatores:
 - $\text{sim}(Q, N)$
 - $\text{tsize}(N)$
 - $\text{anc-des}(N)$
- Tais fatores tornam possível avaliar tanto a estrutura quanto o conteúdo dos resultados.

3.3. XSearch

- Utiliza uma repositório de índices:
 - Índice invertido de palavra-chave;
 - Índice invertido de rótulo;
 - Índice de interconexão; *
 - Pode ser criado *offline* por programação dinâmica;
 - Pode ser criado *online*, *checando interconexão apenas dos elementos da consulta*;
- Índice de caminho: *
- Permite encontrar facilmente o LCP de um par de elementos;

4. Extensão do tema

4. Extensão do tema

- *Liu e Chen 2008:*
 - Avalia abordagens de busca existentes quanto à **monotonicidade** e **consistência** de dados e consulta;
 - Propõem um mecanismo de busca por palavras-chave que considera ambas as propriedades e apresenta melhorias quanto à identificação de **resultados relevantes**;

4. Extensão do tema

- *Liu e Chen 2008:*
 - **Monotonicidade:** define a alteração desejada no número de resultados ao atualizar os dados ou as consultas;
 - **Dados:** adição de novos elementos indicam novos candidatos à resposta, logo, um crescimento na quantidade de resultados;
 - **Consulta:** adição de novos termos indica uma maior restrição na busca, logo, um decaimento na quantidade de resultados;

4. Extensão do tema

- *Liu e Chen 2008:*
 - **Consistência:** define como a quantidade de resultados deve se alterar frente a uma atualização nos dados ou consultas;
 - **Dados:** ao adicionar um novos elemento, se surgirem novas sub-árvores de resposta, elas irão conter tal elemento;
 - **Consulta:** ao adicionar um novo termo, se surgirem novas sub-árvores de resposta, elas irão conter ao menos uma ocorrência de tal termo;

4. Extensão do tema

- *Liu e Chen 2008*:
 - Propõe mecanismo *MaxMatch*, parte do *XSeek*;
 - Abordagem SLCA
 - Privilegia sub-árvores de resultado que não contém outros resultados;
 - Avalia relevância de resultados pela quantidade e informação que ele contém relativo aos seus *siblings*;

4. Extensão do tema

- *Kong e Outros 2009:*
 - Implementa o mecanismo *ValidMatch*
 - Idéia do **contribuinte válido**;
 - Tenta resolver um problema do *MaxMatch*;
 - Não prejudica elementos com pouca informação, se essa informação for única;
 - Mantém propriedades (monot. e consist.)
 - Mantém *performance* compatível com o *MaxMatch*.

4. Extensão do tema

- *Bao e Outros 2009*:
 - Estabelece que existem nós que são procurados (*search for*), e nós que são utilizados na procura (*search via*);
 - Identifica com isso problemas de de intenção de busca e análise de relevância no *ranking*;
 - Aponta o problema da ambiguidade de palavras-chave.

4. Extensão do tema

- *Bao e Outros 2009:*
 - Define a XML-TF e XML-DF (análogo ao tf-idf);
 - Difere-se do TF-IDF apresentado na abordagem XSearch;
 - XML-DF Considera quantidade de ocorrências do termo na sub-árvore de um nó, e não em uma folha apenas.

4. Extensão do tema

- *Bao e Outros 2009*:
 - Através do XML-TF*IDF, computa o nível de confiança de nós candidatos à *search for* e *search via*;
 - Utiliza esse conceito para gerar um *rank* mais próximo da intenção do usuário;
 - Implementa tais técnica num sistema chamado *XReal*.

5. Conclusões

5. Conclusões

- É um tema novo:
 - Poucas publicações;
 - Artigos relevantes à partir de 2003;
- Literatura já apresenta mecanismos de consulta interessantes e extensíveis;
- Uma das preocupações atuais é tornar as abordagens mais sensíveis quanto à relevância e validade dos resultados;

6. Referências

- [1] **Balmin, A.; Hristidis, V.; Koudas, N.; Papakonstantinou, Y.; Srivastava, D.; Wang, T.** *A System for Keyword Proximity Search on XML Databases*. Proceedings of the International Conference on Very Large Data Bases, 2003, pp. 1069-1072.
- [2] **Cohen, S.; Mamou, J.; Kanza, Y.; Sagiv, Y.** *XSEarch: A Semantic Search Engine for XML*. Proceedings of the International Conference on Very Large Data Bases, 2003, pp. 45-56.
- [3] **Guo, L.; Shao, F.; Botev, C.; Shanmugasundaram, J.** *XRANK: ranked keyword search over XML documents*. Proceedings of the ACM SIGMOD International Conference on Management of Data, 2003, pp. 16–27.
- [4] **Liu, Z. and Cher, Y.** *Reasoning and identifying relevant matches for XML keyword search*. Proceedings of the International Conference on Very Large Data Bases, 2008, pp. 921-932.
- [5] **Kong, L., Gilleron, R., and Lemay, A.** *Retrieving valid matches for XML keyword search*. Proceedings of the ACM Symposium on Applied Computing, 2009, pp. 1536-1540.
- [6] **Zhifeng Bao; Tok Wang Ling; Bo Chen; Jiaheng Lu.** *Effective XML Keyword Search with Relevance Oriented Ranking*. International Conference on Data Engineering, 2009, pp.517-528.

Fim.

Perguntas?