Construção e utilização de um protótipo para o processamento da linguagem de interrogação IXDIRQL

Alda Lopes Gançarski[1] and Pedro Rangel Henriques[2]

[1] Laboratoire d'Informatique de Paris 6, Paris, France
email : Alda.Lopes@lip6.fr
[2] Universidade do Minho, Braga, Portugal
email: prh@di.uminho.pt

Abstract. O XPath é uma linguagem de interrogação para acesso a componentes de documentos XML usando restrições estruturais e sobre o conteúdo. A linguagem de interrogação IXDIRQL estende o XPath com *operações de similaridade textual* típicas da Recuperação de Informação e com *operações de selecção* que permitem uma construção interactiva das perguntas manipulando os resultados intermédios. O protótipo, criado para o processamento da linguagem IXDIRQL, foi posto à disposição de utilizadores reais para realizar inúmeros testes. Por um lado, quis-se verificar o seu correcto funcionamento; por outro lado, pretendeu-se experimentar a facilidade de compreensão da própria linguagem IXDIRQL, analisando a capacidade dos utilizadores para formularem as perguntas correctas face a determinados pedidos pré-definidos. Neste artigo vamos apresentar alguns aspectos que julgamos particularmente interessantes sobre a construção do protótipo e sua utilização.

1 Introdução

A **recuperação de informação (RI)** [1] (*Information Retrieval* em inglês) consiste na pesquisa e entrega ao utilizador dos documentos que se julgam relevantes para satisfazer o seu pedido formulado através de uma pergunta (ou interrogação, do inglês *query*). Uma pergunta consiste numa expressão em linguagem natural que descreve o assunto procurado. Usualmente, a relevância dum documento é obtida através duma função de similaridade entre o documento (uma sua síntese, ou caracterização) e a pergunta. Os documentos são entregues ao utilizador ordenados por ordem decrescente da sua relevância.

Para tirar partido da informação estrutural dos documentos XML, o formato das perguntas foi enriquecido de modo a aceder-se a partes específicas dos documentos. Uma pergunta consiste, agora, numa sequência de restrições estruturais (caminhos, filtros, grupos) sobre o conteúdo (comparações com valores textuais ou numéricos). O XQuery é proposto pelo *W3C Consortium* como a norma de interrogação para XML [3]. O XQuery baseia-se em diferentes linguagens existentes, entre as quais o XPath [2] e o XML-QL [5]. Estas linguagens são

consideradas como de *recuperação de dados* (em inglês, *data retrieval*) porque a resposta à pergunta é um conjunto com todos os elementos que *satisfazem* literalmente a respectiva pergunta, indistintamente. Não há, portanto, nestes sistemas a noção de relevância, ou seja não há qualquer indicação sobre as melhores ou piores respostas. Alguns trabalhos [7] [4] propuseram extensões a estas linguagens para incluir restrições de similaridade textual cujo resultado é uma lista de elementos ordenados pela sua relevância. Recentemente, o sistema de RI associado à linguagem IXDIRQL [8, 9], baseada no XPath, propõe, além da similaridade textual, um paradigma interactivo na construção das perguntas. Esta abordagem permite a manipulação dos resultados intermédios, i.e. os resultados obtidos numa fase podem ser tomados em consideração (é possível seleccionar o sub-conjunto de elementos interessantes) para refinar a pergunta seguinte.

Este artigo apresenta alguns aspectos que nos pareceram particularmente relevantes sobre a construção e utilização de um protótipo para o processamento da linguagem IXDIRQL. Na secção 2, a linguagem é introduzida. Depois, a secção 3 mostra como são calculadas as relevâncias dos elementos perante uma operação de similaridade textual e como são propagadas ao longo das perguntas. A secção 4 introduz o conceito de índice e a sua relevância para optimizar a operação de procura. O editor IXDIRQL associado a um processador incremental é o tema da secção 5. A secção 6 aborda uma questão fundamental para o sucesso desta abordagem: a apresentação dos resultados. O protótipo construído foi posto à disposição de utilizadores para testar a sua funcionalidade e a facilidade com que as operações de selecção são usadas em determinados pedidos de informação. Esta experiência é descrita na secção 7. O artigo termina com uma breve conclusão, dando directrizes para trabalho futuro.

2 A linguagem IXDIRQL

Como se explica ao pormenor em [9], a linguagem IXDIRQL estende o XPath com os seguintes operadores de similaridade textual sobre elementos:

ROSearch Procura elementos de qualquer tipo onde os termos indicados ocorrem. Por exemplo, $ROSearch\ 'XML'$ procura elementos de qualquer tipo sobre *'XML'*.

ROSearchTE Semelhante ao anterior, procura elementos de qualquer tipo onde os termos indicados ocorrem, mas agora apresenta-os ordenados em listas diferentes, uma para cada tipo de elemento. Por exemplo, $ROSearchTE\ 'XML'$ procura elementos de qualquer tipo onde o termo *'XML'* ocorre, sendo o resultado uma lista ordenada de artigos, outra de secções, outra de referências, etc.

Sim Procura elementos de um tipo especificado onde o termo indicado ocorre. Este operador é integrado nas perguntas estruturadas do XPath. Por exemplo, $/artigo/titulo\ Sim\ 'XML'$ procura elementos do tipo *titulo* que contenham no seu interior o termo *'XML'*. O resultado é uma lista de títulos ordenados por ordem decrescente de relevância.

Este artigo inside apenas na operação *Sim*.

Os elementos resultantes destes três operadores são associados a uma relevância cujo valor está no intervalo [0, 1]: 0 para elementos não relevantes, 1 para elementos totalmente relevantes e valores intermédios para diferentes níveis de similaridade textual entre o conteúdo do elemento específico e a expressão em linguagem natural pedida. As operações importadas do XPath conduzem sempre a uma relevância 1 porque os elementos encontrados satisfazem completamente as restrições estruturais e sobre o conteúdo. As operações lógicas importadas do XPath retornam agora, não valores booleanos (verdadeiro e falso), mas *valores de verdade* representados pela relevância R associada, o que significa *'verdadeiro com probabilidade R'*. Assim, verdadeiro é o valor de verdade de relevância 1 e falso é o valor de verdade de relevância 0. Valores de verdade com relevâncias entre 0 e 1 podem ocorrer em expressões lógicas como

$$titulo\ Sim\ 'XML'\ \$and\$\ autor =' Silva'$$

Neste caso, o resultado é uma série de valores de verdade calculados tendo em conta a relevância dos títulos.

Face aos resultados intermédios das perguntas, o sistema IXDIRQL permite, graças ao seu carácter interactivo e incremental, três operadores de selecção:

Select Selecciona entre os elementos apresentados aqueles que o utilizador achou interessantes; a selecção é feita indicando a respectiva chave (o identificador único de cada elemento). Por exemplo

$$artigo/titulo\ Select\ tit\_4, tit\_8$$

selecciona os títulos tit_4, tit_8 do conjunto de títulos encontrados.

SelectN n Selecciona o conjunto dos primeiros n elementos. Por exemplo, a pergunta

$$/artigo[autor =' Silva']\ SelectN\ 10$$

conduz aos 10 primeiros artigos do autor *'Silva'* encontrados na colecção.

JudgeRel Restrinje a lista de documentos resultante de uma operação de similaridade textual (*Sim*) previamente realizada, por selecção explícita dos elementos que pertencem aos documentos que aparentam ser mais interessantes para as necessidades do utilizador. Por exemplo,

$$artigo[titulo\ Sim\ 'XML'\ JudgeRel\ tit\_4, tit\_8]/referencia$$

é uma pergunta que permite obter a lista de de referências bibliográficas citadas nos dois artigos cujos títulos (contendo *'XML'*) parecem ser os mais relevantes. Primeiro, obtem-se, com o operador *Sim*, uma lista de títulos (sobre *'XML'*) ordenada pela relevância. Depois, através do operador *JudgeRel*, o utilizador escolhe o conjunto de títulos dessa lista que ele considera importantes. Por fim, obtem-se a lista de referências bibliográficas citadas nos artigos correspondentes aos títulos relevantes.

3 O cálculo das relevâncias

Para incluir a similaridade textual numa linguagem de interrogação para XML houve necessidade de adaptar as fórmulas de cálculo da RI tradicional para o formato estruturado dos documentos.

3.1 Na recuperação de informação tradicional

Em RI tradicional [1], um termo com uma frequência grande num documento é seguramente um bom representante desse documento. Sejam: $\{d_1, ..., d_N\}$ um conjunto de documentos; $\{t_1, ..., t_T\}$ um conjunto de termos; n_i o número de documentos onde aparece o termo t_i (i=1,...,T); $freq_{ij}$ a frequência do termo t_i no documento d_j (j=1,...,N). A frequência tf_{ij} **normalizada** do termo t_i no documento d_j é calculada por:

$$tf_{ij} = \frac{freq_{ij}}{\max_{l=1,\ldots,T} freq_{lj}}$$

Um termo que aparece numa pequena fracção da colecção de documentos tem um bom **poder discriminante** dessa fracção em relação à colecção. O poder discriminante do termo t_i é calculado por:

$$idf_i = \log N/n_i$$

A multiplicação das duas medidas (*tf*idf*) é usada frequentemente para a representação dos termos e é referida como *tf.idf*. Os documentos e as perguntas em linguagem natural são muitas vezes representados por vectores cujos componentes são as medidas tf.idf dos termos relativamente ao documento (ou pergunta) em causa. Neste modelo vectorial, a relevância de um documento em relação a uma pergunta é o resultado de uma função de correlação entre os vectores correspondentes. Uma função simples, eficiente e, portanto, normalmente utilizada é o cosseno. A similaridade textual entre o documento d_j (j=1..N) e e pergunta q é, então, estimada pela expressão seguinte:

$$sim(d_j, q) = \frac{d_j \times q}{|d_j| \times |q|}$$

Seja w_{ij} a medida tf.idf do termo t_i (i=1..T) no vector d_j (j=1..N). A similaridade calcula-se, então, por:

$$sim(d_j, q) = \frac{\sum_{i=1}^{T} w_{ij} \times w_{iq}}{\sqrt[2]{\sum_{i=1}^{T} w_{ij}^2} \times \sqrt[2]{\sum_{i=1}^{T} w_{iq}^2}}$$

3.2 Na recuperação de informação em XML

Tendo em conta que a relevância no IXDIRQL é calculada para os elementos, a fórmula tf.idf foi adaptada para que as frequências sejam calculadas no conjunto dos elementos (e não dos documentos) da colecção. Assim, sejam: Ne o número de elementos da colecção; n_i o número de elementos onde o termo t_i aparece (i=1..T); $freq_{ij}$ a frequência do termo t_i no elemento e_j (j=1..Ne). A representação tf.idf passa a chamar-se *tf.ief* onde:

$$tf_{ij} = \frac{freq_{ij}}{\max_{l=1..T} freq_{lj}} \quad \text{e} \quad ief_i = \log Ne/n_i.$$

Para combinar as relevâncias dos resultados de diferentes operações, são usadas seguintes fórmulas de cálculo probabilístico para a disjunção e a conjunção de eventos [7]:

$$P(p_1 \vee ... \vee p_n) = \sum_{i=1}^{n} (-1)^{i-1} (\sum_{1 \leq j_1 < ... < j_i \leq n} P(p_{j1} \wedge ... \wedge p_{ji}))$$

$$P(p_1 \wedge ... \wedge p_n) = P(p_1) \times ... \times P(p_n), \textit{ assumindo } p_1, ..., p_n \textit{ independentes.}$$

Um evento é o facto de um elemento ser relevante. A relevância é a probabilidade associada a um evento, i.e. a probabilidade de que um elemento seja relevante. Por exemplo, seja a pergunta

$$/artigo\ Sim\ 'XML'/referencia\ Sim\ 'XSL'$$

Há aqui dois predicados de similaridade textual, um associado aos artigos, outro às referências. Suponhamos que o artigo a_1 tem relevância $P(a_1)$ interpretada como a probabilidade de que a_1 seja relevante em relação ao assunto *'XML'*. Suponhamos, também, que a referência bibliográfica r_1 é citada no artigo a_1 e tem relevância $P(r_1)$. $P(r_1)$ é a probabilidade de que r_1 seja relevante em relação ao assunto *'XSL'*. No resultado final, há que combinar as duas relevâncias de forma a calcular a relevância total de cada referência. Isso é feito usando $P(a_1 \wedge r_1)=P(a_1) \times P(r_1)$ que traduz a probabilidade de que o artigo a_1 e a referência r_1 sejam ambos relevantes.

Quando o operador *Sim* é seguido do operador *JudgeRel*, primeiro é calculada a relevância associada ao operador *Sim*; depois, a relevância de cada elemento seleccionado por *JudgeRel* é substituída por 1, sendo 0 a dos restantes.

Quanto aos valores de verdade, eles resultam das operações lógicas do IXDIRQL. Se não houver nenhum predicado de similaridade na pergunta, a relevância associada aos valores de verdade é 1. Caso contrário, a relevância é calculada pelas fórmulas de cálculo probabilístico apresentadas para os elementos, assumindo agora um evento como o facto do valor de verdade ser o valor lógico verdadeiro.

4 Os índices de texto e de estrutura

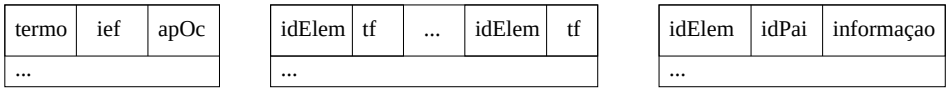
Em RI, a informação sobre os termos da colecção é guardada em *índices* para um rápido processamento das perguntas. Os termos são extraídos a partir de

operações efectuadas sobre certas palavras encontradas nos documentos. As palavras seleccionadas excluem as proposições, os determinantes, os pronomes e outras consideradas de *significado 'vazio'* (*sem interesse*) para a RI. As palavras seleccionadas são, depois, reduzidas à sua raíz etimológica. Estas operações reduzem o tamanho dos índices a cerca de 40%.

Em relação aos documentos textuais, os índices mais usados são os *inverted files*. Uma *inverted file* é composta por dois ficheiros: o *vocabulário*, que guarda a informação sobre os termos; e o das *ocorrências* (*postings* em inglês), que assinala as posições de cada termo nos documentos.

No protótipo do IXDIRQL, a representação vectorial *tf.ief* é guardada, conforme manda a tradição, nas tabelas `vocabulário` e respectivas `ocorrências`. Como se pode ver na figura 1, o vocabulário (a) associa a cada termo o seu valor de *ief* e um apontador para a lista de ocorrências respectivas (*apOc*). Esta lista, representada na figura 1 (b) contém, agora, nas suas células os identificadores dos elementos onde o termo aparece (*idElem*) e o respectivo valor de *tf*.

Neste caso criou-se, ainda, um índice de estrutura (figura 1, c) que associa a cada elemento (*idElem*) o seu pai (*idPai*) e informação (*informação*) que inclui o seu tipo, um apontador para o próximo elemento do mesmo tipo e os atributos XML respectivos. Este índice permite o acesso aos elementos existentes na colecção para se escolherem aqueles que satisfazem cada operação, de acordo com o seu tipo e as suas relações hierárquicas. A relevância associada aos elements obtidos numa operação de *Sim* é calculada usando em paralelo os índices de estrutura e de texto (o vocabulário e as ocorrências, onde se faz a correspondência entre os termos e os elementos).



(a) Vocabulario (b) Ocorrencias (c) Estrutura

Fig. 1. Os índices de texto (vocabulário (a) e ocorrências (b)) e de estrutura (c).

5 O editor/processador do IXDIRQL

Para a escrita e o processamento de perguntas IXDIRQL, foi criado automaticamente um editor associado a um processador incremental. Para isso, foi usado o LRC [10], um gerador de ambientes incrementais baseados na definição formal da linguagem em causa. A definição da linguagem é feita através duma gramática de atributos onde a semântica dinâmica corresponde ao cálculo do resultado das perguntas. A meta-linguagem usada para definir essa gramática é a *Synthesizer Specification Language* (SSL) [11]. Para dar uma ideia do estilo da especificação SSL da gramática abstracta IXDIRQL, considere-se o símbolo não-terminal *Operation* que permite a derivação de uma operação de conjuntos, de

uma operação lógica ou de um caminho. A produção (designada por *OPath*) relativa à derivação de *Operation*, a um caminho (*Path*) escrita em SSL é a seguinte :

$$
\begin{aligned}
&Operation : OPath\ (Path)\\
&\qquad \{\ Operation.aRes = Path.aRes;\\
&\qquad Path.aContext = Operation.aContext;\\
&\qquad Path.aPathOp = Operation.aPathOp;\ \};
\end{aligned}
$$

As regras semânticas associadas a esta produção (entre '{' e '}') calculam os valores do atributo sintetizado *aRes* (guarda o resultado do caminho) e dos atributos herdados *aContext* (guarda o conjunto de elementos a partir dos quais a procura dos novos elementos é feita) e *aPathOp* (corresponde ao operador de caminho filho ou descendente).

O editor construído é estruturado, sendo a edição dirigida pela sintaxe do IXDIRQL. Os símbolos não-terminais e pseudo-terminais da gramática são associados a indicações de introdução de informação. Estas consistem numa expressão que sugere o tipo de informação a introduzir, entre < e >. Quando um símbolo não-terminal pode ser derivado por diferentes produções alternativas, o utilizador escolhe a que lhe interessa num menu, como o que é mostrado na figura 2. Nesta figura, uma operação estruturada (derivada pelo símbolo *Operation* já referido) é indicada por $<Operation?>$ e pode ser uma das seis alternativas do menu. Estas alternativas correspondem às diferentes produções em que *Operation* deriva, dentre as quais a produção que deriva o caminho dada atrás como exemplo. Se a operação escolhida no menu for uma intersecção, o editor passa a ter o aspecto da figura 3, onde um dos operandos da intersecção foi já introduzido ($article/ref$).

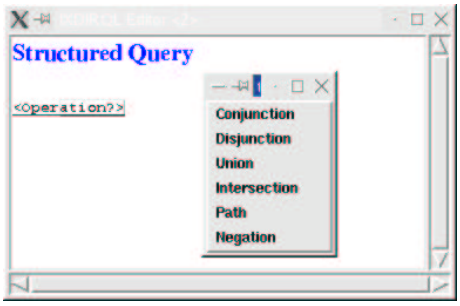


Fig. 2. O editor da linguagem IXDIRQL mostrando uma indicação de introdução de informação.

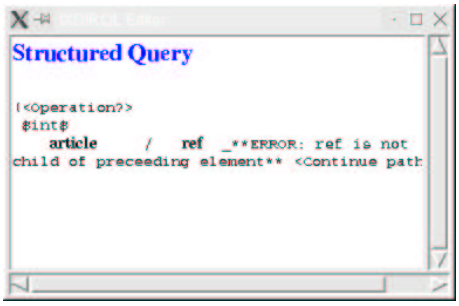


Fig. 3. O editor da linguagem IXDIRQL mostrando uma intersecção.

Todos os símbolos terminais da gramática são automaticamente colocados no sítio correcto, dispensando-se o utilizador dessa tarefa. Por exemplo, na figura 3, o operador *int* é introduzido automaticamente.

A semântica estática da linguagem IXDIRQL consiste na verificação da consistência entre os tipos de elementos introduzidos e as respectivas relações hierárquicas. Por exemplo, na figura 3, existe um erro semântico no caminho $/article/ref$ pois o tipo de elementos ref não é filho do tipo de elementos *article*. Durante a escrita de uma pergunta, estes erros são anunciados ao utilizador logo após a operação onde o erro é cometido. Assim, o utilizador pode corrigir imediatamente o seu texto.

6 Apresentação dos resultados

No protótipo construído, os resultados das perguntas são mostrados numa janela de visualização de documentos estruturados (foi escolhido o *Mozilla*). Após cada operação, a lista de elementos ou valores de verdade do resultado é incluída numa série de documentos HTML, dependendo do tamanho do resultado. A figura 4 é um exemplo de um resultado. Os resultados são ordenados por ordem decrescente da relevância dos elementos ou dos valores de verdade. Se as relevâncias forem iguais, é usada a ordem da colecção.

Na janela de visualização, cada elemento é apresentado com o seu identificador (que pode ser usado em operações de selecção), a sua relevância, um apontador para o documento respectivo e o seu conteúdo textual. Por sua vez, os valores de verdade são apresentados com a sua relevância, o identificador do elemento respectivo (por exemplo, o elemento associado ao filtro onde uma condição lógica é efectuada) e um apontador para o documento onde está inserido esse elemento. A figura 5 apresenta um exemplo de resultado composto por valores de verdade.

De momento, o acesso aos resultados intermédios é feito usando a janela de controle apresentada na figura 6. O utilizador acede ou não a cada resultado, respondendo às questões da janela. Quando o utilizador decide ver um resultado intermédio, este é-lhe mostrado, à parte, na janela de visualização (do tipo indicado nas figuras 4 e 5).

7 Utilização do protótipo por utilizadores reais

O protótipo foi posto à disposição de 5 utilizadores para efectuarem 3 procuras de informação pré-definidas. Por um lado, pretendeu-se testar a funcionalidade do sistema e, por outro, verificar se as perguntas introduzidas pelos utilizadores incluem operações de selecção. Isto significa que os utilizadores compreenderam as operações de selecção e souberam utilizá-las correctamente.

A interface do protótipo foi apresentada oralmente aos utilizadores. Os utilizadores foram escolhidos entre os membros do laboratório de informática onde

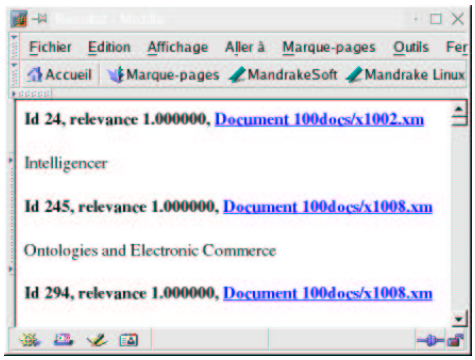


Fig. 4. A janela de visualização de um resultado composto por elementos.

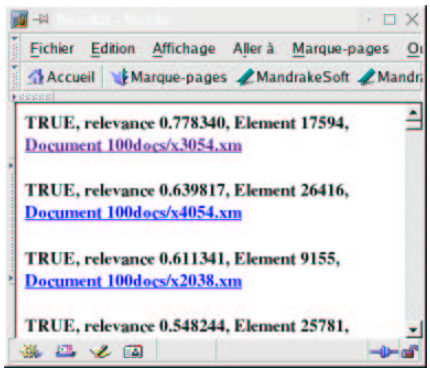


Fig. 5. A janela de visualização de um resultado composto por valores de verdade.

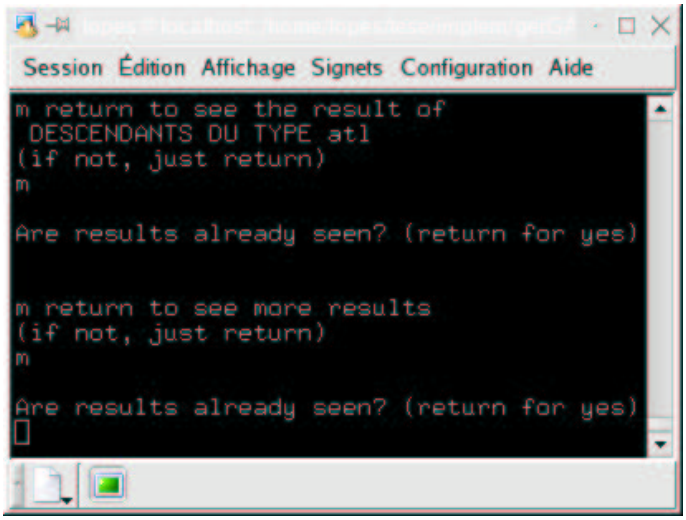


Fig. 6. A janela de controle para o acesso aos resultados intermédios.

o trabalho decorreu, com a condição de conhecerem XML e XPath. Assim, a linguagem IXDIRQL foi-lhes apresentada através de uma explicação sobre as operações de similaridade textual e de selecção.

A colecção de documentos utilizada consiste num conjunto de 100 artigos sobre *Inteligent Systems* da colecção INEX[3] [6], dos anos 2000 e 2001. Os artigos têm entre 110 e 2568 elementos. O DTD associado define, entre outros, os tipos de elementos e respectivas relações hierárquicas da figura 7.

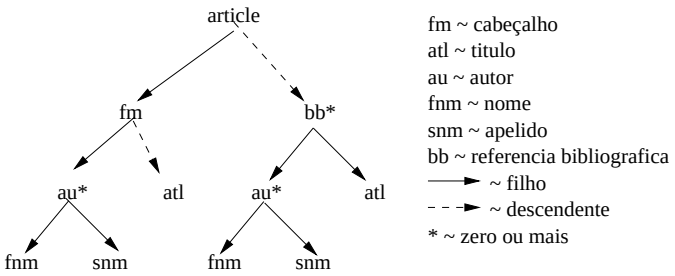


Fig. 7. Alguns tipos de elementos e relações hierárquicas do DTD INEX.

Na figura 8 pode ver-se um exemplo de documento da colecção onde existem elementos do tipo dos da figura 7. Este exemplo foi mostrado aos utilizadores, juntamente com a figura 7 e respectivas explicações.

```
<article>...
        <fm>...
                <atl>What's Next for the E-Book?</atl>
                <au><fnm>Giovanni</fnm><snm>Flammia</snm>...</au>...
        </fm>...
        <bb id="bibx10181">
                <au><fnm>B.N.</fnm><snm>Schilit</snm></au>
                <atl>"As We May Read: The Reading Appliance..."</atl>
        </bb>
        <bb id="bibx10182">
                <au><fnm>B-W.</fnm><snm>Chang</snm></au>...
                <au><fnm>J.</fnm><snm>Mackinlay</snm></au>...
                <atl>"Fluid Links for Informed and Incremental..."</atl>...
        </bb>...
</article>
```

Fig. 8. Um exemplo de artigo da colecção.

[3] Iniciativa para a avaliação da recuperação de informação em XML (*INEX: initiative for the evaluation of XML retrieval*), http://qmir.dcs.qmw.ac.uk/INEX/.

7.1 Os pedidos de informação

Foram propostos aos utilizadores três pedidos de informação baseados apenas nos tipos de elementos da figura 7. Esses pedidos foram definidos de forma a que o seu resultado implique a utilização de operações de selecção na construção das perguntas, a menos que os resultados sejam alterados manualmente. Assim, foram entregues aos utilizadores os pedidos seguintes:

1. Procurar as referências bibliográficas citadas em dois artigos do autor cujo nome é *"Giovanni"*.
2. Procurar as referências bibliográficas citadas em artigos cujo título é, na sua opinião, sobre *"XML"*.
3. A partir das referências bibliográficas obtidas em 2., procurar os autores daquelas cujo título é interessante para si (escolher pelo menos um título).

As perguntas que satisfazem os três pedidos anteriores são, respectivamente:

1. *article[fm//fnm='Giovanni'] SelectN 2//bb*
   article[fm//fnm='Giovanni'] Select $\{id_1, id_2\}$*//bb*
2. *article[fm//atl Sim 'XML' JudgeRel {...}]//bb*
3. *article[fm//atl Sim 'XML' JudgeRel {...}]//bb[atl Select {...}]//au*
   article[fm//atl Sim 'XML' JudgeRel {...}]//bb[atl Sim '...' JudgeRel {...}]//au

Na primeira pergunta, o operador *SelectN* permite reduzir o número de artigos do autor *"Giovanni"* a 2. Assim, as referências bibliográficas encontradas são citadas em 2 artigos. Outra possibilidade é fazer a selecção dos dois artigos através do operador *Select* aplicado a dois identificadores $\{id_1, id_2\}$.

A segunda pergunta procura, primeiro, os títulos de artigos sobre *"XML"* usando o operador *Sim*. De seguida, o operador *JudgeRel* reduz os artigos encontrados aos que o utilizador julga relevantes. Assim, as referências bibliográficas encontradas são as citadas nos artigos cujo título é, na opinião do utilizador, sobre *"XML"*.

Por fim, a terceira pergunta inclui o operador *Select* para que o utilizador escolha os títulos das referências bibliográficas que lhe interessam. Alternativamente, o utilizador pode fazer a procura de títulos usando *Sim* sobre um assunto que lhe interesse e seleccionar com *JudgeRel* aqueles que ele considera relevantes.

7.2 As perguntas feitas pelos utilizadores

As perguntas feitas pelos utilizadores podem pertencer a um dos quatro casos seguintes:

A: a pergunta inclui uma operação de selecção e satisfaz o pedido de informação;
B: a pergunta não inclui uma operação de selecção e não satisfaz o pedido de informação;

C: a pergunta inclui uma operação de selecção e não satisfaz o pedido de informação;
D: a pergunta não inclui uma operação de selecção mas satisfaz o pedido de informação porque o utilizador alterou manualmente os resultados.

A tabela 1 relaciona os 3 pedidos de informação e os 5 utilizadores através da classificação A a D da pergunta efectuada, indicando também os operadores de selecção que foram utilizados.

Pedido/Utilizador	**1**	**2**	**3**	**4**	**5**
1	A *SelectN*	A *Select*	A *SelectN*	A *SelectN*	A *Select*
2	A *JudgeRel*	A *JudgeRel*	B	B	B
3	A *JudgeRel*	A *Select*	B	A *Select*	A *Select*

Table 1. A classificação das perguntas feitas pelos utilizadores.

O comportamento dos utilizadores inclui aspectos que não podem ser controlados, como a concentração na execução da experiência, o nível de conhecimento do XPath, a compreensão do documento entregue sobre a experiência (que inclui explicações sobre IXDIRQL, a colecção de documentos e os pedidos de informação). Apesar desta dificuldade, pela tabela 1 constata-se o seguinte:

- Não há perguntas do tipo C. Todas as perguntas onde uma operação de selecção foi feita satisfazem o pedido, i.e. 100% das utilizações da selecção são correctas. Isto mostra que os utilizadores souberam sempre porque é que a selecção foi usada.
- O caso D nunca se verifica pois nenhum utilizador modificou os resultados manualmente. Em princípio, espera-se que um sistema de RI possa entregar o resultado desejado, sem ser necessário fazer alterações posteriormente.
- Nas 15 perguntas feitas pelos utilizadores, há 5 possibilidades de utilizar *SelectN* (no pedido 1), 1 de utilizar *Select* (nos pedidos 1 e 3) e 10 de utilizar *JudgeRel* (nos pedidos 2 e 3). As frequências verificadas são 2 em 5 para *SelectN* (40%), 6 em 10 para *Select* (60%) e 3 em 10 (30%) para *JudgeRel*. *Select* é o operador mais utilizado, o que pode mostrar que os utilizadores perceberam melhor a sua funcionalidade ou o julgam mais interessante ou fácil de aplicar.
- Por fim, o facto mais interessante é que, dentre as 15 perguntas efectuadas, 11 são do tipo A, i.e. em 73% das perguntas, os utilizadores recorrem à selecção correctamente.

8 Conclusão e trabalho futuro

Neste artigo pretendeu-se dar especial destaque a 2 assuntos relacionados com a recuperação incremental de informação em documentos estruturados, anotados

em XML.
Por um lado, pretendeu-se discutir algumas estratégias e decisões tomadas para implementar o sistema IXDIRQL de RI, montado sobre uma extensão ao XPath, que tem a capacidade de tratar perguntas baseadas na similaridade textual e oferece um modo de procura incremental. Nesta abordagem, que requereu uma adaptação dos cálculos de relevância tradicionais, uma pergunta pode ir sendo refinada através da selecção parcial dos resultados que se vão obtendo. Para o seu sucesso, enfatizámos a importância do recurso a 3 índices e a forma como os resultados parciais vão sendo mostrados no écran ao utilizador.
Por outro lado, falou-se do modo pragmático seguido numa primeira fase para validar o IXDIRQL, avaliando o comportamento de vários utilizadores a quem foi disponibilizado o sistema e fornecido um conjunto de perguntas.

Os testes realizados ao protótipo construído para o processamento da linguagem IXDIRQL mostraram que o mesmo satisfaz as características necessárias, pois permite o acesso aos resultados intermédios das perguntas e faz correctamente o cálculo de relevância de cada operação. Esse cálculo é incremental para que, após cada alteração da pergunta, apenas os cálculos dependentes do que foi alterado sejam efectuados. A utilização do protótipo por utilizadores mostrou, não só o seu correcto funcionamento, mas também que as operações de selecção são, em geral, correctamente utilizadas para obter o resultado de certas procuras de informação. Embora tenhamos consciência das limitações das experiências realizadas —as possíveis até à data— com um número reduzido de utilizadores, de questões e de documentos, o trabalho realizado mostrou o caminho que deve ser seguido nesta fase de validação da proposta e avaliação do sistema. Estes resultados, apesar de tudo, foram fundamentais para consolidar a proposta, evidenciando a sua exequabilidade.

Como trabalho futuro, apontamos alguns projectos que temos em mente, uns complementares, outros alternativos:

- Permitir a especificação de caminhos mostrando ao utilizador um documento exemplo da colecção onde ele selecciona os elementos desejados. Este mecanismo é conhecido em inglês por *"query by example"*.
- Estender o XQuery com operações de similaridade textual e com o paradigma interactivo de construção das perguntas (à semelhança do IXDIRQL). Construir um protótipo adequado para o processamento e avaliar a sua funcionalidade face a utilizadores reais.
- Desenvolver uma metodologia para utilizar o XQuery assim estendido para interrogação de fontes de informação organizadas com ontologias, como sucede na Semantic Web, criando assim um novo sistema a que chamámos OntIX-Query. Para tal, pretendemos explorar duas hipóteses: (1) fazer as perguntas aos documentos XML da Web e refinar a resposta obtida usando a meta-informação ontológica associada ao recurso; (2) fazer as perguntas à ontologia em uso para integrar os recursos e recuperar os documentos associados a cada componente da ontologia obtido como resposta.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.
2. A. Berglund, S. Boag, D. Chamberlin, M. Fernandez, M. Kay, J. Robie, and J. Siméon. XML Path Language (XPath) 2.0 W3C Working Draft. http//www.w3c.org/xpath20/, October 2004.
3. S. Boag, D. Chamberlin, M. Fernandez, D. Florescu, J. Robie, and J. Siméon. XQuery 1.0: An XML Query Language. W3C Working Draft. http://www.w3.org/TR/xquery/, October 2004.
4. T. T. Chinenyanga and N. Kushmerick. Expressive Retrieval from XML Documents. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information retrieval (SIGIR'01)*, New Orleans, Louisiana, USA, September 2001.
5. A. Deutsch, M. Fernandez, A. Levy, and D. Suciu. XML-QL: A query language for XML. W3C Note, August 1998.
6. N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas. INEX: Initiative for the Evaluation of XML Retrieval. In R. Baeza-Yates, N. Fuhr, and Y. Maarek, editors, *Proceedings of the Workshop on XML and Information Retrieval, International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02),* , Tampere, Finland, August 2002.
7. N. Fuhr and K. Grobjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information retrieval (SIGIR'01)*, New Orleans, Louisiana, USA, September 2001.
8. A. Gançarski and P. Henriques. Interactive Information Retrieval from XML Documents Represented by Attribute Grammars. In *Proceedings of the 2003 ACM Symposium on Document Engineering*, Grenoble, France, November 2003.
9. A. Gançarski and P. Henriques. IXDIRQL: an Interactive XML Data and Information Retrieval Query Language. In *Proceedings of the 7th ICCC/IFIP International Conference on Electronic Publishing*, Guimarães, Portugal, June 2003.
10. M. Kuiper and J. Saraiva. Lrc: A generator for incremental language-oriented tools. In K. Koskimies, editor, *7th International Conference on Compiler Construction*, volume 1383, pages 298—301. Lecture Notes in Computer Science, April 1998.
11. T. Reps and T. Teitelbaum. *The Synthesizer Generator Reference Manual.* Texts and Monographs in Computer Science. GrammaTech, Inc., 1993.