DOI:
10.1007/s10791-010-9144-6

Document Version
Early version, also known as pre-print

Specificity Aboutness in XML Retrieval

Tobias Blanke and Mounia Lalmas

Department of Computing Science, University of Glasgow
tobias.blanke@dcs.gla.ac.uk
mounia@acm.org

Abstract. This paper presents a theoretical methodology to evaluate filters in XML retrieval. Theoretical evaluation is concerned with the formal investigation of qualitative properties of retrieval models. XML retrieval deals with retrieving those document components that specifically answer a query, and filters are a method of delivering the most focused answers. Our theoretical evaluation will critically analyse how filters achieve this.

1 Introduction

According to INEX, the evaluation initiative for XML retrieval [6], the aim of XML retrieval is to retrieve not only relevant document components, but those at the right level of granularity, i.e. those that specifically answer a query. To evaluate how effective XML retrieval approaches are, it is necessary to consider whether the 'right' level of the structure is correctly identified. For this purpose, INEX has developed a new relevance criterium next to general relevance, which measures how focused an XML element is with respect to an information need. The general relevance of an element is captured in the INEX exhaustivity dimension[1] while the specificity dimension indicates the focus.

In this paper, we analyze retrieval models developed at INEX that aimed at delivering results that specifically answer a query. Delivering these so-called *most specific* answers has proven to be a complex retrieval task. In addition, it has been noted that traditional information retrieval (IR) evaluation might not be sufficient to properly assess the effectiveness of such more complex retrieval tasks [10]. This paper proposes an alternative theoretical evaluation that complements an experimental evaluation, especially when dealing with complex retrieval tasks such as those developed for XML retrieval.

A theoretical evaluation can be done through the use of a meta-theory, as proposed in previous work based on the logical approach to IR [7]. Van Rijsbergen and others have expressed logical relevance in terms of the implication $d \rightarrow q$ [10]. Chiaramella [4] used two implications to describe the XML retrieval task[2]: $d \rightarrow q$ characterizing exhaustivity and $q \rightarrow d$ characterizing specificity.

[1] Since 2006, INEX does not refer to exhaustivity anymore, just relevance and specificity.

[2] When this work was published, it referred to the more general case of structured document retrieval, for which XML retrieval is a special case.

Following Huibers' work [7], we call such implications between query and document *aboutness*. IR models propose specific ways to implement the aboutness of a document to a query. With this view in mind, the theoretical evaluation of an IR model thus consists of characterizing aboutness and investigating its underlying reasoning process.

Aboutness has been discussed sporadically in IR literature, most notably in the work of Lalmas and Van Rijsbergen [11], Huibers and Bruza [3], and recently Wong et al.[13]. However, aboutness has yet to be applied to more complex IR tasks such as those occurring in XML retrieval. In this paper, we use the concept of aboutness to evaluate XML retrieval models that aimed at identifying the most specific document components for a query.

In INEX, the retrieval task that aims at finding the most specific answers has been referred to as the focused task. This is to be compared to the thorough task, that aims at estimating the relevance of document components to a query. In this latter task, all relevant document components are to be identified, and then ranked according to their degree of relevance. In the focused task, the result set should consist of non-overlapping document components, ranked according to how specific they are to the query. Overlap occurs when a document component (e.g. a section) and one of its descendent (e.g. a paragraph in this section) or ascendent (e.g. the chapter containing that section) are both returned as answers. The aim of the focused task is therefore to identify among overlapping document components, the component that is the most specific to the query, and to return it as what is referred to as a *focused* answer. In this paper, we concentrate on retrieval models developed for the focused task at INEX 2005, as the fundamentals of these models have not changed much since. In addition, we restrict ourselves to models aiming at delivering these focused answers for content-only queries.

Models developed at INEX to implement the focused retrieval task can be viewed as filters. Indeed, these models mostly consist of the post-processing of an answer set produced by models aiming at implementing the thorough retrieval task. The post-processing phase consists of eliminating all but the most focused document components from the answer set. We therefore analyze filters as an aboutness decision in their own right. Several types of filters have been developed in INEX. The most popular filter is the so-called brute-force one, which eliminates all but the most relevant elements[3] on a particular XML path. However, in the experimental evaluation, it performs less well than others that look at the relationships between elements [8]. We therefore compare it to an alternative approach based on the re-ranking of elements.

This paper is organised as follows: In Section 2, we introduce the background of our theoretical evaluation methodology. In Section 3, we briefly draw on earlier results to demonstrate parts of the theoretical evaluation of two XML retrieval approaches implementing the thorough task. We then introduce in Section 4 our theoretical methodology to analyse filters as aboutness decisions, before applying

[3] In this paper, elements and document components are used interchangeably.

it to the brute-force and re-ranking filtering models in Section 5. Finally, we relate our findings to those of the experimental evaluation in INEX in Section 6.

2 Theoretical evaluation background

In this section, we introduce the steps of our theoretical evaluation methodology (see [2] for a complete overview). A theoretical evaluation methodology needs a formalism powerful enough to characterize the fundamental properties of retrieval models. Following Huibers, we use Situation Theory (ST), developed by Barwise and Perry [1], for this purpose. ST is a mathematical theory of meaning and information with *situations* as primitives [7]. Situations are partial descriptions of the world and are composed of *infons*. For IR modelling, queries and documents are modelled as situations, while infons represent a model's information items like keywords or phrases.

Using ST, we model documents and queries as situations [3]. Let document D and query Q be situations, then $D \,\Box\!\!\rightsquigarrow\, Q$ means that the information in D is about the information need expressed in Q. For instance, in standard IR, a document containing 'garden' and 'house' would be about a query asking for 'garden'. Likewise, $D \,\Box\!\!\not\rightsquigarrow\, Q$ symbolises that D is not about Q. For XML retrieval, we can use Chiaramella's distinction and say that $D \,\Box\!\!\rightsquigarrow\, Q$ symbolizes exhaustivity and $Q \,\Box\!\!\rightsquigarrow\, D$ specificity. With $\otimes$, we formalise the composition of situations, while $\equiv$ states that two situations are equivalent, i.e. they contain the same information.

Translation is the symbolic representation of an IR model's handling of information using a formal language. It is formally represented by a function *map* that 'maps' situations to their formal representation. In IR, mapping a document (or a document component) to its formal representation corresponds to the indexing process. For standard IR, the outcome would consist of a set of infons represented by $\langle\langle k \rangle\rangle$, where k stands for an indexing term. A set of infons is a situation: $\{\langle\langle k_1 \rangle\rangle, \langle\langle k_2 \rangle\rangle\}$. An example would be $\{\langle\langle house \rangle\rangle, \langle\langle garden \rangle\rangle\}$.

For representing information in XML retrieval, we furthermore use N-ary relationships R between infons i_j, to model relationships: $\langle\langle R, i_1, ..., i_n \rangle\rangle$. For instance, a section with two paragraphs will be symbolized by: $\{\langle\langle ElementType, Sec, s \rangle\rangle, \langle\langle ElementType, Para, p_1 \rangle\rangle, \langle\langle Value, garden, p_1 \rangle\rangle$, $\langle\langle ElementType, Para, p_2 \rangle\rangle, \langle\langle Value, house, p_2 \rangle\rangle, \langle\langle Parent, s, p_1 \rangle\rangle, \langle\langle Parent, s, p_2 \rangle\rangle\}$. This reflects the fact that each XML element has an element type infon, expressed with the relation $ElementType$. Content infons (i.e. the actual text in the element) are modeled as $Values$. The relation $Parent$ expresses that the two paragraphs (p_1 and p_2) are the children of the section (s). Translation is therefore based on building a document representation through indexing, which according to Van Rijsbergen [12] leads to the view that index terms represent properties of documents (or document components), which may then be studied.

Next to the symbolic characterization of an IR model, we need means to describe the functional behavior of the model, i.e. what makes a document (or document component in XML retrieval) about a query. This is done through

so-called *reasoning rules*. Indeed, an IR model's aboutness decision is specified by the reasoning rules it incorporates. These can be either fully, partially, or not at all supported. Together with the symbolic representation of documents, queries and information for an IR model, they make up the *aboutness decision system* that characterizes how the model decides that a document (or document component) is relevant to query.

An example of a rule is Left Monotonic Union (LMU), which plays an important role in our theoretical study of filters. LMU states that if a document D is about a query Q, then also the composition of D and D':[4]

- LMU: If $D \boxdot\!\!\rightarrow Q$, then also $D \otimes D' \boxdot\!\!\rightarrow Q$.

By comparing the reasoning rules each decision system incorporates and the way it does so, we are able to give an overall comparison of the behaviour of the retrieval model characterized by the aboutness decision system.

There are over 20 reasoning rules to be considered in the theoretical analysis of retrieval models (see [7] and [13]). In this paper, we restrict ourselves to the following rules (including LMU), as they are sufficient for our investigation:

- Reflexivity: $S \boxdot\!\!\rightarrow S$.
- Transitivity: If $S \boxdot\!\!\rightarrow T$ and $T \boxdot\!\!\rightarrow U$, then also $S \boxdot\!\!\rightarrow U$.
- Euclid: If $S \boxdot\!\!\rightarrow T$ and $S \boxdot\!\!\rightarrow U$, then also $T \boxdot\!\!\rightarrow U$.
- Mix: If $S \boxdot\!\!\rightarrow T$ and $U \boxdot\!\!\rightarrow T$, then also $S \otimes U \boxdot\!\!\rightarrow T$.

3 Aboutness in XML retrieval

To carry out our theoretical evaluation of filters (the focused retrieval task at INEX), it is necessary to present (albeit briefly) the theoretical evaluation of models developed for the thorough task. This is because of the relationships between the tasks, one being a post-processing of the other. This also provides an illustration of our theoretical methodology. We focus on two such models, both building upon well-known flat document retrieval models.

3.1 Vector Space Model

A vector space model for XML retrieval is presented in [8]. There, XML documents are split into several disjoint indexes of the most useful components (which can be determined for a given application). In the model, a standard vector space approach is used to retrieve from a query (Q) XML elements (D) instead of full documents:

$$rsv(Q, D) = \frac{\sum_{t_i \in \{Q \cap D\}} w_Q(t_i) * w_D(t_i) * idf(t_i)}{\|Q\| * \|D\|}$$

[4] Throughout the paper, we use upper case letters from the middle of alphabet such as S, T, for situations if we are not talking about queries and document components. In that case we use Q and D. Anything that situations are made of, e.g. keywords but also structural relationships, is symbolized with letters from the beginning of the alphabet like A or B.

where $w_Q(t) = \frac{log(TF_Q(t))}{log(AvgTF_Q)}$ and $w_D(t) = \frac{log(TF_D(t))}{log(AvgTF_D)}$. $||Q||$ and $||D||$ are the numbers of unique terms in Q and D, respectively. Both are scaled by the average document length in the collection [8].[5]

Structure is used in the model mainly to allocate document components across different indexes. The translation is limited to those infons of document components most commonly assessed as relevant. In the INEX 2005 collection, these included paragraph ('Para','Para1'), subsection ('SS1', 'SS2'), section ('Sec'), etc. Regarding the translation, let A by a document component and e be an element type, then $map(A) = \{\langle\langle ElementType, e, i\rangle\rangle, \langle\langle Value, t, i\rangle\rangle | e \in \{Art, Abs, Sec, SS1, SS2, Para, Para1\}$.

For the aboutness decision, let Q be a query and D be a document component. That we consider components instead of full documents is the main difference to the flat vector space model. The *XML retrieval vector space aboutness decision* is then defined by:

$$D \sqsubset\!\!\rightarrow Q \text{ if and only if } rsv(D, Q) \geq n$$

In the model, only the top N documents are considered. We call the value that has to be reached in order to be part of the top N documents n. Thus, the model implements thresholded vector space retrieval [13].

Regarding the reasoning rules, we can prove that LMU is conditionally satisfied. As the model implements thresholded vector space retrieval, the extension of D to $D \otimes D'$ will only continue to be about Q if there is still sufficient information overlap between $D \otimes D'$ and Q (the full proof can be found in [2]).

3.2 Language Models

A second model, that was based on a model for flat document retrieval and that performed well at INEX, uses language modeling [9]. The model builds several indexes, each of which is separately populated: one for all elements, one length based one, one for elements frequently assessed as relevant, one for sections. The full article is kept in another index. A language model for each document component is calculated by interpolating the element ($P_{mle}(t_i|e)$), the document ($P_{mle}(t_i|d)$) and the collection ($P_{mle}(t_i)$) language models:

$$P(t_i|e) = \lambda_e * P_{mle}(t_i|e) + \lambda_d * P_{mle}(t_i|d) + (1 - \lambda_e - \lambda_d) * P_{mle}(t_i)$$

This model is built on the decision that a document component D is about a query Q if and if only the information in Q can be found in the indexes. We actually have different aboutness decisions, depending on which index is used to generate the element language model. We therefore must provide a *map* function to translate infons for each chosen index. We only demonstrate 2 of the 6 indexes, as the others are built very similarly. Let A be an element with term t and an element type e:

[5] The obtained scores are modified using an Automatic Query Refinement (AQR) approach based on Lexical Affinity (LA), which is out of scope for this paper.

– Length based index: $map_{length}(A) \equiv \{\langle\langle ElementType, e, i\rangle\rangle, \langle\langle Value, t, i\rangle\rangle | |A| > \kappa\}$
– Section index: $map_{sec}(A) \equiv \{\langle\langle ElementType, e, i\rangle\rangle, \langle\langle Value, t, i\rangle\rangle | e \in \{Sec\}\}$

where κ is a threshold that discards small elements. Apart from the article index, the main difference to a flat document language model is the division into document components instead of documents. This *XML language model retrieval aboutness decision* is the same as for the flat document language model: D about Q if and if only $P(t_i|e) > \theta$. The threshold θ is the smoothing value, which is the collection language model $(1 - \lambda_e - \lambda_d) * P_{mle}(t_i)$. Contrary to the vector space model threshold, it is internal to the aboutness decision, as it is dependent on the overall distribution of the terms in the collection. This allows the model to be adjusted well to specific collections like INEX. We have shown in [2] that LMU is unconditionally supported.

Both discussed models performed well at INEX, which we could relate to their aboutness behaviour in [2]. However, both models performed better for the thorough retrieval task than for the task aiming at returning the most focused elements, i.e. the focused retrieval task. This paper provides a theoretical explanation for this behaviour.

4 Defining specificity aboutness

In this section, we describe our theoretical methodology to evaluate filters. We rely on some initial work by Huibers on the relationship between the filter aboutness system (characterizing the focused task) and the corresponding underlying aboutness system (characterizing the thorough task) [7], which we adapt to the requirements of XML retrieval. We go beyond his work by actually applying his theoretical work to analyze two filters developed at INEX in Section 5.

As already explained, the task of finding the most focused elements consists of filtering the ranked result list produced by an XML retrieval model like the two described in Section 3. Generating this ranked result list is itself based on an aboutness decision system, which characterizes the model used to deliver that list. Thus, with filtering, a further aboutness decision is applied, one which removes overlapping elements from the result list.[6]

Huibers [7] describes that one aboutness decision system is a filter to another aboutness decision system if the two corresponding aboutness systems are embedded — meaning their reasoning behaviour is related by supporting the same or sufficiently similar properties. In the context of XML retrieval, this translates

[6] It should be pointed out that the use of filters is not exclusive to XML retrieval. Filters are used in IR to improve performance [7], if, for instance, at first a fast but less accurate approach is used to identify relevant documents from a very large set documents, and then a second retrieval system is used to search the initial result set more accurately. Pseudo-relevance feedback and passage retrieval are examples of such a process.

to having to relate the aboutness decision system associated with the model for the focused task to that of the underlying aboutness system associated with the model used to generate the ranked list (the thorough task) to then be filtered.

The theoretical analysis of filter is done in three steps. We first formalize the translation process, as we did in Section 3 for retrieval systems. Secondly, we identify the reasoning rules associated with the filter. Finally, we analyse the relationship between the filter and the underlying aboutness systems. For the later, we make use of the filtering function f-*answer* defined in [7], which we adapt to XML retrieval:

Definition 1 *Let A_p, B_p be aboutness systems and $\mathcal{D}$ be a set of documents and Q be a query. The filtering function f-answer of A_p with respect to B_p is defined by: f-answer$(A_p; B_p; Q; \mathcal{D})$ = answer$(A_p; Q;$ answer$(B_p; Q; \mathcal{D}))$, where answer describes a function that delivers an answer set from the set $\mathcal{D}$ based on query Q.*

Using this definition, we can investigate the filtering process by looking at the relationship between f-*answer* and *answer*. Without going into detail, Huibers has identified three important distinctions between f-*answer* and *answer* [7]:

- A filtering function f-*answer*$(A_p; B_p; Q; \mathcal{D})$ is called *useless* if for all sets of documents $\mathcal{D}$ and queries Q f-*answer*$(A_p; B_p; Q; \mathcal{D})$ = *answer*$(B_p; Q; \mathcal{D})$. An example of a useless filter is the application of the coordinate retrieval model as a filter to an answer set generated by simple vector space retrieval, as both are based on the same aboutness decisions, according to which a document D is about a query Q if both share information items.
- The aboutness systems A_p and B_p are said to be *f-equivalent* if and only if f-*answer*$(A_p; B_p; Q; \mathcal{D})$ = *answer*$(A_p; Q; \mathcal{D})$. An example of an *f-equivalent* filter is to use strict coordinate retrieval to filter a result set generated by vector space retrieval. Strict coordinate retrieval defines that a document D is about a query Q if and only if the information items of Q are a subset of the information items in D. This delivers a subset of the answer set from simple vector space retrieval, for which D is about Q if they share information. Strict coordinate retrieval therefore fully determines the final answer set.
- A_p and B_p are said to *intersect* if and only if the filter is neither useless nor f-*equivalent*.

In our analysis of the relationship between f-*answer* and *answer*, we first determine whether a filter is 'useless', i.e. the filtering function does not change the original answer set. If this is not the case, next we investigate whether the filter f-*answer* uses f-*equivalent* aboutness systems. We call a filter aboutness system to be f-*equivalent*, if its A_p alone will determine the final result set. If the filter is not useless and not f-*equivalent* with regard to the underlying aboutness system, we then define how the filter and underlying aboutness system 'intersect' by comparing their aboutness properties.

The following section will demonstrate the presented methodology for the analysis of two filers at INEX.

5 Applying specificity aboutness at INEX

Two main types of models have been proposed for the focused task at INEX: a simple model that keeps the highest ranked element of each XML path and a more complex model that takes into account the relations in the tree hierarchy between retrieved elements.

5.1 Brute-force filter

Our first method of removing overlap in the result set of an XML retrieval model has also been referred to as 'brute-force filter', because only the highest scored element from each of the paths is selected. The advantage of this filter is that it is relatively easy to implement and that it can be used on top of any kind of underlying aboutness system.

Aboutness decision The aboutness decision of brute-force filtering can be defined as:

$$D \text{ about } Q \text{ if and only if } rsv(D,Q) = max(rsv_u(D,Q))$$

$max(rsv_u(Q,D))$ is delivering the XML element with the maximum retrieval status value for the underlying aboutness system. For the translation, let A be a document component, e_n element types, k_n values in an element, and i an identifier to enumerate all $\{1,...,n\}$ elements in an XML tree in a depth-first traversal manner:
$map(A) = \{\langle\langle ElementType, e_1, i_1\rangle\rangle, \langle\langle ElementType, e_2, i_2\rangle\rangle, \langle\langle Parent, i_1, i_2\rangle\rangle,$
$..., \langle\langle ElementType, e_n, i_n\rangle\rangle, \langle\langle Parent, i_{n-1}, i_n\rangle\rangle, \langle\langle Value, e_n, k_1\rangle\rangle, ..., \langle\langle Value, e_n,$
$k_n\rangle\rangle\} | \forall i_i \in \{\langle\langle Parent, i_i, i_k\rangle\rangle\}, count(i_i) = 1\}$.
The translation expresses that we only consider elements on the same XPath, meaning each element is the parent and the child of exactly one other element, unless it is the root or leaf element.

Reasoning behaviour We now continue analysing the functional behaviour of brute-force filtering using the reasoning rules from Section 2. Reflexivity holds for brute-force filtering. A maximum element will be about itself. More interesting are those reasoning rules that are not supported: The Transitivity rule, for instance, is not supported, as two situations cannot be the maximum scoring answers towards the same query. If T is the maximum scoring answer to U, S cannot be the maximum scoring answer to U, too. This means whatever the status of Transitivity in an aboutness system, if we apply brute-force filtering on top of it, it will not be supported. The same applies for Euclid from Section 2: If S is the maximum scoring answer to U, how could T be the maximum scoring answer to the same U, too? This means Euclid is never supported.

Mix is another rule that cannot be supported. It states that with the assumptions $S \boxdot\!\!\rightsquigarrow U$ and $T \boxdot\!\!\rightsquigarrow U$, we can also say that $S \otimes T \boxdot\!\!\rightsquigarrow U$. S and T, however,

cannot be at the same time the maximum answer to U. The assumptions contradict each other. LMU would imply in the context of brute-force filtering that if one extends S to $S \otimes U$ and aboutness would be preserved for both, both S and $S \otimes U$ would be maximum scoring answers, which is a contradiction. This means LMU is not supported either.

All the rules analysed in this section are important in the analysis of XML retrieval models' behaviour [2]. When we analyse the experimental results related to brute-force filtering in Section 6, we shall see the impact of excluding the rules' reasoning behaviour.

F-answer In this section, we shall look at the relation between f-*answer* and *answer*. First, we need to show that the brute-force filter is not *useless*. This can be formally proven by demonstrating that the aboutness systems of filter and underlying system differ in at least one reasoning characteristics — be it a certain rule, be it a single condition of this rule. We have just seen that brute-force filtering disallows LMU, Transitivity, etc., which means it is not useless as a filter for both the XML vector space and language model retrieval models (and many XML retrieval approaches we have analysed in [2]). As $max(rsv_u(D, Q))$ is dependent on the underlying retrieval status value rsv_u, brute-force filtering is also not f-*equivalent*.

As the filter is neither useless nor f-*equivalent*, neither brute-force filtering nor the underlying aboutness systems from Section 3 fully determine the outcome of combining both. They 'intersect', which means that we need to look at the differences in reasoning behaviour, the filter creates: E.g., LMU reasoning is excluded, which will change any aboutness system that follows the strict structural constraints of XML documents: If an element is a child, it will share information with its parent. This means for language modeling from Section 3.2, for instance, that both are about the same queries. However, such aboutness due to overlap in (redundant) information is what is supposed to be excluded by brute-force filtering.

We analyse the impact of brute-force filtering on the experimental results in INEX 2005 in Section 6, but first we look at a second alternative approach to dealing with overlapping elements: re-ranking. The assumption is here that sometimes overlap can be beneficial. In [8], they used a similar kind of re-ranking and found that it delivers better performance than the brute-force filtering alone.

5.2 Controlling the overlap: Re-ranking approach

The next approach [5] we present will re-rank the elements with a new context-dependent retrieval status value, but not entirely eliminate overlapping elements. The approach is based on iteratively reducing the score of those elements that contain highly relevant elements. The input into the re-ranking method is a list of XML elements x. These are each associated with $x.\overrightarrow{f}$ as the term frequency vector per query term and with $x.\overrightarrow{g}$ as the adjustment vector, and other information required to process the algorithm such as the set of children per element.

The adjustment of each term x_t is based on $x_t = f_t - \alpha * g_t$, where α is an adjustment weight. For parents y containing a highly scoring child x their adjustment score $y.\overrightarrow{g}$ will be increased. For the children of highly ranked parents, we know that its terms have already been considered in the reported parent element. Hence, its $x.\overrightarrow{g}$ will become $y.\overrightarrow{f}$. The tree is traversed until all elements are covered and re-ranked.

Aboutness decision According to the algorithm in [5], no element will be filtered out unless the adjusted score becomes 0. Therefore, the aboutness decision is described by:

$$D \text{ about } Q \text{ if and only if } rsv_{adjusted}(D, Q) > 0$$

The translation of the model is out of scope for this paper, as it would require a deeper analysis of how XML structures can be translated into situations. We have done that analysis in [2].

Reasoning behaviour The first reasoning property to look at will be Reflexivity, which as seen in Section 2 states that $S \sqcap\!\!\sim S$. Reflexivity is not given. With $S \sqcap\!\!\sim S$, then $f_t = g_t$. If in $x_t = f_t - \alpha * g_t$, $\alpha = 1$ [5], then $x_t = f_t - 1 * g_t$, which means $rsv_{adjusted} = 0$, with $f_t = g_t$. Thus, Reflexivity is not supported.

Re-ranking does not fundamentally change the aboutness decision of the XML retrieval models but adds emphasis to the ranking of elements. For our analysis of the impact of filters we therefore need to relate it to the models we have developed in Section 3 directly. For both models, Transitivity, Euclid and Mix behaviour, will not be changed. LMU would be given if $S \otimes U \sqcap\!\!\sim T$ and $S \sqcap\!\!\sim T$ are given. Regarding the XML vector space model, re-ranking with $x_t = f_t - \alpha * g_t$ can of course reduce the extension to fall below the threshold n. This will mainly effect the children of the highly ranked parents. LMU is only conditionally supported if re-ranking does not lower the retrieval result to fall below n. An interesting case forms the language modeling approach in Section 3.2. Its internal threshold based on the smoothing value might be missed if the added information leads to a re-ranking below the smoothing value. Therefore, applying re-ranking on top of language modeling means that LMU is now conditionally supported, while language modeling alone fully supported LMU.

F-answer Re-ranking is certainly not *useless*, because the LMU thresholds for vector space retrieval and language modeling have been changed. It will not be f-*equivalent* either, as it is dependent on the underlying aboutness decision, because re-ranking is a function of the original retrieval status value. Thus, re-ranking will also be 'intersecting'. Reflexivity is changed through the impact of α. That Transitivity and Mix behaviour is preserved is a clear advantage towards the brute-force filtering approach, as both are important properties of XML retrieval behaviour [2]. In particular the support for Mix, will add to the better performance of the model in the experimental results.

In the next section, we briefly look at how conclusions from the theoretical evaluation of both filters help explain experimental behaviour in INEX 2005.

6 Impact of filters on experimental behaviour at INEX

We first investigate the impact of brute-force filtering on the experimental behaviour in INEX 2005. The XML vector space retrieval model has been overall very successful in the experimental evaluation in INEX 2005 [8], but its performance decreases for the tasks to deliver only non-overlapping document components, ranked according to how specific they are to the query. They implemented these tasks by using various filters. Particularly, in their run which used simple brute-force filtering, the performance was much worse. The situation is similar for the XML retrieval model based on language modelling [9]. Its performance decreases, too, when brute-force filtering is used to filter the original language modeling retrieval results.

If we try to understand why brute-force filtering decreases performance in XML retrieval, two changes of reasoning properties are highly conclusive:

1. LMU is not supported by brute-force filtering. The XML vector space retrieval model, for instance, successfully used conditions on LMU reasoning to adjust the behaviour of flat document vector space retrieval to the requirements of XML retrieval [2]. This ability is lost once the brute-force filter is applied, which will explain a decrease in performance.
2. Mix is not supported by brute-force filtering. Among other things, Mix describes that, if two children D and D' are about a query, then their parent item $D \otimes D'$ will also be about the same query. This behaviour is typical to XML based resasoning. If it is not supported, problems might arise, such as the elimination of potentially highly relevant children. Say, we have one relevant child and a more relevant parent, then the child will be eliminated from the result set after applying brute-force filtering. Another child of the same parent that is about the same query, will also be eliminated, as the parent is already chosen for its path. However, this child might be highly relevant, too.

Looking at the second filter, re-ranking, it is difficult to make general statements regarding its impact on XML retrieval, as it has been developed for a particular model [5]. The authors, however, report limitations of their algorithm according to their experimental evaluation [5]. From a theoretical evaluation point of view, an immediate recommendation on how to potentially improve the model would be to introduce a threshold to control the monotonic behaviour of the re-ranking aboutness decision: Only if $rsv_{adjusted}(D, Q) > \theta$, the element would be reported. We have seen in earlier theoretical evaluations [2], that thresholds effectively add to the control of the monotonic behaviour and improve models' performance.

7 Conclusion

In this paper, we have shown how a theoretical evaluation (in this paper based on ST), can aid the analysis of filters in XML retrieval. To this end, we introduced a theoretical evaluation methodology to help investigate filters based on an ST formalism. We have considered filters as a second layer aboutness decision and asked how they influence the underlying aboutness system. We could do so, as we regarded them as aboutness systems. This has led to conclusions about why and how they change the performance of their underlying systems in the experimental evaluation in INEX. Our primary interest has been whether the filters are suitable extensions of the underlying aboutness decision. We could show how particularly brute-force filters significantly change the underlying aboutness behaviour of the retrieval models. In the future, we we would like to deepen our analysis of how filters for focussed retrieval have an impact on aboutness behaviour, especially on specificity aboutness, by analysing in more detail the impact on experimental results in INEX 2005.

Acknowledgements Mounia Lalmas is currently funded by Microsoft Research/Royal Academy of Engineering.

References

1. J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, 1983.
2. T. Blanke and M. Lalmas. A framework for the theoretical evaluation of XML retrieval. Paper in preparation for publication.
3. P. D. Bruza and T. W. C. Huibers. Investigating aboutness axioms using information fields. In *ACM SIGIR '94*, pages 112–121, 1994.
4. Y. Chiaramella. Information retrieval and structured documents. In *Lectures on information retrieval*, pages 286–309. 2001.
5. C. L. A. Clarke. Controlling overlap in content-oriented XML retrieval. In *ACM SIGIR '05*, pages 314–321, 2005.
6. N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Advances in XML Information Retrieval and Evaluation (INEX 2005), Dagstuhl*, 2006.
7. T. W. Huibers. *An Axiomatic Theory for Information Retrieval*. Universiteit Utrecht, PhD Thesis, 1996.
8. Y. Mass and M. Mandelbrod. Using the INEX environment as a test bed for various user models for XML retrieval. In Fuhr et al. [6], pages 187–195.
9. B. Sigurbjörnsson and J. Kamps. The effect of structured queries and selective indexing on XML retrieval. In Fuhr et al. [6], pages 104–118.
10. C. J. van Rijsbergen. Towards an information logic. In *ACM SIGIR '89*, pages 77–86, 1989.
11. C. J. van Rijsbergen and M. Lalmas. Information calculus for information retrieval. *J. Am. Soc. Inf. Sci.*, 47(5):385–398, 1996.
12. C. J. v. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.
13. K.-F. Wong, D. Song, P. Bruza, and C.-H. Cheng. Application of aboutness to functional benchmarking in information retrieval. *ACM Trans. Inf. Syst.*, 19(4):337–370, 2001.