A Framework for the Theoretical Evaluation of XML Retrieval

Tobias Blanke

King's College London
Centre for e-Research, 26-29 Drury Lane
WC2B 5RL, London, Uk
tobias.blanke@kcl.ac.uk

Mounia Lalmas

Yahoo! Research Barcelona, Avinguda Diagonal 177
08018 Barcelona, Catalunya, Spain
mounia@acm.org

Theo Huibers

University of Twente
Department of Computing Science, Drienerlolaan 5
7522 NB Enschede, The Netherlands
theo.huibers@thaesis.nl

This article presents a theoretical framework to evaluate XML retrieval. XML retrieval deals with retrieving those document components, the XML elements, that specifically answer a query. In this paper, theoretical evaluation is concerned with the formal representation of qualitative properties of retrieval models. It complements experimental methods by showing the properties of the underlying reasoning assumptions that decide when a document is about a query. We define a theoretical methodology based on the idea of 'aboutness' and apply it to current XML retrieval models. This allows comparing and analyzing the reasoning behaviour of XML retrieval models experimented within the INEX evaluation campaigns. For each model we derive functional and qualitative properties that qualify its formal behaviour. We then use these properties to explain experimental results obtained with some of the XML retrieval models.

Background and Motivation

According to INEX, the evaluation initiative for XML retrieval (Gövert, Kazai, Fuhr, & Lalmas, 2006), 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. XML, contrary to HTML, separates the logical structure of documents from the layout. The logical structure of an XML document forms a tree of elements, which starts with a root element and has edges between elements.[1] Queries can also contain structural hints or be purely content-based.

To evaluate how effective XML retrieval approaches use this structure to return specific answers, it is necessary to consider whether the 'right' level of the structure is correctly identified. For this purpose, two dimensions of "relevance" have been used at INEX when evaluating XML retrieval effectiveness. The general relevance of an element (how relevant the information contained in the element is) is captured in the INEX *exhaustivity* dimension[2], whereas the *specificity* dimension indicates how focused an element is (the element does not contain non-relevant information). INEX uses these two dimensions to evaluate the effectiveness of the more complex task of retrieving structured documents, where the structure is represented through the XML mark-up.

C. J. van Rijsbergen (1989) suggested that, given the increasing complexity of a retrieval task, as it is apparent in XML retrieval, an experimental approach to information retrieval (IR) should be complemented with a theoretical evaluation. In this sense, a theoretical evaluation can be complementary to an experimental evaluation if it helps to clarify the assumptions of retrieval models and if it can identify the characteristics leading to a particular experimental behaviour.

Theoretical evaluation

In this paper, the basis of our theoretical evaluation is the logical approach to IR (Huibers, 1996). In 1971, Cooper (1971) coined the term 'logical relevance' for an objective view on relevance, in which the topical relation between document and query is considered. Van Rijsbergen and others have expressed logical relevance in terms of the implication $d \rightarrow q$, where d and q represent, respectively, the document and the query (C. J. van Rijsbergen, 1989). Chiaramella (2001) used two implications to describe structured document retrieval (XML retrieval is a special albeit most dominant application of structured document retrieval (Lalmas, 2009)): $d \rightarrow q$ modelling exhaustivity and $q \rightarrow d$ modelling specificity. Following Huibers' formalism and approach, we call topical implications between query and document 'aboutness', where aboutness is described by formally deriving the reasoning process involved in IR models.

---

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

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

As early as 1977, aboutness is discussed in (Hutchins, 1977), in the context of summarisation of texts and how manual indexers decide a document to be about a topic. Already then, aboutness is linked to topically and therefore an objective view on the content of documents. Bruza, Song, and Wong (2000) contribute an analysis of a common form of aboutness used in information retrieval and discuss how specific properties of aboutness can have an impact on experimental performance. Hjørland (2001), on the other hand, emphasises that a common-sense approach might not be sufficient and should be amended with an investigation into the epistemological view of aboutness, which takes into consideration the different views groups of people see as making a query to be about a document.

Our work is less concerned about how users of information retrieval system derive aboutness but more with the question how these systems themselves decide aboutness. In this sense, it is related to the research in (Greisdorf & O'Connor, 2003), which asks what are the underlying topical characteristics that link the output of an information retrieval system with a user's information need, or, as Greisdorf and O'Connor (2003) put it, how it is deciced that the output of an IR system is 'on topic'. We also use 'aboutness' to describe the topical relation between a query a document, as an IR system sees it, and evaluate subsequently the performance of the system in a theoretical evaluation.

We follow the approach in (C. J. van Rijsbergen & Lalmas, 1996) who have used 'aboutness' to theoretically evaluate IR models, as have Bruza and Huibers (1994), and Wong, Song, Bruza, and Cheng (2001). They have all shown that an aboutness-based theoretical evaluation can be a powerful tool to evaluate flat document retrieval models. This paper extends the existing aboutness approaches by concentrating on XML retrieval models and the assumptions made in some successful XML retrieval models to decide aboutness using a combination of structure and content. In (Blanke & Lalmas, 2011), we have already applied an 'aboutness' approach to analyse how XML retrieval models attempt to deliver only highly specific results. In this paper, we concentrate on explaining general performance results, i.e. what makes a document component 'about' a query. The particular emphasis is the influence of the structure on the aboutness decision.

More recent studies using a theoretical evaluation approach are the ones by Hui Fang, Cheng Xiang Zhai and Tao Tao (Fang, Tao, & Zhai, 2004) (Fang & Zhai, 2005). They present a formal study and a universal framework for the analysis of IR models, using a set of basic desirable constraints that any reasonable retrieval function should satisfy for good retrieval performance. At first sight, their approach looks similar to ours. However, they do not rely on a logical framework with their constraints but rely on the generalisation of intuitions mainly related to TF-IDF (term frequency and inverse document frequency) measures. These include the formalisation of a sensible interaction between TF-IDF: if given a fixed number of occurrences of query terms, a document that has more occurrences of discriminative terms (higher IDF) should achieve a better ranking. Such questions are interesting to anybody working on a new IR model.

Their approach has advantages towards ours. Ours is more abstract and high-level, as we will see. As Fang and Zhai (2005) do not employ a high level of abstraction such as aboutness but remain within the parameters of standard measures to improve retrieval directly, immediate recommendations for further improvements of models can be made. This advantage is, however, also a disadvantage when it comes to the analysis of different and new retrieval tasks such as XML retrieval. Here, we do not yet have commonly agreed foundations. TF-IDF, for instance, is subject to discussions in XML retrieval, as, for example, it does not deal well with the problem of overlap in XML retrieval (Lalmas, 2009).

We remain more abstract and therefore use an aboutness-based approach. Furthermore, we use concepts from logic such as monotonic behaviour. To us, on a very abstract level, a relevance score is a function with various variables that include often terms and their frequency values, but also other parameters. We suggest to study aboutness rules and monotonicity and how they behave with respect to these variables. These will be the basis of our theoretical evaluation framework for XML retrieval.
"

Theoretical evaluation framework

Our theoretical evaluation methodology consists of three components, which are presented in the following sections. This section introduces the first component, a *formalism* with a translation to express aboutness symbolically, and the second component, which specifies aboutness by deriving *rules of reasoning behaviour*. The third section presents the third component, specific to XML retrieval, the pure type XML retrieval model to capture the influence of the XML structure on aboutness. In the forth section, all these components of the theoretical evaluation are applied to XML retrieval models, which were successful in the INEX evaluation. The fifth section, finally, uses the results of the theoretical evaluation to explain experimental results in INEX. The paper ends with our conclusions and insights on theoretical evaluation both in the context of XML retrieval and other areas in information retrieval.

In this section, we systematically introduce the first two theoretical evaluation components, starting with the basic formalism in the next section.

Basic Formalism

Any theoretical evaluation methodology requires formalisms, complete enough to characterize the fundamental properties of retrieval models and complete enough to study their properties. Following (Bruza & Huibers, 1994), we use Situation Theory, developed by Barwise and Perry (1983), for this purpose. Situation Theory is a mathematical theory of meaning and information with *situations* as primitives (Huibers, 1996). Situations are partial descriptions of the world and are composed of *infons*. In the context of IR, queries and documents are modelled as situations,

while infons represent information items such as keywords or phrases.

The aboutness relation between two situations is represented with the symbol $\square\leadsto$, using the same symbol as (Huibers, 1996). If we consider documents and queries to be situations, then $D \square\leadsto Q$ means that the information in the document represented by situation D is about the information need expressed in the query represented by situation Q. In standard IR models, a document containing 'garden' and 'house' would be about a query asking for 'garden'. Query and document would share the term 'garden', and most IR models consider a document to be relevant to a query if they overlap in index terms. However, there might also be IR models that are not based on pure information overlap between query and document and would not consider D to be about Q in that case. A model could, for instance, limit aboutness to those cases, where there is a significant overlap in terms.

We use this basic formalism to define the translation of a model as the first step in our theoretical evaluation.

Translation

Translation is the symbolic representation of the retrieval model's handling of information. It is described by a function *map* that 'maps' information to its formal representation using the results of the indexing process. Infons are represented by $\langle\langle k \rangle\rangle$, where k stands for any indexed term or other information descriptor. A set of infons is a situation: $\{\langle\langle k_1 \rangle\rangle, \langle\langle k_2 \rangle\rangle\}$. N-ary relationships R between infons i_j are themselves infons and are modelled by $\langle\langle R, i_1, ..., i_n \rangle\rangle$. A simple example without relations is $\{\langle\langle house \rangle\rangle, \langle\langle garden \rangle\rangle\}$. Later on, we cover a more complex example. Translation is closely linked to building a document information representation through indexing.

According to (C. J. v. van Rijsbergen, 2004), with aboutness we come from the concrete notion that descriptors in indexes represent properties of documents. The simple example above demonstrates the translation of a simple document representation as a bag of terms. For XML retrieval, we need to add structural relationships to such simple representations in order to capture the XML structure. We define a translation to capture these later on.

Throughout the paper, we use upper case letters from the middle of the alphabet such as S, T for situations. Anything these situations represent like keywords but later on also structured information is symbolized with letters from the beginning of the alphabet like A or B.

Operators

In addition to the definition of aboutness, we need operators that we can use to relate situations with one another. For instance, situations can be merged to form new situations. With $\otimes$, we formalise the composition of situations, e.g. $\{\langle\langle house \rangle\rangle\}$ and $\{\langle\langle garden \rangle\rangle\}$ can be combined to $\{\langle\langle house \rangle\rangle, \langle\langle garden \rangle\rangle\}$. Preclusion, symbolised by $\perp$, expresses that information in situations clashes. They cannot be meaningfully combined such as

$\{\langle\langle flying \ birds \rangle\rangle, \langle\langle penguins \rangle\rangle\}$. If defined at all, preclusion describes mostly semantic relationships (Wong et al., 2001). However, most models have no notion of information clashes. $\equiv$ states that two situations are equivalent, i.e. they contain the same information. According to the equivalence relationship, situations should be compared according to the meaning they bear not the names we give them. Lastly, containment $\rightarrow$ describes that a situation contains at least the same information another one has. In Boolean retrieval this corresponds to, e.g., the implication that for any valid expression $x \wedge y$, x is also valid.

Rules

Aboutness is defined as a relationship between situations. In a theoretical evaluation framework, rules are used to define the reasoning aspects of this relationship. Rules are the logical representation of how a system decides a document to be about a query. The rules do not hold for all aboutness decisions but only for particular ones. Thus, an aboutness decision can be specified by the reasoning rules it incorporates. The aboutness decision can be further qualified by analysing how these reasoning rules are implemented by it: fully, conditionally or not at all. If the rules are fully supported they hold without conditions. If they are conditionally supported, they are generally supported, except for some special cases. For instance, aboutness decisions often do not consider any kind of overlap in a document's information with an information need to be sufficient to decide aboutness but only an information overlap of a particular size. As not all rules hold for all systems and in the same way, we can use them to deliver commonalities and differences between systems. By comparing the rules each system incorporates and the way it does so, we are able to give an overall comparison of the behaviour of retrieval systems.

We use a subset of rules given by (Huibers, 1996) and by (Wong et al., 2001) to describe aboutness proof systems. We use those rules that in our experience best describe system performance in XML retrieval[3]. These are generally based on the non-monotonic reasoning rules developed in (Kraus, Lehmann, & Magidor, 1990). For a complete analysis of the rules, please compare (Blanke, 2012), where we also motivate our choice of rules based on their effectiveness to explain experimental performance.

In this paper, we concentrate for space reasons on the following rules:

• Reflexivity states that a situation S is about itself: $S \square\leadsto S$.

• Symmetry states that if situation S is about another situation T ($S \square\leadsto T$), then also $T \square\leadsto S$.

• Transitivity makes a transitive conclusion: If $S \square\leadsto T$ and $T \square\leadsto U$, then also $S \square\leadsto U$.

• Left Monotonic Union (LMU) states that if $S \square\leadsto T$, then also $S \otimes U \square\leadsto T$. LMU has a variant called Mix, which

---

[3] There is an ongoing debate on which of the rules to use to best functionally describe aboutness systems. For a comparison see (Wong et al., 2001) and (Huibers, 1996).

adds a second assumption. With Mix, we state that if $S \mathbin{\square\!\!\rightsquigarrow} U$ and $T \mathbin{\square\!\!\rightsquigarrow} U$, then also $S \otimes T \mathbin{\square\!\!\rightsquigarrow} U$.

• Right Monotonic Union (RMU) states that if $S \mathbin{\square\!\!\rightsquigarrow} T$, then also $S \mathbin{\square\!\!\rightsquigarrow} T \otimes U$.

Let us discuss LMU in more detail. Huibers (1996) and Wong et al. (2001) both see the careful consideration of monotonicity as an important feature of IR. In an aboutness model unconditionally supporting LMU, a query containing 'house' is not only about documents with 'house', but equally valid answers are components with 'house' and 'garden'. This means, aboutness decisions supporting LMU are not affected by document length. For systems supporting RMU, query expansion does not change the aboutness decision. This means that systems with RMU can expand the original query and gain a higher recall base while at the same time not losing what the original query was about. Both document component and query length are decisive aspects of aboutness decisions, which underlines the importance of monotonicity (Wong et al., 2001).

The translation and the aboutness rules were developed by Huibers (1996) as parts of the theoretical evaluation of any retrieval model. In XML retrieval, we are in addition interested in how much a retrieval model uses structure to support its aboutness decision. Theoretically, we measure this by determining the qualitative reasoning distance of the XML retrieval model to its 'flat' retrieval model equivalent (if there is one) and what we call pure type XML retrieval, which we introduce in the next section.

Pure type XML Retrieval

'Pure types' have been developed by the sociologist Max Weber (see (Weber, 1997 (1903-1917)))) and have proven to be useful tools for comparative studies. A pure type describes aspects of phenomena, but is not meant to describe perfect things nor all aspects of any one particular case. It is rather a purposeful emphasis of aspects common to most cases of the observed item. In our case, the emphasis is on the impact of XML structure on the aboutness behaviour. For each model we compare its reasoning behaviour with those of other models and look at its consideration of structure by determining its qualitative reasoning distance to the pure type model and the 'flat' document model, the XML retrieval model it is based upon. The qualitative reasoning distance is described by the differences and commonalities in reasoning properties. The latter part has not been considered yet in aboutness investigations and is particularly useful for the theoretical evaluation of XML retrieval.

As the pure type XML retrieval model is an aboutness system in itself, we now go though the steps of an aboutness based theoretical evaluation, as presented in the theoretical framework section, namely, aboutness, translations, operators and rules.

Pure type aboutness

The definition of aboutness for pure type XML retrieval is directly linked to the INEX view on exhaustivity and specificity, as developed in detail in (Blanke, 2012). As seen in the introduction, exhaustivity describes how far the document component contains all the information in a query, while specificity describes how little it is about other information than the one in the query. We use this description to define pure type aboutness: Say, that a document d is indexed in such a way that its XML structure is preserved. Then, it is about a query q according to the INEX view if structure and content information of q are contained in the structure and content of d. We represent this hierarchical inclusion with $\trianglelefteq$.

To define $\trianglelefteq$, we use structure and content at the same time. Thus, to make this definition work for XML retrieval, we have to give a symbolic representation of situations that includes structure and content. We introduce this representation later on and give a Situation Theory formalism that allows us to represent indexing that preserves XML structure. But first we give two examples that illustrate the definition of hierarchical inclusion. In the following example, a section containing a paragraph about house and garden is generally a relevant answer to a query asking for a paragraph on house:

Example $\langle paragraph \rangle$house$\langle /paragraph \rangle$ $\trianglelefteq$ $\langle section \rangle$ $\langle paragraph \rangle$house, garden$\langle /paragraph \rangle \langle /section \rangle$

In the example we use the standard notation for XML elements using tags (Lalmas, 2009) for element types. The paragraph about house and garden is, however, not a particularly specific answer to the same query as it contains additional unwanted information about garden. With hierarchical inclusion, a paragraph just containing house would be a fully specific answer to a query asking for a paragraph on house, as in the following example:

Example $\langle paragraph \rangle$house$\langle /paragraph \rangle$ $\trianglelefteq$ $\langle section \rangle$ $\langle paragraph \rangle$house$\langle /paragraph \rangle \langle /section \rangle$

The two examples are both examples of hierarchical inclusion, as the information on the right hand side (document component) of the $\trianglelefteq$ operator contains the information on the left hand side (query). The examples also illustrate how for hierarchical inclusion, structure is an equal component of aboutness. With pure type XML retrieval, we can compare the impact the XML structure has on the reasoning of different retrieval models. As discussed earlier, most XML retrieval models are extensions of models that work with flat and unstructured documents, for which content alone is important in their reasoning. With the pure type model, we create a reference model, for which XML structure is fully included in aboutness.

In the next section we introduce a notation that allows us to use the power of set theory to perform reasoning that includes structure and content. This notation substitutes the direct XML representation of hierarchical inclusion using $\trianglelefteq$ as seen in the two examples above and allows for simpler derivations of reasoning.

Pure type translation

For the pure type translation we make the assumption that in pure type XML retrieval the XML structure is preserved

during indexing. We want to express the translation of a model that uses XML structure in its matching. To define the translation, we reuse the conceptual graph translation, as defined in (Huibers, Ounis, & Chevallet, 1996). Their conceptual graphs map well onto XML trees.

The conceptual graph model has been developed in (Sowa, 1992) and analysed from an aboutness point of view by Huibers et al. (1996). In the model, a query q and a document d are both seen as conceptual graphs. The knowledge in the document collection is modelled as conceptual relationships between concept types. Content is found as referents to concept types. For XML, we consider these concept types to be element types, while we limit the set of relationships to the parent and the attribute relationship. Content is found in XML as part of element types. Based on these considerations, we define pure type map next.

Definition of Map. In this section, we define a translation that preserves the XML structure. As in (Huibers et al., 1996), we want to represent a graph (in our case an XML tree) as a set of infons in order to preserve the XML structure. Intuitively, we can easily map a hierarchical organisation of information (in an XML tree) to sets, if we consider the parent elements to be the supersets of all sets of information that its children contain. We then also need a way of representing the relationship between parents and children in the same framework. Fortunately for us, we are considering sets of infons, which can either express content or relationships between content in the same formalism.

Huibers et al. (1996) develop the approach we are mapping on to XML retrieval and state that a conceptual graph carries information and can be seen as a situation. We say that an XML element carries information and can be seen as a situation. For (Huibers et al., 1996), the conceptual graph situation is constituted of the concepts, referents and relations that define the information the conceptual graph carries. We need to consider instead element types, content in element types and parent and attribute relations. As Huibers et al. (1996) propose to translate each item of a conceptual graph (concept, referent and relation) into a specific infon, we propose to do the same with XML elements. Using element type, content and relation infons, we next define map for pure type XML retrieval.

An XML tree consists of XML *elements* that have *element types* and associated content, which we refer to as *values*. These elements are connected with edges. We now suggest to translate XML elements together with their values into set of infons and to distinguish relational, content and element types.

Let us assume that d is an XML document. Then, it can be translated into situations by using a *map*:

• For each XML element p of d with element type U, map has a result $\{\langle\langle ElementType, U, p\rangle\rangle\}$, where p is the unique parameter. Such an infon is called an element type infon.

• For each XML element p of d with a type U containing descriptors k_1 to k_n, map is $\{map(U) \otimes \langle\langle Value, k_1, p\rangle\rangle, ..., \langle\langle Value, k_n, p\rangle\rangle\}$. p is the unique param-

eter that identifies U. $k_1 ... k_n$ is the set of n descriptors (for instance, index terms) that are values of the element type U. We call these infons *value infons*. $\otimes$ is explained later on.

• Say R is a relation between two XML elements A and B in d. Let E_1 and E_2 be element types and $\{\langle\langle ElementType, E_1, p\rangle\rangle\} \in map(A)$ and $\{\langle\langle ElementType, E_2, q\rangle\rangle\} \in map(B)$. We can then say that $map(R(AB)) = map(A) \otimes \{\langle\langle R, p, q\rangle\rangle\} \otimes map(B)$. p is an identifier for E_1 and q for E_2. We call such an infon a relational infon, where its parameters are ordered, so that, for instance, $\{\langle\langle Parent, i_1, i_2\rangle\rangle, \langle\langle ElementType, Article, i_1\rangle\rangle, \langle\langle ElementType, Section, i_2\rangle\rangle\}$ reads as: Article is a parent of Section.

Let us consider some examples. A paragraph about 'garden' can be expressed (translated) as $\{\langle\langle ElementType, Paragraph, p\rangle\rangle, \langle\langle Value, garden, p\rangle\rangle\}$ using an identifier p. A section with two paragraphs is translated into $\{\langle\langle ElementType, Section, s\rangle\rangle, \langle\langle Parent, s, p_1\rangle\rangle, \langle\langle Parent, s, p_2\rangle\rangle, \langle\langle ElementType, Paragraph, p_1\rangle\rangle, \langle\langle Value, garden, p_1\rangle\rangle, \langle\langle ElementType, Paragraph, p_2\rangle\rangle, \langle\langle Value, house, p_2\rangle\rangle\}$. We use the relation *Parent* to express that the two paragraphs are the children of the section.

We need to make sure that each XML document corresponds to exactly on set of infons and vice versa. To this end, we sketch a translation production algorithm leaving out the definition of the infon parameters, which can be easily deduced. We traverse the XML tree in a depth-first manner. Each time we visit an XML element we create an element type infon. If we reach a leaf we collect all the descriptors in the leaf and create a value infon for each of them. We then backtrack through the tree and while backtracking we create relational infons to connect the element type infons. Following this algorithm, we create a unique XML situation (set of infons) from each XML document. Furthermore, we can re-create the tree of the XML document bottom up, starting with the value infons to create the leaves and then reconnect the elements by following the relational infons using the element type infons to define the types of the elements. We only allow XML situations (set of XML infons), which lead to a valid XML document according to the definition by the W3C and therefore conform to the rules of a Document Type Definition (DTD) or an XML Schema (XSD) (in our case given by INEX).[4]

Pure type operators

To perform our aboutness reasoning for pure type XML retrieval, we need to define operators between XML situations. We use the same operators as defined earlier, but change these to make them work for XML situations. We need these operators to perform aboutness reasoning later in this paper. Here, we define equivalence $\equiv$ and composition $\otimes$. Containment $\rightarrow$ and preclusion $\perp$ can be defined similarly.

---

[4] This can be proven by running it against the official W3C markup validation service: http://validator.w3.org/

Let us assume that we have two XML situations S and T, and a translation function *map*. Then:

- Two XML situations are equivalent if we can rename their identifiers and have an equivalent set, e.g.: $\langle\langle ElementType, Section, i_1\rangle\rangle \equiv \langle\langle ElementType, Section, i_2\rangle\rangle$.

- $S \otimes T \equiv map(A) \cup map(B)$, where $S \equiv map(A)$, $T \equiv map(B)$ and A and B are XML documents.

With this pure type translation, structure and content of XML documents come together in the same formalism. This makes it easier for us to draw conclusions about the reasoning incorporated in a specific XML retrieval system. By using $\cup$ to form larger situations, we can now employ the power of set theory to derive the aboutness reasoning, which allows us to reuse earlier results developed in (Huibers, 1996).

Rules

Now we show the reasoning rules of pure type XML retrieval. We first introduce an auxiliary proposition that greatly simplifies the proofs needed to analyze the pure type's reasoning. The proposition is based on our definitions for translation and aboutness decision and Huibers' analysis of conceptual graphs in (Huibers, 1996):

Proposition 1 *For XML document components A and B, $B \trianglelefteq A$ if and if only (or iff) $map(A) \supseteq map(B)$.*

Proof $\Rightarrow$: Let $B \trianglelefteq A$. Then, we know that B has the content and the structure of a subdocument of A. Using the algorithm from page 5, we construct two situations $S \equiv map(A)$ and $T \equiv map(B)$. This means all relational, element types and value infons from T are also in S. Thus, $map(A) \supseteq map(B)$ using the parameter renaming defined in (Huibers et al., 1996).
$\Leftarrow$: Let $map(A) \supseteq map(B)$. Then, we know that $map(B)$ has only infons also found in $map(A)$. If we apply the algorithm to transform situations into XML documents from as shown on page 5, B needs to have the structure and content of a subdocument of A and is therefore hierarchically included in it: $B \trianglelefteq A$. $\square$

The aim of our translation was to use the power of set theory in the aboutness proof. Proposition 1 verifies that we can do aboutness proofs with a relatively simple set operation on the representation (as situations) of XML trees.

Using Proposition 1, we can now easily show that Reflexivity is given, as $map(A) \supseteq map(A)$ with $S \equiv map(A)$. Transitivity holds. If $S \Box\rightsquigarrow T$ and $T \Box\rightsquigarrow U$, then also $S \Box\rightsquigarrow U$. Say, that $S \equiv map(A)$, $T \equiv map(B)$, and $U \equiv map(C)$. Then, $map(A) \supseteq map(B)$ as well as $map(B) \supseteq map(C)$. Thus, $map(A) \supseteq map(C)$. Symmetry is not given. From $S \Box\rightsquigarrow T$, we do not derive that $T \Box\rightsquigarrow S$. Again, $S \equiv map(A)$ and $T \equiv map(B)$. Then, $map(A) \supseteq map(B)$ is not equivalent to $map(B) \supseteq map(A)$.

Now, we demonstrate that the aboutness system of pure type retrieval fully supports LMU. Given the assumption that a situation S is about another situation T ($S \Box\rightsquigarrow T$), LMU offers the conclusion that also $S \otimes U \Box\rightsquigarrow T$, where U is a situation. Let us assume, that $S \equiv map(A)$, $T \equiv map(B)$ and $S \otimes U \equiv map(C)$. The premise of LMU can then be rewritten as $B \trianglelefteq A$, which is implied by $map(A) \supseteq map(B)$ according to Proposition 1. With the definition of the translation, we also have $map(C) \supseteq map(A)$. Therefore: $map(C) \supseteq map(B)$. Thus, according to Proposition 1 also $C \trianglelefteq B$ and LMU is fully supported. Mix is a special case of LMU and thus given, too. RMU, on the other hand, does not hold. Indeed, from $S \Box\rightsquigarrow T$, we cannot conclude $S \Box\rightsquigarrow T \otimes U$.

In the next section, we use the pure type aboutness system to find out what role structure plays in actual XML retrieval models, such as those developed and evaluated during INEX.

Theoretical evaluation of XML Retrieval Models

In this section, we apply our proposed theoretical evaluation framework to evaluate XML retrieval models experimented with during INEX. We do not fully specify all the properties of these models, instead we concentrate on demonstrating important aspects of our theoretical evaluation approach that help us explain experimental behaviour. We begin with a model that builds upon existing flat document retrieval strategies — XML language model. Afterwards, we investigate a model specifically designed for XML retrieval – Gardens Point.

For these two models, we look at the corresponding translation into formal situations, the reasoning rules, and the relationship to pure type XML retrieval. In a full theoretical evaluation, we would cover over 20 rules but for the purpose of this paper, we focus on Transitivity, Reflexivity, Symmetry and Monotonicity. These often form the basis of a theoretical evaluation, as we have shown in (Blanke & Lalmas, 2007). In related work (Blanke, 2012), we could identify these rules as those that have the strongest impact on the experimental behaviour in INEX. The two models, on the other hand, have been chosen because of their behaviour in the experimental task we investigate later on. In (Blanke & Lalmas, 2011), we look at the performance of models in another INEX task, while Blanke (2012) offers the evaluation of five highly performing models in INEX (in terms of effectiveness).

Language Models

An XML retrieval model that is based on a model for flat document retrieval and that performed well at INEX is the language modelling described in (Sigurbjörnsson & Kamps, 2005). We refer to this model as the XML language modelling.

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 an element d is about a query q if and if only the information in q can be found in the element (its representation). We actually have several aboutness decisions depending on which elements are indexed, e.g. those above a given size, those that correspond

to types frequently assessed as relevant, or some types of elements only. We therefore must provide a *map* function to translate infons for each chosen approach. We only demonstrate 2 of them, as the others are built very similarly. Let A be an element with term t and type e:

- Length based approach: $map_{length}(A) \equiv \{\langle\langle ElementType, e, i\rangle\rangle, \langle\langle Value, t, i\rangle\rangle |\ |A| > \kappa\}$ where κ is a length threshold.
- Section based approach: $map_{sec}(A) \equiv \{\langle\langle ElementType, e, i\rangle\rangle, \langle\langle Value, t, i\rangle\rangle |\ e \in \{Sec\}\}$

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 $P(t_i|e) > \theta$. In (Blanke, 2012), details of the threshold θ's dependency on the query are discussed. For now, it is enough to say that θ is based on the smoothing value, which is in any language model the lowest possible value for an element without query terms. 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 (in our case, those used at INEX).

We now continue with the analysis of the reasoning rules supported by our XML language modelling. To prove the reasoning rule, we use the proposition that $P(t_i|e) > \theta$ if there is an overlap in information between document component and query. We omit the proof here. Huibers (1996) demonstrated that retrieval systems based on this proposition, generally support Reflexivity, Symmetry and Transitivity reasoning. We therefore only demonstrate LMU. Say, that $S \equiv map(A)$, $T \equiv map(B)$ and $S \otimes U \equiv map(C)$. Then, we have $C \cap B \neq \emptyset$ and $C \supseteq A$. Thus, $A \cap B \neq \emptyset$, and LMU is unconditionally supported.

Mix is a special case of Left Monotonic Union and is therefore also supported. Similarly to Left Monotonic Union, only those situations can be combined, which are part of the same index. This is interesting, as for XML retrieval parents and children are about the same queries and Mix should therefore be an automatic property, because it extends Left Monotonic Union. However, this is not the case for this model, where children and parents can be part of distinct indexes.

This language modelling approach for XML retrieval is distinctively different from our pure type XML retrieval model. Structure is included in the aboutness decision by a priori dividing elements into several different indexes and not part of the aboutness reasoning itself. We will see in the section on the experimental evaluation results how this expresses itself in the experimental evaluation results.

Gardens Point

The *Gardens Point Model* (Geva, 2005) is a model that was specifically designed to fit the requirements of XML retrieval, by discriminating aboutness for leaf and branch elements. A leaf element is considered to be about a query if it contains at least one query term. A branch element is about a query if its subtree contains at least one leaf element that is about the query. A document component d is about a query q if $rsv(D, Q) > 0$.

For leaf elements L, the scoring function is defined as $rsv_L = K^{n-1} \sum_{i=1}^{n} \frac{t_i}{f_i}$. Here, n is the number of unique query terms, t_i is the frequency of the i-th query term in the leaf element and f_i its collection frequency. Thus, rsv_L favours document components with many unique query terms while penalizing query terms frequent in the collection. The weights of the leaf elements are propagated to form the weights for the branch elements. For a branch element B, the scoring function is given by $rsv_B = D(c) \sum_{i=1}^{c} rsv_{L_i}$. c stands for the number of retrieved children elements. A decay factor $D(c)$ is used to control the propagation, where $D(c) = 0.49$ for $c = 1$ and $D(c) = 0.99$ otherwise.

In the Gardens Point model, the location of each term is identified by an absolute XPath expression (Geva, 2005). This means that the complete XML tree structure is preserved in the element representation. The translation is therefore the same as in the pure type XML retrieval model. However, hierarchical inclusion $\trianglerighteq$ is not implemented, because Proposition 1 is not given. $rsv(d, q) > 0$ does not mean $map(d) \supseteq map(q)$, as the following example demonstrates. In the Gardens Point model, a leaf element containing 'house' and 'garage' is about a query asking for 'house'. But according to Proposition 1, $\{\langle\langle ElementType, Paragraph, i_1\rangle\rangle, \langle\langle Value, House, i_1\rangle\rangle, \langle\langle Value, Garage, i_1\rangle\rangle\}$ is not about $\{\langle\langle ElementType, Paragraph, i_1\rangle\rangle, \langle\langle Value, House, i_1\rangle\rangle\}$, as it has additional information about garages.

According to (Geva, 2005), each term in an XML document is identified by three elements in the index: file path, absolute XPath context and term position within the XPath context. As a query language, however, Gardens Point uses NEXI (Geva, 2005), which is an (enhanced) subset of XPath. Content-only (Geva, 2005) queries are expressed as a search over the entire article element using NEXI. This leads to asymmetric reasoning behaviour. Say, we have a query $//article[about(house)]$ that is about an element $article[1]/bm[1]/bib[1]/bibl[1]/bb[13]/pp[1]$. Then, we cannot use $article[1]/bm[1]/bib[1]/bibl[1]/bb[13]/pp[1]$ as a query again, as it is not a valid query expression. Thus, Gardens Point does not support the Symmetry rule.

LMU would be a property of the aboutness systems, if with $S \mathrel{\square\!\rightsquigarrow} T$ we could derive that $S \otimes U \mathrel{\square\!\rightsquigarrow} T$. Let us assume that $S \equiv map(A)$, $T \equiv map(B)$ and $S \otimes U \equiv map(C)$. Thus, $rsv(A, B) > 0$. We are able to then also say that $rsv(A, C) > 0$. The sum in the calculation for leaf elements stay at least the same when adding new information. Sums in relevance calculation generally promote monotonic behaviour (Blanke & Lalmas, 2006). RMU on the other hand is not given, which again shows how close Gardens Point is to pure type XML retrieval.

Elsewhere (Blanke, 2012), we have done a complete comparison of the reasoning properties with the pure type model and a 'flat' document model derived from an aboutness decision purely based on rsv_L. This comparison reveals that the model holds for almost the same set of rules as the pure type

model does. In our theoretical evaluation of XML retrieval models (limited to two models in this paper), we could see how XML retrieval models are concerned with controlling monotonic behaviour as well as other reasoning that support good performance in XML retrieval. As an example, we have discussed Symmetry.

GPX does also not support right monotonic reasoning (RMU). RMU does not necessarily support better retrieval results. RMU allows us to conclude from the assumption $S \mathbin{\square\!\!\rightsquigarrow} T$ that also $S \mathbin{\square\!\!\rightsquigarrow} T \otimes U$. However, in XML retrieval T might well include a structural condition. For instance, T alone might point to a section while $T \otimes U$ might point to a paragraph within a section, which would completely change the aboutness relation. It is this kind of desirable behaviour that implies that XML retrieval systems should be able to change an aboutness decision if the XML context changes. The description of the monotonic reasoning behavior of XML retrieval models is key to the distinction with flat document retrieval.

GPX compared to XML language modelling is closer to pure type XML retrieval especially in terms of the monotonic reasoning behaviour. This is one reason for its convincing behaviour in the experimental evaluation and why it outperforms XML language modelling in many experimental evaluation conditions.

Theoretical Analysis of the INEX Experimental Evaluation Results

In this section, we show how our proposed evaluation framework for XML retrieval is complementary to experimental results obtained at INEX, on the so-called thorough retrieval task for content-only (CO) queries (Gövert et al., 2006). We have concentrated on the INEX 2005 campaign because since then the fundamentals of the models for that task have not changed much. The thorough task is concerned with retrieving relevant elements for a given query, and to rank them according to their estimated relevance to that query. The thorough task is to be contrasted with the focused task, which is concerned with returning a list of non-overlapping elements as answers to a given 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. In (Blanke & Lalmas, 2011), we presented a theoretical framework, also based on Situation Theory, to evaluate the focused task. The framework included a new theoretical evaluation methodology to evaluate filters, commonly used in INEX to focus the retrieval results on only the most specific XML elements. In this paper, we are interested in the underlying task to retrieve relevant XML elements in general.

We proceed as follows. In the following parts, we introduce first some of the background of the effectiveness measures used to evaluate the thorough task. We analyse which of the aboutness reasoning properties promise to deliver good results under these particular measures. Afterwards, we use this analysis to explain experimental outcomes for the two models we have theoretically evaluated.

Aboutness reasoning behind the INEX metrics

We concentrate on the effort-precision/ gain-recall ($ep - gr$) metrics used to evaluate the thorough task. Effort-precision (Kazai & Lalmas, 2006) is based on the amount of relative effort that a user has to make following a ranking of a system compared to the effort an ideal ranking would take. Though other measures are used by now in INEX, these measures were used during INEX 2005, and our theoretical framework is used to explain experimental results for that same year.

Effort-precision ep is defined in (Kazai & Lalmas, 2006) by $ep[r] = \frac{i_{ideal}}{i_{run}}$. i_{ideal} is measured as the rank position at which the cumulated gain of r is reached by the ideal curve of the ranking. i_{run} is measured by the same rank position in the real run. Let us assume that that three returned elements i_1, i_2 and i_3 have relevance scores of 2, 1, and 0, respectively. In an ideal system we would have the following ranking: $\{i_1, i_2, i_3\}$. Assume that our system however returns $\{i_3, i_1, i_2\}$. The cumulated gain of 2 would be reached by the ideal system at rank 1, while our system delivers it only at rank 3. This means $ep[r] = 1/3$. This demonstrates that $ep[r] = 1$ indicates a perfect performance and that scores are always between 0 and 1, as the ideal cumulated gain is always higher than or equal to the actual cumulated gain.

In 2005, ep is captured at gain-recall points gr (Kazai & Lalmas, 2006), where gain-recall is calculated as the cumulated gain value divided by the total achievable cumulated gain. $ep - gr$ therefore measures where the most relevant XML elements are found in the ranking produced by a system. The more they are concentrated in the tail of the result list, the worse the performance. Secondly, $ep - gr$ measures how completely the set of most relevant elements is represented at the top of the ranking. A small number of highly relevant elements that appear at the tail of the ranking have a much stronger impact on the performance than, for instance, highly irrelevant elements at the head of the ranking list. This is the case as the denominator, which dominates $ep[r]$, is the actual ranking result. If the actual ranking is much worse than the ideal ranking, $ep[r]$ becomes very small. For a further discussion on the relationship between the cumulative gain measure and other standard information retrieval measure, readers are referred to (Carterette & Voorhees, 2011) and (McSherry & Najork, 2008).

To understand how systems can develop a ranking that has at its top only the most relevant elements, we need to look at how systems can preserve aboutness. Our assumption is that those elements that are about a query are related in the information they have. Then, related relevant XML elements differ in how much relevant information they contain. According to our reasoning rules, a relevant element that has at least the same relevant information or more than another element can be found by either applying Left Monotonic Union or Mix reasoning. As defined in the section on the aboutness rules, LMU concludes that $S \otimes U \mathbin{\square\!\!\rightsquigarrow} T$, given $S \mathbin{\square\!\!\rightsquigarrow} T$. With respect to Mix, we can from the assumptions $S \mathbin{\square\!\!\rightsquigarrow} T$ and

Table 1
INEX thorough task evaluated using $ep - gr$.

Rank	Model	Run	MAep
4	Language Model	LMQrelbasedIndex	0.0829
5	Language Model	LMLengthbasedIndex	0.0802
6	Language Model	LMElementIndex	0.0793
7	Gardens Point	GPX-1-Thorough	0.0757
14	Gardens Point	GPX-2-Thorough	0.0706

$U \ \Box \rightsquigarrow \ T$ conclude that also $S \otimes U \ \Box \rightsquigarrow \ T$.

Looking at LMU, we already know from our discussions from the rules section, that the added information U does not necessarily have to be about the query. Systems, which support LMU, have therefore problems returning only highly relevant elements early in the ranking. Mix, however, provides a more conservative approach to monotonicity. Contrary to LMU, the added information is also about query. Therefore, Mix supports better performance in $ep - gr$, while with LMU we need to be more careful.

For all XML retrieval results under $ep - gr$, we next look at how these left monotonic reasoning properties explain some of the experimental outcomes.

Theoretical Analysis of Experimental Results

Table 1 presents the results for the models discussed in this paper evaluated at INEX 2005 with $ep - gr$. The rank column is the overall ranking of the model, named in the second column. The run is one of the submissions of the model, which we will describe later on. MAep stands for Mean Average effort precision, which is calculated by averaging the effort-precision values whenever a relevant XML element is found in the ranking (Fuhr, Lalmas, Malik, & Kazai, 2006). We can clearly see how the analysed models perform for these particular tasks. Generally speaking, both models perform well. Next, we explain these good performances and differences in the different runs submitted to INEX for these models. We rely on the left-monotonic reasoning rules, we have introduced.

With respect to $ep - gr$, XML language modelling (Sigurbjörnsson & Kamps, 2005) looks at reducing the number of indexing units with two special indexes only: one based on element length (run referred to as *UAmsCOT LengthbasedIndex* in Table 1) and another based on past relevance assessments Q_{rel} (run referred to as *UAmsCOT QrelbasedIndex* in Table 1). These two indexes perform particularly well and come 4th and 5th in the overall assessment ranking (see Table 1).

LMU is fully supported by the language modelling approach. This means that the aboutness decision is preserved across elements that share the same relevant information with smaller elements but also contain other information. This good performance through LMU reasoning is further fostered by the fact that Mix is fully supported, too.

Looking at LMU, it is also no surprise that Q_{rel} is the model's best submitted run with some distance (see Table 1). Here, the side effect of unwanted information added by LMU reasoning occurs less likely, as Q_{rel} contains only those elements that according to the experience of previous INEX years are more likely than others to be regarded as relevant.

The overall performance of Gardens Point is relatively worse for the thorough task (Table 1) than for others in INEX 2005 (Kazai & Lalmas, 2006). Though often the best-performing model in INEX 2005, here it is outperformed by several other retrieval systems, although its overall performance is still very good (it still ranked among the 10 top models). According to our analysis of Gardens Point, the model differs among other things from language modelling and other models in its decay factor $D(c)$. We can explain the model's experimental behaviour with the impact of $D(c)$ on LMU and Mix. $D(c)$ aims to control the impact of parent elements by putting more relevance onto the relevant children. However, these parent elements might have a better score and should therefore appear at the top according to the ideal ranking. If they do not appear at the top in the run, they reduce the overall performance significantly (as observed when using the measure $ep - gr$).

Using Mix, we can demonstrate this behaviour. Let us assume that we have a component S with a score of 3 for a query T and a component U with a score of 2. Then according to Mix, with $S \ \Box \rightsquigarrow \ T$ and $U \ \Box \rightsquigarrow \ T$ also their parent $S \otimes U \ \Box \rightsquigarrow \ T$. Let us further assume that without $D(c)$ the score of $S \otimes U$ would be 5, with $D(c)$ it is 2.45, which reduces its rank in the actual run, increasing its distance from the ideal rank and therefore decreasing $ep[r] = \frac{i_{ideal}}{i_{run}}$. A similar argument can be made using LMU twice, considering a highly relevant grandchild and child of a parent. As it reduces the scores of parents with highly relevant children, it is not surprising that Gardens Point's performance decreases.

In summary, we used our theoretical evaluation results to compare the experimental ones for XML retrieval in order to find out how the adjustment of an existing flat document retrieval models compares to the creation of completely new one, especially designed to meet the requirements of XML retrieval. We analysed reasoning properties that support good performance in the thorough retrieval task under $ep - gr$ and determined reasoning properties that supported good performance for this task. The monotonic reasoning rules have played an important role here, as already seen for flat document retrieval.

Conclusions and Discussions

This paper presented a theoretical aboutness approach for an evaluation of XML retrieval models. To this end, we amended existing approaches and expanded the aboutness system to demonstrate the reasoning properties and under which conditions these properties are supported. Each of our theoretical evaluations goes through the same steps to define the characteristics of a particular XML retrieval model: translation and aboutness definition, analysis of rules and comparison with pure type XML retrieval. We proposed the

so-called pure type XML retrieval, which we use to compare the behaviour of two XML retrieval models presented at INEX. We were able to show commonalities as well as differences between these models. We also used this theoretical evaluation to explain experimental differences between these XML retrieval models as evaluated in INEX.

Our theoretical evaluation is based on the logical analysis of reasoning processes involved in XML retrieval models. We build upon an advanced mathematisation of natural language semantics called Situation Theory. Logic-based evaluation helps discover underlying assumptions of IR performance, which are often hidden in the functional behaviour of IR models by, for instance, tuning a priori assigned parameters in such a way that they fit best the evaluation task. For example, we saw that the $D(c)$ factor in the GPX model can have undesired side effects. Such transparency leads to new insights about the behaviour of models in general and not only for particular retrieval tasks. This can also help develop new models. We have seen, for instance, that we should be carefully considering the degree to which we allow monotonic behaviour while developing or adapting existing retrieval models. In (Blanke & Lalmas, 2011), we have demonstrated how this type of approaches, and in particular, and again, the role of monotonicity, can be useful for analysing specific approaches to INEX XML retrieval task (more precisely, the focused INEX task, looking at what makes a document component focusing on the query). Here, we have presented the more general framework (what makes a document being about a query) and the analysis of two XML retrieval models.

These were many advantages of a logic-based theoretical evaluation. However, there are also drawbacks, which might prevent researchers from engaging with this type of evaluation approach. Much of XML retrieval work is currently done by adjusting weights to meet the different requirements associated with the task being evaluated. It has been noted (Wong et al., 2001) that the proposed theoretical evaluation formalisms, as promoted in this paper, often deliver too high an abstraction to cover specific cases. However, we have tried to address such problems by introducing some mathematics into the Situation Theory framework delivered by Huibers and others. To this end, for example, the notion of conditionally supported reasoning properties was added.

Further research remains to be done into possible frameworks of theoretical evaluations. It might turn out that logic-based frameworks fall behind frameworks based, e.g., on retrieval heuristics, which have been successfully applied in IR. For instance, Fang and Zhai (2005) and Fang et al. (2004) have shown that such retrieval heuristics can lead to significant performance improvements for IR models. As XML retrieval is a relatively young discipline compared to traditional IR, such heuristics, however, have not been established yet. This might change in the future, and is something we are looking into.

Another advantage of our theoretical evaluation approach is that it can help an emerging field before it is mature or large enough to develop its own evaluation strategies that reflect its specific requirements. The presented methodology can help in the early stages to structure design approaches and develop evaluation strategies for such emerging field. Of particular interest to us is to develop new information retrieval strategies for the emerging web of things, i.e. a web where devices and objects are directly interlinked. Since the web of things relies on a graph-based data model using the W3C standard RDF (Heath & Bizer, 2011), our approach, which combines structure and content, could be useful and easily adopted.

As a matter of fact, there has been, to date, little work on evaluating retrieval from the web of data (but see (Pound, Mika, & Zaragoza, 2010) for more information about this). There are many other emerging fields in information retrieval, which use evidence from a network of information to enhance the retrieval process. These include opinion mining or expert systems, which both use networks of related information (reviews, expert assessments, etc.), to return relevant results (Pang & Lee, 2008). Furthermore, beyond structural relationships new relationships within content such as time relationships in blog retrieval could be analysed using more advanced reasoning rules (Zhang, Yu, & Meng, 2007). In these emerging fields, our approach could help with design decisions for new models but also help to understand how traditional information retrieval techniques could be reused for these new fields and associated approaches.

Acknowledgement

References

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Blanke, T. (2012). *Phd thesis — using situation theory to evaluate xml retrieval.* IOS Press.

Blanke, T., & Lalmas, M. (2006). Theoretical benchmarks of XML retrieval. In *ACM SIGIR '06* (pp. 613–614). New York, NY, USA: ACM Press.

Blanke, T., & Lalmas, M. (2007). Theoretical evaluation of XML retrieval evaluation. In *Dir '07*.

Blanke, T., & Lalmas, M. (2011). Specificity aboutness in XML retrieval. *Information Retrieval*, *14*(1), 68-88.

Bruza, P. D., & Huibers, T. W. C. (1994). Investigating aboutness axioms using information fields. In *ACM SIGIR '94* (pp. 112–121). New York, NY, USA: Springer-Verlag New York, Inc.

Bruza, P. D., Song, D. W., & Wong, K. F. (2000, October). Aboutness from a commonsense perspective. *J. Am. Soc. Inf. Sci.*, *51*, 1090–1105.

Carterette, B., & Voorhees, E. M. (2011). Overview of information retrieval evaluation. In M. Lupu, K. Mayer, J. Tait, A. J. Trippe, & W. B. Croft (Eds.), *Current challenges in patent information retrieval* (Vol. 29, p. 69-85). Springer Berlin Heidelberg.

Chiaramella, Y. (2001). Information retrieval and structured documents. In *Lectures on information retrieval* (p. 286-309). New York: Springer-Verlag.

Cooper, W. (1971). A definition of relevance for information retrieval. *Information Storage and Retrieval*, *7*, 19-37.

Fang, H., Tao, T., & Zhai, C. (2004). A formal study of information retrieval heuristics. In *Sigir '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 49–56). New York, NY, USA.

Fang, H., & Zhai, C. (2005). An exploration of axiomatic approaches to information retrieval. In *Sigir '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 480–487). New York, NY, USA.

Fuhr, N., Lalmas, M., Malik, S., & Kazai, G. (Eds.). (2006). *Advances in XML information retrieval and evaluation, INEX 2005, dagstuhl* (Vol. 3977). Springer.

Geva, S. (2005). GPX - gardens point XML IR at INEX 2005. In N. Fuhr, M. Lalmas, S. Malik, & G. Kazai (Eds.), *INEX* (Vol. 3977, p. 240-253). Springer.

Gövert, N., Kazai, G., Fuhr, N., & Lalmas, M. (2006). Evaluating the effectiveness of content-oriented XML retrieval. *Journal of Information Retrieval*, *9*(6), 699-722.

Greisdorf, H., & O'Connor, B. C. (2003). Nodes of topicality: Modeling user notions of on topic documents. *JASIST*, *54*(14), 1296-1304.

Heath, T., & Bizer, C. (2011). *Linked data: Evolving the web into a global data space* (1st ed.). Morgan & Claypool.

Hjørland, B. (2001, July). Towards a theory of aboutness, subject, topicality, the domain, field, content and relevance. *J. Am. Soc. Inf. Sci.*, *52*, 775–.

Huibers, T. W. (1996). *An axiomatic theory for information retrieval*. Utrecht: Universiteit Utrecht.

Huibers, T. W., Ounis, I., & Chevallet, J.-P. (1996). Conceptual graph aboutness. In *Proceedings of the 4th international conference on conceptual structures: Knowledge representation as interlingua* (pp. 130–144). London, UK.

Hutchins, W. J. (1977). On the problem of 'aboutness' in document analysis. *Journal of Informatics*, *1*(1), 17–35.

Kazai, G., & Lalmas, M. (2006). INEX 2005 evaluation measures. In *Advances in XML information retrieval and evaluation*. Springer Berlin/Heidelberg.

Kraus, S., Lehmann, D., & Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, *44*(1-2), 167–207.

Lalmas, M. (2009). XML retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *1*(1), 1-111.

McSherry, F., & Najork, M. (2008). Computing information retrieval performance measures efficiently in the presence of tied scores. In *Advances in information retrieval* (p. 414-421).

Pang, B., & Lee, L. (2008, January). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, *2*, 1–135.

Pound, J., Mika, P., & Zaragoza, H. (2010). Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on world wide web* (pp. 771–780). New York, NY, USA: ACM.

Rijsbergen, C. J. van. (1989). Towards an information logic. In *ACM SIGIR '89* (pp. 77–86). New York, NY, USA: ACM.

Rijsbergen, C. J. van, & Lalmas, M. (1996). Information calculus for information retrieval. *J. Am. Soc. Inf. Sci.*, *47*(5), 385–398.

Rijsbergen, C. J. v. van. (2004). *The geometry of information retrieval*. Cambridge University Press.

Sigurbjörnsson, B., & Kamps, J. (2005). The effect of structured queries and selective indexing on XML retrieval. In N. Fuhr, M. Lalmas, S. Malik, & G. Kazai (Eds.), *INEX* (Vol. 3977, p. 104-118). Springer.

Sowa, J. F. (1992). Conceptual graphs. *Knowl.-Based Syst.*, *5*(3), 171-172.

Weber, M. (1997 (1903-1917)). *The methodology of the social sciences*. New York, NY, USA: Free Press.

Wong, K.-F., Song, D., Bruza, P., & Cheng, C.-H. (2001). Application of aboutness to functional benchmarking in information retrieval. *ACM Trans. Inf. Syst.*, *19*(4), 337-370.

Zhang, W., Yu, C., & Meng, W. (2007). Opinion retrieval from blogs. In *Proceedings of the sixteenth acm conference on conference on information and knowledge management* (pp. 831–840). New York, NY, USA: ACM.