queries. We identify the performance bottleneck, and reduce the problem to answering XML aggregation queries.

2. We study the problem of using XML cubes to answer XML aggregation queries. We examine view usability and query rewriting in our problem setting.

2a. A novel aggregation scheme per last week's discussion

3. We propose a columnar storage for non-aggregation views to facilitate aggregations and our query semantics.

2 Query Semantics & Ranking Model

In this section, we formally define query semantics and ranking model for tree pattern queries with text predicates. We start by introducing query semantics. Then we revisit IR heuristics, and evolve them for tree pattern queries to deliver IR-consistent ranking functions.

2.1 Query Semantics

We consider queries that consist of an XML tree pattern and keyword search predicates, \( Q(Q_T, K_P) \).

The XML tree pattern is expressed by an XPath fragment, namely, node tests, the parent-child axis (/), the ancestor-descendant axis (/[]), wildcards (*), predicates ([]) and value comparisons (including attribute values and text nodes). In addition, we consider a special type of predicates for keyword search: a text predicate \( k_s \) in the form of \([xpe \text{ contains } W]\), where \( xpe \) represents a path pointing to the inner text of XML elements, and \( W = \{w_1, \ldots, w_n\}\) \( s \) a set of keywords. Without loss of generality, we assume contains has AND semantics and \( xpe \) must contain all keywords in \( W \). (Again: this was not the case in Example 1.)

In the following, the notion of "tree pattern queries" only refers to conventional XPath constructs, and does not include text predicates. We use \( Q_T \) to denote the tree pattern query, \( K_P \) to denote a set of text predicates, and \( Q(Q_T, K_P) \) to denote the complete query.

Definition 2.1 (Ranking Target). The ranking targets of the query \( Q(Q_T, K_P) \) are a set of XML elements projected by the XPath expression \( Q_T \), denoted by \( R_{Q_T} = \{e_1, \ldots, e_n\}\). Each target is associated with one or more textual elements as its textual content.

Many questions:

1. Do keyword predicates apply to the projected element only?
2. How much of the path "xpe" does not "count" for \( Q_T \)? Why?