Such a heuristic, however, is not captured existing ranking schemes, by which we would normalize individual elements and cannot achieve the same effect as LNC for ranking authors.

1.2 Our Ranking Framework

The fundamental reason that causes various violations is the mismatch between query targets and keyword statistics. IR heuristics, which describe keyword statistics and their interactions, characterize desired constraints on how the targets should be ranked. In IR, query targets are always documents. Hence, document-based statistics can deliver reasonable ranking functions. In XML databases, however, the concept of “element” is not equivalent to the target; rather, the expressive power of XML query languages enables users to specify a collection of XML fragments to be the results. Ranking functions relying on element-based statistics can only regulate the ranking of elements, but not the fragments. Therefore, these functions always violate some IR heuristics when ranking the real targets.

In this paper, we present a ranking framework that treats XML fragments satisfying the tree pattern as a collection, upon which keyword statistics are collected to characterize ranking heuristics for the real targets. The ranking functions based on these statistics satisfy IR heuristics.

This ranking framework not only guarantees the consistency between ranking functions and IR heuristics, but sheds light on the underlying philosophy of the marriage of structured queries and relevance ranking: structured queries augment ranking with expressive contexts. Consider the query:

\[ Q_2: \text{paper}./\text{year} > 2004 \text{ AND } \text{.}./\text{content} \text{ contains "MapReduce hash join"} \]

The tree pattern query specifies targets as papers published after 2004, in which year MapReduce was introduced to the public. Such a personalized context affects result ranking through term discrimination: MapReduce is an extremely common term after 2004; hash join, on the other hand, is not a hot topic in this period. In other words, hash join is more discriminative in the context.

Surprisingly, none of the conventional IR ranking functions satisfy all heuristics unconditionally, as proved in [6]. Since our derived ranking functions rely on mathematical formulas of existing ranking functions, they cannot satisfy all heuristics unconditionally either. However, the ranking framework guarantees that they do not violate more heuristics than existing IR ranking functions.

*This example is simple and right-on at the same time. Perfect for intro. We should consider altering the order. Let’s discuss an alternate intro where the XML stuff is pushed later.*