keyword query with two terms \( w_1, w_2 \), and \( d_1, d_2 \) be two conventional documents. Let \( tf(w, d) \) be the number of the occurrences of \( w \) in \( d \), \( idf(w) \) be the inverse of the number of documents that contain \( w \), and \( score(Q_k, d) \) be the relevance score of \( d \) w.r.t. \( Q_k \). The constraint is described as follows:

TDC. Assume \( |d_1| = |d_2| \) and \( tf(w_1, d_1) + tf(w_2, d_1) = tf(w_1, d_2) + tf(w_2, d_2) \). If \( idf(w_1) \geq idf(w_2) \) and \( tf(w_1, d_1) \geq tf(w_1, d_2) \), then \( score(Q_k, d_1) \geq score(Q_k, d_2) \).

TDC formalizes the effect of keyword discrimination and regulates the interaction between TF and IDF in ranking documents. When we consider the tree pattern query //paper/author[./bio contains "DB, privacy"], a parallel constraint should regulate the effect of keyword discrimination in ranking authors, because the query specifies targets as a set of authors. Among the three author elements in the XML tree, one contains privacy and two contain DB. Hence, privacy is more discriminative w.r.t. authors. Let’s consider the ranking order between \( a_1 \) and \( a_2 \). Since \( te_1 \) and \( te_2 \) each match one term and have equal lengths, \( a_2 \) should have a higher ranking score than \( a_1 \).

Existing ranking functions, however, will conclude a different ranking order. In Figure 1, DB is contained in 3 out of 7 textual elements, and privacy is contained in 4 out of 7. Therefore, DB is more discriminative \( (\text{when compared to } te_1 \text{ and } a_1) \) is ranked higher than \( te_1 \) (and \( a_2 \)). This order violates TDC, by which \( a_2 \) should be ranked higher than \( a_1 \). A simple solution to fix this violation is to separate elements by their tag names [2, 3]: if we only consider the bio elements as a collection, privacy will be more discriminative. However, we still miscalculate \( idf \): DB is contained in 3 out of 4 bio elements and \( idf(DB) = \frac{4}{3} \). But there are only three authors to rank, and \( idf(DB) \) should be \( \frac{3}{2} \).

TDC is not the only constraint that is violated. Length normalization, formalized as \textit{Length Normalisation Constraints} (LNC) in [6], penalizes long documents even if they contain many occurrences of a query term. In the example query, since we are ranking authors, we should normalize authors’ lengths to resemble the effect of LNC. In Figure 1, though \( a_3 \) contains more occurrences of DB than \( a_1 \), since the total length of \( a_3 \)’s bio elements is much longer than \( a_1 \), \( a_3 \)’s score may be even lower than \( a_1 \). This coincides with the intuition that the two bio elements under \( a_3 \) may have duplicate information, or may be less specific, and thus \( a_3 \) is not necessarily more relevant than \( a_1 \).