XML 検索におけるデバッグ支援システム

内田 隆彦 (410017)

北陸先端科学技術大学院大学 情報科学研究科

2006 年 2 月 9 日

キーワード: XPath, デバッグ, 問い合わせ式, 検索式.

1 あらまし

XPath による XML データからの検索を行う場合，ユーザが記述した問合せ式になんらかの誤りがあったために，意図したデータが取り出せないということが，しばしば生じる．原因としては，タイプミスや，対象データの構造に関するユーザの勘違いなどが考えられる．そこで，本研究では，そのようなユーザによる問合せ式のデバッグ作業を支援するシステムを開発する．本研究で開発するシステムでは，ユーザが実行した問合せの結果が当初の意図に反しており，問合せ式に何か誤りがあったと思われる場合は，結果のどのような点が意図に反しているのかを指定することができる．システムはその情報を用いて，ユーザが当初に記述した問合せ式に近く，かつ，結果に関するユーザが指定した問題点を解消するような問合せ式を全て見つけ，それらを，データの統計情報などを用いた確率論的な手法で，ユーザが本来書きたかった問合せである可能性が高い順にランキングして表示する．

2 システムの概要

前章で述べたようなシステムを開発する場合，以下の三つの点について考慮する必要がある．

1. 問合せ結果のどのような点が意図に反しているのかをユーザにどのように指定させるか.
2. 当初の問合せ式に近くて，かつ，ユーザの指定した問合わせ結果中の問題点を解消するような問合せ式群をどのように見つけるか.
3. それらの問合せ式をランキングするために必要な,「各問合せ式が，ユーザが本来書きたかった問合せ式である可能性」をどうやって求めるか.

以下，これら三点について，順に説明する．

2.1 意図に反する点の指定方法

ここでは，問合せ結果のどこが意図に反しているのかを，ユーザは以下のいずれかの形式で指定することにする．

- このデータは問合せ結果に含まれるべきなのに含まれていないというデータを指定する．
- このデータは問合せ結果に含まれないべきなのに含まれているというデータを指定する．
- 問合せ結果が空集合であったが空集合となるはずがないと指定する．

2.2 候補となる問合せ式の発見

ここでは，ユーザが犯す可能性のある間違いとして，以下の物を考え，以下の間違い (または，それら複数の組み合わせ) によって，ユーザが書いた問合せ式へとなりうる物をまず考え，それらの中で，ユーザが指定した条件を満たす物を候補解とする．

2.3 候補となる問合せ式のランキングの方法

上述の方法で候補となる問合せ式を求めた結果，複数の問合せ式が候補となった場合，これらの候補を，ユーザが書きたかった問合せである可能性が高いものから順にランキングして表示することが望ましい．本研究では，そのような確率を求め，それに基づいて候補解をランキングする手法を開発する．

本研究の手法では，ユーザが書いた問合せ式が Q_0 であった時に，本当に書きたかった問合せ式は Q_1 であった確率を，ユーザが問合せ Q_1 を実行したい確率と，問合わせ式 Q_1 を書きたい時に誤って Q_0 と書いてしまう確率から，ベイズの定理を用いて求める．また，問合わせ式 Q_1 を書きたい時に誤って Q_0 と書いてしまう確率を求める際には，単純に Q_1 と Q_0 の近さだけでなく，Q_0 が単なる誤りによって書く確率が高そうな問合せ式であるか，それとも，単なる誤りで，うまくそのような問合せ式を書くような可能性は低いと思われるような特徴を持った問合せ式であるかという点についても考慮する．

3 まとめと今後の課題

本論文では，ユーザが XPath による XML 問合せを行う場合に，問合せ式のデバッグを支援するシステムの概要について提案し，特に，ユーザが誤った問合せ式を書いた時に，本来書きたかった問合わせ式であろうと思われる候補を，その確率が高そうな順にランキングする手法の基本的な考え方について示した．

しかし，現段階では，基本的なアイデアを示した段階であり，実際のシステムを実現するためには，今後，以下のような問題について検討する必要がある．

- ランキング計算の詳細，かつ，より形式的な定義
- 本文中でも述べたが，ユーザがある問合わせを実行したいと思う確率 $i(Q)$ をどのようにして与えるか．
- 様々な種類の誤りについて，ユーザがそれらの誤りを犯す確率をどのようにして与えるか．
- 今回，挙げたような三種類の誤り以外のもの，たとえば,「/」と「//」の誤りや,「and」と「or」の誤り等への拡張．
- 実装手法．
- 提案システムの有効性の評価をどのように行えば良いかの検討．

特に，実装手法については，非常に単純な実装方法として，ユーザが書いた問合せ式から与えられた誤りの種類によって到達できて，かつ，ユーザが指定した意図に合致する問合せ式を全て求め，それらについて確率を計算してランキングをするのでは，候補問合わせ式の数が多くなった場合，ランキングをユーザに提示できるまでの反応時間が非常に長くなり，使い勝手の悪いシステムになってしまうと思われる．よって，なんらかの手法を使って，最初から最終的なランキングで上位になる可能性が高いような解候補のみに絞り込んで計算を行うような，top-k 問合せの技術が必要になると思われる．よって，今後は，特に実装手法について検討を行う予定である．