4L-02

複数の類似度を考慮した木構造データに対する類似部分木検索

小久保柚真† 天笠俊之‡ 北川博之‡

†筑波大学情報学群情報科学類 ‡筑波大学計算科学研究センター

1 はじめに

JSON や XML などのフォーマットは半構造データと呼ばれ，木構造を持つという特徴がある．ビッグデータの利活用が叫ばれる中で，近年注目されている NoSQL に分類されるデータベースにおいてもこのようなデータが大量に蓄積されており，様々な場所で活用されている．

木構造データに対する類似検索は，重複の削除，剽窃検出，情報補完などに応用できる問題である．木構造データは文字列の類似度に加えて構造の類似度を考慮する必要があるため，古くから様々な研究が行われている．問合せとしてユーザが与えた木構造に対して，木構造データに含まれる全ての類似した部分木を見つける処理を類似部分木検索と呼ぶ．類似性を測るために，木構造の構造に基づく類似性と各ノードが持っている値（テキスト）に基づく類似性が考えられる．

木構造データの類似部分木検索を扱った研究として，TASM[1] や小柳らのテキストの類似度を考慮した手法 [2] がある．TASM では，類似度として木編集距離のみを採用しており，リーフノードについては値の完全一致だけを評価している．小柳らは Jaccard 係数をテキストの類似度として利用した手法を提案したが，同義語や類義語などといった単語の意味までは考慮していない．

また文字列の類似結合の分野において，Xu らによって複数の類似度を組み合わせて結合を行う手法が提案され，文字列間のより柔軟な類似性を検出できるようになった [3]．

本稿では，テキストの類似度として単語の重複だけでなく，シソーラス等から得られる概念階層や同義語規則に基づく単語の意味情報を利用した類似度を採用し，これらを組み合わせることによって，より柔軟な類似部分木検索手法を提案する．

Subtree similarity search over tree-structured data considering multiple similarities

Yuma KOKUBO†(kokubo@kde.cs.tsukuba.ac.jp),
Toshiyuki AMAGASA‡(amagasa@cs.tsukuba.ac.jp) and
Hiroyuki KITAGAWA‡(kitagawa@cs.tsukuba.ac.jp)
†College of Information Sciences, University of Tsukuba
‡Center for Computional Sciences, University of Tsukuba

2 提案手法

2.1 問題定義

提案手法では，データベース木 D に含まれる任意の部分木 D_i とクエリ木 Q に共通して含まれるタグを持つリーフ間でのみテキスト類似度を計算する．この際考慮する単語の重複，同義語規則，概念階層に基づく類似度の三つの類似度を以下のように定める．

$$sim_j(n_{D_i}, n_Q) = Jaccard(n_{D_i}, n_Q) \tag{1}$$

$$sim_s(n_{D_i}, n_Q) = \begin{cases} 1 & if \exists R \in \mathcal{R} \\ 0 & otherwise \end{cases} \tag{2}$$

$$sim_t(n_{D_i}, n_Q) = \frac{|LCA(n_{D_i}, n_Q)|}{max(|n_{D_i}| + |n_Q|)} \tag{3}$$

また D_i と Q 間の構造類似度，テキスト類似度を以下のように定める．

$$Sim_s(D_i, Q) = 1 - \frac{TED(D_i, Q)}{|D_i| + |Q|} \tag{4}$$

$$Sim_w(D_i, Q) = \sum^{|tag|} \frac{\max \sum_{k=1}^{|n_{D_i}|} \sum_{j=1}^{|n_Q|} I_{kj} \max_{\forall \mathrm{f}} (\mathrm{sim_f(n_{D_{i_k}}, n_{Q_j})})}{|tag|}$$
$$\text{where } I_{kj} = 0 \text{ or } 1, \sum_k I_{kj} \leq 1, \text{ and } \sum_j I_{kj} \leq 1 \tag{5}$$

ここで，D_i は D における帰りがけ順 i 番目のノードを根とする部分木を表し，n_{D_i},n_Q は D_i と Q に共通して含まれるタグの子要素であるリーフの文字列を表す．式 2 は同義語規則セット $\mathcal{R}$ 中に n_{D_i},n_Q の間で適用できる同義語規則 R が存在する場合 1 となる．式 3 の $|n_{D_i}|$ は概念階層における n_{D_i} の深さを表し，LCA(n_{D_i},n_Q) は n_{D_i},n_Q の最小共通祖先を表す．式 5 の分子は D_i,Q の間で同じタグを持つリーフの重み付き二部グラフマッチングとなる．

式 4,5 の二つの類似度をパラメータ α によって重み付けした値を D_i,Q 間の類似度とする．よって本手法では，類似度閾値 θ が与えられたとき, 以下の式を満たす全ての部分木 D_i を見つけることを目的とする．

$$\alpha \, \mathrm{Sim}_s(D_i, Q) + (1 - \alpha) \, \mathrm{Sim}_w(D_i, Q) > \theta \tag{6}$$

2.2 提案手法のアイデア

探索中に適用する類似度を決定するには時間がかかるため，事前にデータ木を走査し，各タグごとにリーフに適用する類似度を決定しておく．事前走査と類似

度計算時の規則の適用には，同義語規則セットと概念階層のノードテキストセットから作成したトライ木を利用することで，処理の高速化を図る．

また $|Q|$ が一定であること，および式 6 から類似条件を満たしている時の構造類似度の下限が決まっていることから，類似条件を満たす D_i のサイズを事前に計算することができる．これにより $|D_i|$ がこの範囲を超えている部分木は類似し得ないため，計算を省略できる．

2.3 アルゴリズム

提案手法のアルゴリズムの流れは以下のようになる．

1. D 全体を走査し，リーフを持つ各タグに対して適用する類似度を決定する
2. D を帰りがけ順に走査した結果から 1 ノード読み込む
3. 読み込んだノードを根とする D_i のサイズが類似し得る範囲内になければスキップ
4. D_i と Q のテキスト類似度（式 5）を求める
5. テキスト類似度が類似条件（式 6）を満たし得るなら構造類似度（式 4）を求める
6. 類似しているなら D_i を結果に追加する
7. 2 から 6 を D の全ノードに対して行う

3 評価実験

XML データセットを使用し，実行時間およびクエリのテキストに変更を加えたときの同一部分木の検出精度を計測した．D として医学文献のデータセットである MEDLINE を使用し，概念階層には MEDLINE の各文献に付与される MeSH を，同義語規則には MeSH の Entry Term を使用した．実験設定について，図 1 は $\theta = 0.8, \alpha = 0.5$, 表 1 は $\theta = 0.9, \alpha = 0.2$ とした．実験環境は 64bit windows10, Corei7@3.00GHz CPU, 32GB RAM で行い，コンパイラには x86 64-w64-mingw32 を使用した．

既存手法である TASM と小柳らの手法との実行時間の比較を図 1 に示す．既存手法では意味を考慮していないため，その分実行速度は早くなるが，TASM と比較すると意味を考慮してもなお実行時間について上回る性能が確認できた．

クエリの各リーフノードに対して，20 %の確率で単語の挿入，削除，同義語や類義語への置換を行ったうえで，変更を行っていない場合と同一の部分木を検出することができるかを 100 回行った時の検出回数を表 1 に示す．使用したデータセットにおいて同義語や類義語の含まれるタグの割合が少なかったため，大きな差とはならなかったものの二つのクエリにおいて小柳らの手法よりも検出精度が高いことが確認できた．

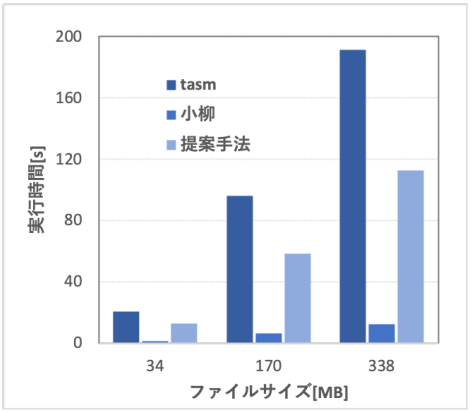

図 1: 既存手法との実行時間の比較

表 1: 既存手法との検出回数の比較

手法	クエリ 1	クエリ 2
小柳らの手法	72	74
提案手法	90	84

4 まとめ

テキストの類似度として単語の意味を考慮した類似度を組み合わせることで，より柔軟な類似部分木検索を実現する手法を提案した．また実験により，意味を考慮していない既存手法に匹敵する速度で実行できることが確認できた．

今後の課題として，ストップワードや数値の除去，JSON 形式のデータを用いた有効性の検証，分散表現などのベクトル表現を組み合わせることによる効果の検証などが挙げられる．

参考文献

[1] Nikolaus Augsten, Denilson Barbosa, Michael M. Bohlen, and Themis Palpanas. ”Efficient top-k approximate subtree matching in small memory”. IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 8, pp. 1123–1137, 2011.

[2] 小柳 涼介, 天笠 俊之, 北川 博之. ”テキストおよび構造の類似度に基づいた XML データに対する効率的な類似検索”. 第 6 回データ工学と情報マネジメントに関するフォーラム (DEIM2014), 2014.

[3] Pengfei Xu, Jiaheng Lu. ”Towards a unified framework for string similarity joins”. Proceedings of the VLDB Endowment, Vol. 12, No. 11, pp. 1289–1302, 2019.