1N-8

関連ファイルの発見におけるファイルRMC操作の考慮

呉 怡† 渡辺 陽介†† 横田 治夫‡

†,‡ 東京工業大学大学院 情報理工学研究科計算工学専攻
†† 東京工業大学 学術国際情報センター

1 はじめに

近年，ファイル数の急増により，ファイルシステム上に散らばるファイル間の関係を把握することが困難である．だが，ファイル内のテキスト情報を基にファイルを探索するデスクトップサーチは，文書を含まないファイルには有効ではない．この問題に対して我々は，ファイルアクセスログに注目し，キーワード非含有ファイルも検索できるFRIDAL[1]を開発し，また，同一作業に関連するファイルは頻繁に近い時間にアクセスされる性質を利用して，同一作業で使われたファイル群を発見する手法を提案している[2]．なおここでの作業とは，論文執筆のような複数のファイルをアクセスしながら行う論理的に一つの仕事である．

しかしこれまで，ファイルの改名（Rename）・移動（Move）・コピー（Copy）（RMC）操作を考慮しなかったため，RMC操作のあった両ファイルは全く異なるものとされてしまうという問題点があった．過去の作業で使ったファイルをコピーなどをして他の作業で再利用する場合では，RMC操作を考慮することで作業間の依存関係を知ることができる．そこで，我々はファイルRMC操作を考慮したタスク間関連度を利用して関連ファイルを発見する手法[3]とファイルを検索する手法[4]を提案している．本稿では被験者実験によって手法の有効性を確認する．

2 関連ファイルの発見

関連ファイルを発見するため，まず，同一作業に使われたファイルの集合をタスクとしてを抽出する（2.1節）．次に，タスク間の関連の強さを表すタスク間関連度を算出するため，タスクにあるファイルの重複の度合いを考慮したタスク間類似度とファイルRMC操作を考慮したタスク間RMC関連度を用いる（2.2節）．最後に，タスク間関連度を使って関連ファイルを発見する処理について述べる（2.3節）．

2.1 タスクマイニング

同一作業に使われたファイル群を抽出するため，参照・書込などの基本操作のほかにファイルRMC操作を含むファイルアクセスログを使用する．抽出対象は頻出アイテム集合（Frequent Itemset）である**FIタスク**と，RMC操作に基づく**RMCタスク**の2種類ある．

FIタスクは同一作業に使われたファイルは頻繁に近い時間にアクセスされるという考え方に基づくもので，我々はアクセスログを一定の時間幅($TransactionTime$)でトランザクションに分割し，全トランザクションにおいて$MinSuppCnt$回以上出現しているファイルの極大集合をFIタスクとする．

RMCタスクは，短時間にまとめてRMC操作されたファイル群は同一作業に由来する可能性が大きいことから抽出されるタスクである．

2.2 タスク間関連度

2.1節で得られたタスク$Task_i$をノードとした時，ノードをつなぐリンクの重みはタスク間関連度である．$Task_m$から$Task_n$へのタスク間関連度$R(Task_m \rightarrow Task_n)$は式(1)によって算出される．

$$\begin{aligned} &R(Task_m \rightarrow Task_n) \\ &= sim_t(Task_m \rightarrow Task_n)^{\theta_1} \\ &= rmc_t(Task_m \rightarrow Task_n)^{\theta_2} \end{aligned} \tag{1}$$

ただし，$sim_t(Task_m \rightarrow Task_n)$は後述するタスク間類似度で，$rmc_t(Task_m \rightarrow Task_n)$はタスク間RMC関連度である．また，$\theta_1$と$\theta_2$は$[0,1]$間の値をとるパラメータである．

タスク間類似度 多くの共通ファイルを使った作業間の関係が強いことに着目したタスク間類似度は，両タスクに含まれる同一ファイルの数を用いて算出される(式(2))．

$$sim_t(Task_m \rightarrow Task_n) = \frac{|Task_m \cap Task_n|}{|Task_m|} \tag{2}$$

タスクRMC関連度 他の作業で使ったファイルをコピーなどして再利用する場合がよくあるため，RMC操作のあったタスク間は関連が強いと考えられる．そこで，両タスクにまたがったファイルRMC操作を考慮してタスク間RMC関連度を算出する(式(3))．

$$\begin{aligned} &rmc_t(Task_m \rightarrow Task_n) \\ &= \sum_{(f_i,f_j)\in(Task_m,Task_n)} \{rmc_f(f_i \rightarrow f_j) \\ &* T(f_i,f_j) * E(f_i,f_j) * S(f_i,f_j)\} \end{aligned} \tag{3}$$

ただし，$rmc_f(f_i \rightarrow f_j)$はファイル間RMC関連度で，ファイル$f_i$を$f_j$にRMCした場合，操作の

Related File Discovery by Using the RMC Logs
Yi WU†, Yousuke WATANABE††and Haruo YOKOTA‡
†,‡ Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology
††Global Scientific Information and Computing Center, Tokyo Institute of Technology
†, †† {goi,watanabe}@de.cs.titech.ac.jp
‡ yokota@cs.titech.ac.jp

種類に応じて α_1，β_1，γ_1 のいずれかの値が付与され，ファイル f_j を f_i に RMC した場合には α_2，β_2，γ_2 の値が付与される．また，ファイル間 RMC 関連度は時間の経過や編集などによって弱くなることを考え，RMC 操作が発生してからの経過時間，編集回数，ファイルサイズの増減によるファイル間 RMC 関連度の減衰係数として $T(f_i, f_j)$，$E(f_i, f_j)$，$S(f_i, f_j)$ を用いる．

$$T(f_i, f_j) = \Delta_{time}(f_i, f_j)^{-\tau} \quad (4)$$

$$E(f_i, f_j) = \Delta_{edit}(f_i, f_j)^{-\epsilon} \quad (5)$$

$$S(f_i, f_j) = \Delta_{size}(f_i, f_j)^{-\sigma} \quad (6)$$

ただし，$\Delta_{time}(f_i, f_j)$ はファイル f_i，f_j 間で RMC 操作が発生してからの経過時間を表し，$\Delta_{edit}(f_i, f_j)$ は両ファイルに対する書込み操作の回数の和を表し，$\Delta_{size}(f_i, f_j)$ は両ファイルのサイズの増加分と減少分の絶対値の和を表す．また，τ，ϵ，σ はそれぞれ経過時間，編集回数，ファイルサイズの増減による減衰の影響を調節するためのパラメータである．

2.3 関連ファイルの発見

ファイル f_q の関連ファイルを発見するためにまず，ファイル f_q を含む $Task_i$ に対して $score(Task_i)^0 = 1$ とする．次に，FRIDAL[1] で用いた手法を基に全タスクに対してスコアを更新させる処理を K 回繰り返す．このように算出された $score(Task_i)^K$ の値を正規化して，$score(Task_i)^K > T_S$ を満たす全ての $Task_i$ にあるファイルを関連ファイルとする．

3 評価実験

RMC 操作を考慮したタスク間関連度を用いて関連ファイルを発見する手法の有効性を確認するため，我々の研究グループが日常的に使用している共有ファイルサーバ（Windows Server 2003, NTFS）におけるアクセスログで被験者実験を行った．

実験では，2010.4 から 2010.11 までのアクセスログを使用し，被験者がその間にアクセスしたファイルの中から，クエリファイルに関連するファイルを選出し，正解集合を作成する．なお，本稿では 4 名の被験者によって作成した 6 つの正解集合を用いて算出される適合率（precision）と再現率（recall）の平均で評価を行う．

RMC 操作の有用性を確認するため，表 1 で示す 4 構成で実験を行う．実験に使用したパラメータは，タスクマイニングに関して $TransactionTime = 3600sec$，$MinSuppCnt = 2$ とした．また，タスク間関連度に関しては $\theta_1 = \theta_2 = 0.5$，$\tau = \epsilon = \sigma = 0$，$T_S = 0$ に設定した．タスク間類似度のみ使用してタスク間関連度を算出する場合（構成 1，構成 3）は $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \gamma_1 = \gamma_2 = 0$ に設定し，タスク間 RMC 関連度と組み合わせる場合（構成 2，構成 4）は $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \gamma_1 = \gamma_2 = 1$ にした．

実験結果は図 1 で示すように，タスクマイニングとタスク関連度の算出において RMC 操作を考慮した構成 4 が最も高い F 値（0.557）を得られ，また，RMC 操作を考慮した構成 2，3，4 の再現率は構成 1 に比べて大きく改善したことに対して，適合率の下げ幅が小さいことから，提案手法が有効であることを示した．

表 1: 実験構成

	タスクマイニング	タスク間関連度
構成 1	FI タスクのみ	類似度のみ
構成 2	FI タスクのみ	類似度＋ RMC
構成 3	FI ＋ RMC タスク	類似度のみ
構成 4	FI ＋ RMC タスク	類似度＋ RMC

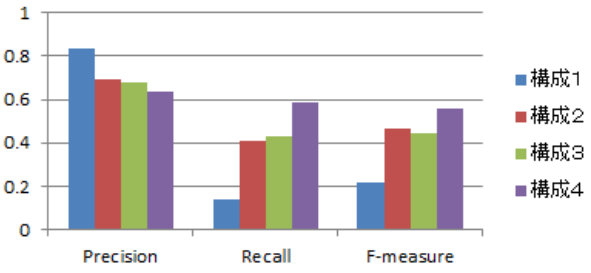

図 1: 実験結果

一方で，発見できなかったファイルを調べた結果，その多くはあまりアクセスされていなかったファイルで，タスクとして抽出できなかったものが多い．そのため，タスクに入っていないファイルの対処を今後の課題として考えていきたい．

4 まとめと今後の課題

本稿では、RMC 操作を考慮したタスク間関連度を利用した関連ファイルの発見手法を提案し，被験者実験により有効性を示した．

今後の課題として，最適なパラメータを調べることや結果の提示方法の検討することなどが挙げられる．

謝辞

本研究の一部は，日本学術振興会科学研究費補助金基盤研究 (A)(#22240005) および文部科学省科学研究費補助金特定領域研究 (#21013017) の助成により行われた．

参考文献

[1] Tetsutaro Watanabe, Takashi Kobayashi, and Haruo Yokota. A method for searching keyword-lacking files based on interfile relationships. In *OTM '08*, pp. 14–15, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] 小田切健一，渡辺陽介，横田治夫．頻出ファイル集合のアクセス時間を考慮した仮想ディレクトリ生成手法．第 2 回データ工学と情報マネジメントに関するフォーラム (DEIM2010), 2010.

[3] 呉怡, 渡辺陽介, 横田治夫. ファイル RMC 操作を考慮した関連ファイルの発見. 第 150 回 データベースシステム研究発表会, 第 2010-DBS-150 巻, 2010.

[4] 呉怡, 渡辺陽介, 横田治夫. ファイル rmc 操作に基づくタスク間関係を用いたファイル検索. 第 3 回データ工学と情報マネジメントに関するフォーラム (DEIM2011), 2011.