ファイル整理ツール HyperClassifier における 多次元ツリー自動構成ツールの開発と評価

山口 章太† 掛下 哲郎†

†佐賀大学大学院工学系研究科 〒840-8502 佐賀県佐賀市本庄町 1 番地

E-mail: †{yamasho, kake}@cs.is.saga-u.ac.jp

あらまし 我々は、分類観点ごとに個別のツリーを構成し、これらを組み合わせることで、コンピュータ内のファイルを系統的に分類するツール HyperClassifier の開発を行っている。しかし、HyperClassifier ではツリーの構成やファイルの登録に関して手動による操作に頼る部分が多く、既存のファイルを登録して検索可能となるまでに大きな手間を要する。そこで、移行の対象とするファイルのパスから分類に用いる多次元ツリーを自動構成するツールを提案し開発を行った。本稿では、その評価実験について結果と考察を述べる。処理時間に関しては、入力データの規模 n が大きくなるほど、処理工程により最大 $O(n^3)$程度の処理時間がかかることが分かった。また、ツリーの品質に関する評価を行った結果、現在の自動構成ツリーは分類観点の一貫性が低く、また、不要な単語がツリーに多く存在することから、これらに対し改善を行う必要があることが分かった。

キーワード OLAP、多次元データベース、ファイル整理

1. はじめに

近年、多くの企業でコンピュータが導入され、大量の情報を電子的に蓄積している。その数は、企業の規模などにもよるが、数万から数十万ファイルにものぼる。蓄積されているファイル数は増大の一途をたどっているが、大量のファイルが蓄積されると、それらの整理や検索が困難になってくる。情報を探すことだけに時間を割かれるため、業務の能率低下につながり、ひいては企業の運営コストを増大させる要因ともなっている。このような背景から、ファイルを系統的に分類・整理し、素早く検索が行えるようなシステムが求められている。そこで、本研究室では、ファイルを系統的に分類・整理し、高速に検索する機能を提供するツール HyperClassifier [1]を提案し、開発を行っている。

HyperClassifier では、多次元ツリーを用いて分類したファイルを、OLAP 操作を用いて柔軟に検索し、検索されたファイルに対する各種の一括操作を提供している。多次元ツリーを用いることで、ツリーを観点別に分け、各ツリーを単純化することにより理解容易性が高まること、複数の観点の項目をツリーから選択することで絞り込み検索が可能であること、蓄積された情報の全体像を把握しやすくなることが利点として挙げられる。しかし、HyperClassifier を導入する際には、既存のフォルダ階層やファイルを、多次元ツリーを用いて再分類する必要がある。

我々は過去に、実データを用いた多次元ツリーの構築を手入力によって行い、その評価および考察を行った[2]。その結果、手入力による分類では、元のファイル階層の各フォルダを多次元のツリーと対応付ける作業など、ファイルシステムを HyperClassifier で使用できるようにするまでに大きな手間を要することが分かり、その他にも、ファイル移行に関するさまざまな課題が明らかになった。

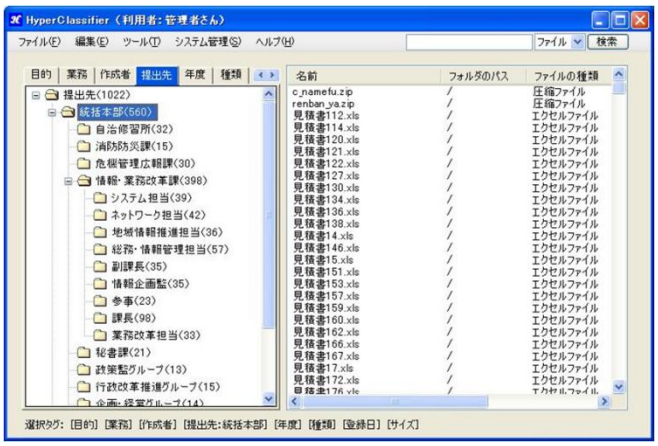

図 1：HyperClassifier の画面

そこで、ツリー移行のための分類作業を自動化および簡略化することを目的として、多次元ツリー自動構成ツールを提案し、開発を行っている[3]。今回は、多次元ツリー自動構成ツールの評価実験を行った。本稿では、多次元ツリー自動構成ツールとその評価実験について報告する。

2. 多次元ツリー自動構成アルゴリズム

多次元ツリー構成ツールは、移行の対象とするファイルのフルパス情報に基づいて多次元ツリーを自動構成し、これを用いて移行対象ファイルを分類する。

多次元ツリー自動構成ツールは以下のステップ 0～4 の 5 ステップによって構成されている。そのうち多次元ツリーの自動構成は以下のステップ 0～3 に従って行われ、ステップ 4 についてはツールの利用者が高水準操作を用いて手動操作により行う。

060 用地第一課¥10 月末¥H21 進行管理(10 月末).xls
060 用地第一課¥H19~H20 各課引継書¥旧用地 2 課¥引継書(佐賀大和).xls
060 用地第一課¥H19~H20 各課引継書¥旧用地 2 課¥引継書(多久武雄).xls
060 用地第一課¥H19~H20 各課引継書¥旧用地 2 課¥引継書様式(佐賀城公園).xls
060 用地第一課¥課長引継書.xls
060 用地第一課¥基金¥H20 基金決算.xls
060 用地第一課¥平成 21 年度用地企画担当¥一般文書¥証明書.doc
060 用地第一課¥未登記関係¥21 年度¥H21 地籍測量図配当要求.xls
060 用地第一課¥未登記関係¥21 年度¥未登記解消状況(様式).xls
060 用地第一課¥未登記原因調(H21 年度)H21 現在.xls
060 用地第一課¥未登記台帳.xls
:

図 2：入力に用いるファイルパスの一覧の例

抽出単語	対応するフォルダ
平成 21 年度	平成 21 年度宅地造成費早見表
宅地造成費	平成 21 年度宅地造成費早見表
早見表	平成 21 年度宅地造成費早見表
6 月 1 日	特記仕様書(平成 21 年 6 月 1 日変更)
特記仕様書	特記仕様書(平成 21 年 6 月 1 日変更)
平成 21 年	特記仕様書(平成 21 年 6 月 1 日変更)
変更	特記仕様書(平成 21 年 6 月 1 日変更)
:	:

図 3：単語とフォルダ名の対応情報の例

2.1. ステップ 0：元のツリーの構築

図 2 のような元のファイル階層に含まれるファイルのパスを列挙したテキスト形式のファイルを入力として与え、移行の対象となるフォルダのツリー構造をツール内に構築する。移行の処理では、このステップで構築したツリーを基準として、再構成操作を用いて多次元ツリーの作成やツリー内でのノードの整合性の判定に使用する。

ツリーの構築は、入力されたパスを階層別に分解し、各階層においてノードを作成する。ノードは、ファイルシステムのフォルダ名・ファイル名の規則に従い、同一ノードの子ノード間では同名のフォルダが重複しないようにノードを作成する。同一ノードの子ノードに同名のノードが作成済みの場合は、その場所には新たに同名のノードは作成されない。ツリーの各ノードはノード ID・ノードの元の名前・ノードを構成している単語を登録する。また、元のファイルシステム上で該当フォルダに格納されているファイルをノードに対応付け、その情報を保持する。

2.2. ステップ 1：単語辞書の構築

ツリーに含まれている単語の情報を保持するための単語辞書を作成する。元のファイル階層に含まれるフォルダ名およびファイル名は多くの場合、複数の単語によって構成されている。フォルダ名やファイル名は、そのファイル等の作成者が分類を行うために名づけたものであるため、分類に有用なキーワードはフォルダ名やファイル名に多く含まれていると考える。そこで、フォルダ名およびファイル名に対し、形態素解析を行って単語に分解する。形態素解析エンジンには MeCab[4]を用いた。形態素解析によって切り出された単語から自立語を抽出して、図 3 のような抽出した単語と、抽出元となるフォルダ名の対応関係を保持するテキスト形式の入力ファイルを作成する。そのファイルをツールに入力として与えることで、単語辞書にツリーが持つ単語を登録することができる。単語辞書は、各単語にカテゴリ ID（初期値は空）を付与したデータ構造である。すでに同一の単語が単語辞書に登録されている場合は、重複登録を行わないものとする。

2.3. ツリーの再構成操作

ツリーの再構成操作は、単語集合およびツリーを入力として、指定した単語集合のみを含むノードで構成されたツリーを出力する操作である。この操作はステップ 2 で単語の切り出しを行ったときのツリーの更新、ステップ 3 で単語を新たに多次元ツリーに分類したときのツリーの更新や単語分類時のツリー内での重複や、親子関係の矛盾のチェック、ステップ 4 で単語の移動や削除を行ったときなどのツリーの更新時に使用される。この操作は、対象となるツリーに対して直接操作を行う。元のツリーなど、構成を変更せずに保持しておきたいツリーを対象に操作を行う場合、ツリーをコピーしたうえでこの操作を行うこともできる。それに対して各ノードの単語集合をチェックし、各ノードから入力された単語集合に存在しない単語を削除する。その結果、単語を 1 つも持たないノードをツリーから削除する。

削除されたノードの子ノードは、直近の祖先ノードに対応付ける。ツリーは兄弟ノード間で単語集合の重複が生じないように作成し、ノードの削除により同一単語集合のノードが兄弟ノード間で出現した場合は、それを統合して 1 つのノードとする。この再構成操作は、どのような単語集合を入力しても実行でき、元のツリーの各ノードの階層の関係を保持したまま単語単位でツリーの構成を行うことができる。

2.4. ステップ 2：ツリーからの重複語切り出し

多次元ツリーでは、ノードを観点別に分離するために、ツリー内で同一ノードが複数箇所に出現しないように構成する。ツリーに同一の単語が複数箇所に出現しているのは、そのツリーの中に複数の分類観点が含まれているためと考えられるからである。そこで、ス

テップ 0 で作成した元のツリーに再構成操作を適用して複数回出現する単語を除去する。これにより、元のツリーから、ノードの重複がないツリーを作ることができる。

重複語の切り出しは、元のツリーを上位レベルから探索し、同一のノードが複数の箇所に出現しているか判定する。複数箇所に出現している単語は、元のツリーから該当単語を取り除いた状態で、元のツリーに再構成操作を行うことにより､元のツリーから取り除く。取り除いた単語は重複語として別のリストに保持する。この操作をツリー内の重複語が全て切り出されるまで行う。この処理を実行した後に元のファイル階層に残された単語のカテゴリ ID を 1 とし､1 つ目の多次元ツリーを作成する。ステップ 2 で除去された単語は、ステップ 3 への入力とし、2 つ目以降の多次元ツリーに配置する。

2.5. ステップ 3: カテゴリの自動分類

ステップ 2 で取り出した単語を、元のツリーにおける構成済みのツリーのノードとの関係をチェックし、同一ツリー内での重複や、親子関係の矛盾が生じないように多次元ツリーに配置する。単語が配置されたツリーに応じてカテゴリ ID を設定する。ノードの重複や親子関係の矛盾のチェックは、作成済みの多次元ツリーに該当単語を追加した状態で追加先のツリーに対して再構成操作を行い、該当単語を持つノードが出力されたツリー内にただ 1 つのみ存在するかどうかで判定を行う。

各単語について、配置可能なツリーが複数存在する場合は、その単語をツリーに配置したことによってそのツリー内で未分類(根ノードに対応付けられている)ファイルの個数を配置候補となる各ツリーについて調べ､未分類ファイル数が少なくなるツリーに配置する。未分類ファイル数が同数となるツリーが複数存在する場合､その単語はただちに分類を行わず､保留とする。保留となった単語は、他の単語の分類がひと通り完了した後で再び同様の方法で分類を行う。それでも配置するツリーが決定できない単語がある場合は、乱数を用いて配置するツリーを決定する。

2.6. ステップ 4: ツリーの洗練

ステップ 0～3 の工程で自動構成された多次元ツリーは、異なる観点のノードを含むなど、妥当とは言えない部分を含む可能性がある。そこで、利用者が多次元ツリーをチェックし、必要に応じて編集するための高水準操作を提供する。ユーザは、作成された多次元ツリーが適切な構成になるように整理を行う。ツリーの洗練操作として、以下の操作を実装している。

- **追加先候補のある単語の強調表示**
  自動構成された各ツリーの単語のうち、他のツリーに移動させてもツリー内での重複が発生しない単語を色分けして表示する。これにより、利用者はツリーの洗練の余地がどれだけあるかを確認することができる。
- **ツリー内での重複単語の表示**
  指定したツリーにおいて、複数の箇所に出現している単語を色分けして表示する。利用者に移動すべき単語を提示し、洗練作業を支援する。
- **指定した単語の移動先ツリー候補の表示**
  指定した 1 個以上の単語に対し、その単語を移動させてもツリー内での重複が発生しないツリーを一覧表示する。単語の移動操作を行う際に、候補を提示することで、それらのツリーに移動させても問題がないことを確認できる。
- **ツリー間での単語の移動**
  選択した単語に対し、配置可能なツリーの候補を表示し、別のツリーに単語を移動させ、カテゴリの分類を適正にするための操作である。単語と移動先のツリーを選択すると、移動元ツリーでは移動対象の単語を除いた状態で再構成操作を行い、また、移動先ツリーでは移動対象の単語を追加した状態で、それぞれ再構成操作を行う。これらの処理により、ツリー間で単語を移動させることができる。移動操作の際は、移動させる単語と移動先のツリーを指定すればよく、ツリー内での具体的な移動先を指定する必要はない。ツリーの再構成操作により、単語を適切な位置に移動させることができる。
- **単語辞書からの単語の削除**
  選択した 1 個以上の単語をツリーから削除する。対象のツリーの単語集合から指定した単語を取り除いた状態で、対象のツリーの再構成操作を行うことで､単語の削除を行うことができる。また、単語辞書上で削除対象の単語のカテゴリ ID を無効値に指定し、ツリーに配置されないように処理する。この操作により、フォルダ名・ファイル名に含まれている単語のうち、不要な単語を取り除くことで、ツリーの理解容易性を高めることができる。
- **多義語の分割**
  多次元ツリーで複数のツリーに配置されている同一表記の単語に対し、配置されている場所ごとに別々の単語 ID を割り当て、異なる意味の単語として扱う。対象となるツリーの単語集合に対象単語の ID を分割後の複数の ID に書き換え、それを用いて、ツリーの再構成操作を行って単語の分

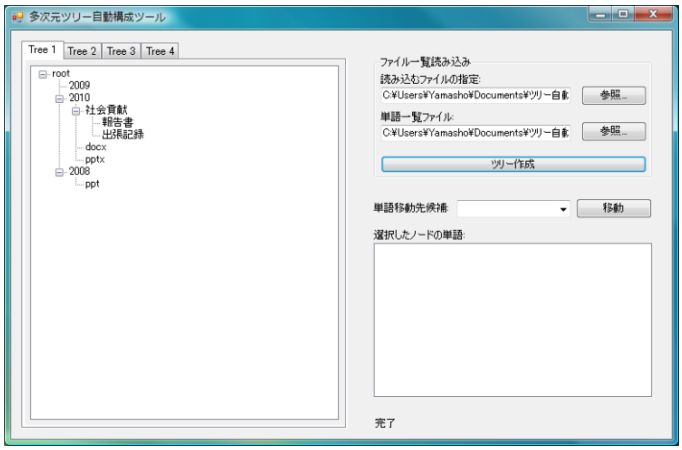

図 4：多次元ツリー自動構成ツール

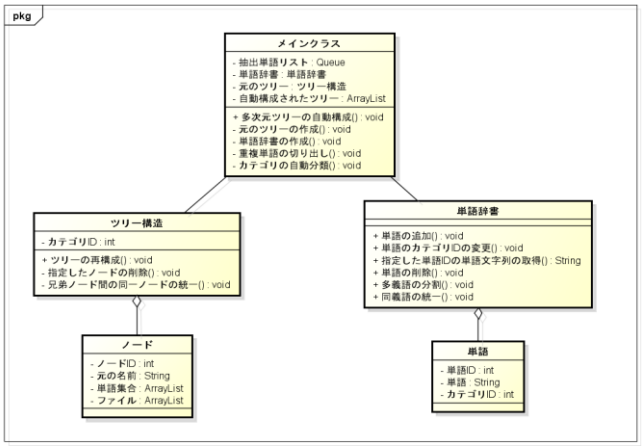

図 5：多次元ツリー自動構成ツールのクラス図

割をツリーに反映させる。

- **同義語の統一**
  表記ゆれなど、意味が同一でも文字列上で異なる表記の単語を同じ意味の単語として指定する。対象となるツリーの単語集合を統一する単語の ID を同一のものに変更し、そのツリーに再構成操作を行うことで、同義語の統一をツリーに反映させる。これにより、兄弟関係にある同義語のノードが統合され、ツリー構造の理解容易性を高められることが期待される。

3. 多次元ツリー自動構成ツールの設計

本ツールは Visual Studio 2008 を開発環境とし、C# を用いて開発を行っている。図 4 は、本ツールの画面である。多次元ツリー自動構成ツールは、ユーザからの操作を受け付け、各種処理を行うメインクラスと、多次元ツリーの構成に必要な各種のデータ構造を定義するクラスで構成されている。本ツールのクラス図を図 5 に示す。以下で各クラスの概要を説明する。

◎ メインクラス

ユーザからの入力を受け付け、それに対する各種処理を行う。本クラスは図 2 のフォームと対応しており、読み込むファイル一覧のパスを入力する処理、ツリー構築を開始するボタン、および、構成したツリーを表示する TreeView に対応したイベントハンドラを実装している。またステップ 2 において元のツリーより切り出した重複出現単語を「抽出単語リスト」として保持する。

また、ステップ 4 におけるツリーの洗練に関する高水準操作は、このクラスのフォーム上で行えるようになっている。

◎ ツリー構造クラス

ステップ 0 で作成される元のツリー、および、ステップ 3 までの処理により自動構成したツリーを保持する。ツリー構造は複数のノードによって構成され、ノード同士が他のノードと接続し階層構造を構成する。単語の文字列自体は直接保持せず、単語 ID を保持する。ノードの単語を表示する際は、単語辞書クラスの「指定した単語 ID の単語の文字列を取得する」メソッドを用いて参照するようにする。ツリー構造の各ノードは、ノード間の親子関係のほかに、以下の情報を持つ。

変数名	保持するデータ
ノード ID	ノードを一意に識別するための番号
元の名前	元のファイル階層において、そのノードに対応するフォルダまたはファイルにつけられている名前
単語	そのノードに対応するフォルダまたはファイル名から抽出された単語 ID の集合
ファイル	そのノードに対応付けられているファイルの集合

ツリー構造に対し、以下の操作を実装する。

- **ツリーの再構成**
  2.3 節に挙げたツリーの再構成操作を行う。以下の 2 つの操作は、ツリーの再構成操作において、サブメソッドとして用いるものである。
- **指定したノードの削除**
- **兄弟ノード間で同一のノードを 1 つに統合する**

◎ 単語辞書クラス

ファイル名およびフォルダ名から取り出した単語を保持し、各単語について配置されたツリーのカテゴリ ID を保持する。ステップ 1 で本クラスのインスタンスが作成され、元のツリーが持つすべての単語が登

録され、単語 ID が割り当てられる。ステップ 3 の処理においてカテゴリ ID が設定される。各単語は以下の各要素からなる構造体である。

変数名	保持するデータ
単語 ID	単語を区別するための番号。同一表記の単語でも、意味が異なる場合は、異なる番号をつける。逆に、表記が異なっていても、同一の意味である場合には同じ番号を付ける。単語に関する各種判定は、単語 ID を用いて行う。
単語	単語の文字列
カテゴリ ID	その単語が、新規に構成される多次元ツリーのうち、どのツリーに属するかを表す番号

単語辞書に対し、以下の操作を実装する。

- **単語辞書に単語を追加する**
  ステップ 1 で単語を登録する際に使用する。
- **指定した単語のカテゴリ ID を変更する**
  ステップ 2 および 3 で単語の分類が決定した際に個の操作を行ってカテゴリ ID を書き換える。
- **指定した単語 ID に対応する単語を取得する**
  ステップ 3 までで作成したツリーを、ステップ 4 において表示するときに、ツリー構造の各ノードが持っている単語 ID に対応する単語を取得する。

4. 処理時間に関する評価

4.1. 評価方法

2 節および 3 節で述べたアルゴリズムおよび設計に基づいて開発した多次元ツリー自動構成ツールを用いて評価実験を行う。この実験では、自動構成されるツリーの構造や規模、処理時間などを評価する。

入力データには、ファイルシステム階層構造を所属する各ファイルのフルパスを列挙したテキストファイル「ファイルパス一覧」と、ファイルシステムに含まれている単語の集合とその単語が用いられているフォルダ名の対応リスト「単語一覧」の 2 種類を用いる。

「ファイルパス一覧」は、佐賀土木事務所で実際に用いられていたファイル階層（約 8,000 ファイル規模）を使用する。データ数に応じて処理がどのように変化するかを確認するため、元の入力データから一定のデータ数を切り出したものを別のファイルとして作成し、入力データとして用いた。「単語一覧」については、ファイル数別の各ファイルパス一覧は、全て同じ単語集合を含む同一のツリーから切り出しているため、ファイルパス一覧の入力データ数に関わらず同一のものを使用している。また、本プログラムの動作には、本研究室で研究用に使用しているデスクトップ PC（OS：Windows Vista Business、プロセッサ：Intel Core2 Duo 3.16GHz、物理メモリ：2GB）を用いた。

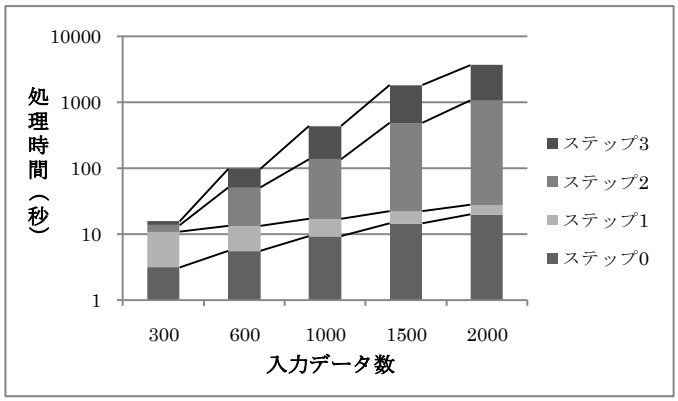

図 6：入力データの規模による処理時間の変化

本実験では、入力データ数（元のツリーに含まれるファイル数）の変化によって、処理にかかる時間や特定の処理を実行した回数がどのように変化するかを検証した。なお、入力に用いる元のツリーのデータは、ツリーに含まれる分類対象のファイル数を基準として、300 から 2,000 ファイル分の 5 種類の規模のツリーを用いた。

4.2. 評価結果と考察

本実験では、入力データ数の変化によって、処理にかかる時間や特定の処理を実行した回数がどのように変化するかを検証した。ツリー規模別の処理時間の結果を図 6 に示す。ただし、データ数が少ない場合と多い場合で処理時間に大きな差があるため、縦軸の処理時間は対数目盛としている。

本実験において、入力ファイル数が多くなるほど、多次元ツリー構成の処理にかかる時間は急速に増加することが分かった。現在の自動構成処理は、アルゴリズム上において、分類対象のファイル数を n とした場合、ステップ 0、1 および再構成操作は $O(n)$、ステップ 2 は $O(n^2)$、ステップ 3 は $O(n^3)$の処理時間を要し、処理全体においても、概ね $O(n^3)$の処理時間を要する。本評価実験において実際にプログラムを動作させた結果、実際の処理に要した時間が概ねこれらのオーダーに従うことを確認できた。

一般的なコンピュータでこのプログラムを用いて 2,000 ファイル規模のファイル階層構造から多次元ツリーを構成する場合、処理が完了するまでに 1 時間程度かかる。ファイルの規模がそれ以上に増えた場合に、数時間かそれ以上の規模で、処理時間がかかるものと考えられる。本ツールは大規模なファイル構造を容易に導入することを目的に開発しているため、アルゴリズムの改良を行い、大規模なツリー構造に対しても、

より短時間で多次元ツリーが構成できるようにする必要がある。特に、ステップ 2 および 3 では、ファイル数増加に伴う処理時間の増加が顕著であるため、この部分の処理について重点的に改良を行う必要がある。

5. ツリーの品質に関する評価

5.1. 評価方法

多次元ツリー自動構成ツールが生成したツリーの品質に関する評価は以下に挙げる観点から行う。

- 各ツリーの単語が分類観点の一貫性を保っているか
- 同一観点の単語が複数のツリーに分散していないか
- ファイルの分類に有用な単語がツリー内にどの程度含まれているか

ツリーの分類観点の一貫性や分散状況の評価は、過去の研究[2]で手作業により作成し、元のツリーを作成した佐賀土木事務所のレビューを受けた多次元ツリーを模範のツリーとし、その各ツリーの単語集合と本ツールにより自動構成された各多次元ツリーの持つ単語集合との類似度を計算することで行う。2 つのツリーの類似度は以下の計算式により算出する。

$$\text{類似度} = \frac{\text{2 つのツリーに共通する単語数}}{\text{2 つのツリーが持つ単語の和集合の単語数}}$$

類似度を求めるためには、自動構成されたツリーと模範となるツリーの各単語が一致しているか否かを判定する必要があるが、そのための基準としては以下の 3 つを用いる。

- 2 つの単語が完全に一致する（完全一致）
- 一方の単語が他方の単語の部分文字列である（包含一致）
- 2 つの単語が共通の部分文字列を含む（部分一致）

以上の各場合について、全体の単語集合のうち同義の単語とみなせるものの割合を算出した。それに応じて各場合を重み付けし、類似度を求める際に反映させた。模範となるツリーの単語と何らかの形で一致した単語のうち、意味的に妥当であるものの割合を調べたところ、包含一致では 51.4%、部分一致では 5.0%であった。この割合をもとに、包含一致の場合を 1/2 単語、部分一致の場合を 1/20 単語として、共通する単語数に加えることとした。

表 1：自動構成ツリーと模範ツリーの単語数

自動構成ツリー	
ツリー番号	単語数
1	833
2	214
3	70
4	22
5	8
6	8
7	4

模範ツリー	
観点	単語数
プロジェクト	336
文書の種類	198
引継ぎ	2
作成日	68
作成者	36

表 2：ツリーの類似度（単位：%）

		自動構成ツリー						
		1	**2**	**3**	**4**	**5**	**6**	**7**
模範ツリー	**プロジェクト**	9.75	7.16	3.17	2.43	1.07	0.16	0.32
	文書の種類	10.88	13.17	8.55	1.48	1.52	0.27	0.52
	引継ぎ	0.13	0.28	0.21	0.21	0.50	0.00	0.00
	作成日	2.90	1.67	1.05	0.72	0.66	0.00	0.07
	作成者	0.02	1.40	0.75	0.34	1.14	0.11	0.13

5.2. 評価結果と考察

本評価では、オリジナルのファイル階層から 1 階層目の特定のフォルダをルートとする部分ツリー（ファイル数 1,336）から多次元ツリーを自動構成し、各ツリーの品質を調べた。自動構成した各ツリーと、その類似度判定の対象に用いた模範ツリーの単語数を表 1 に示す。自動構成された各ツリーと各観点の模範ツリーとの類似度を表 2 に示す。

自動構成されたツリーと模範ツリーとの類似度は高い観点でも 10%程度であり、分類に有効なツリーが生成されているとはいえない。また、各ツリーに複数の観点が分散しており、各ツリーの分類観点の一貫性が保たれていない。これは、本ツールの自動分類方法では、単語の意味は考慮しておらず、ツリーの各ノードの対応関係とノードの単語集合によって分類を行っているため、分類の精度に限界があるものと思われる。分類に有効なツリーにするためにはステップ 4 のツリーの洗練操作において、各ツリーが分類の一貫性を持つように整理する必要がある。しかし、自動構成されるツリーであまりに一貫性が低いと、後に行う洗練操作の手間が増加してしまう。そのため、自動構成の時点でより分類観点の一貫性のあるツリーを構成できるように、プログラムの改良を行う必要がある。

続いて、ツリー内の単語の分類有用性について考察する。模範ツリーの単語のうち、自動構成された多次元ツリーに含まれていなかった単語の割合は 2.3%で、手作業で分類した際に分類に有用と判断した単語は自動構成ツリーにおいても、概ね網羅されていることが

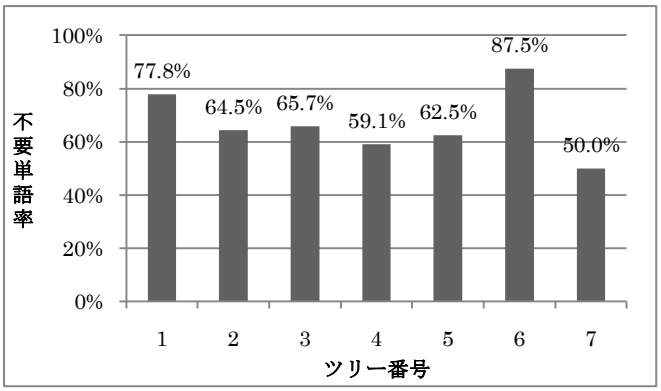

図 7：各自動構成ツリーの不要単語の割合

分かった。

5.1 節でも述べたように、模範ツリーと一致した自動構成ツリーの単語のうちで妥当な意味を持つものは、包含一致ではおよそ半数、部分一致では 5%程度である。模範ツリーの単語と完全一致するものについては分類に有用であることは自明である。包含一致する単語については、およそ半数が意味上で一致する単語であるから、分類においてある程度有用性があるものと考えられる。一方、部分一致する単語はその大半が意味上は模範ツリーの単語と一致しないものであるため、分類に有用なものはないとみなすことができる。ここで、自動分類ツリーの単語のうち、模範ツリーのいずれの単語とも分類に有用な一致（完全一致または包含一致）をしない単語を「不要単語」と定義する。

入力に用いた元のファイル階層において、含まれる全単語のうち模範ツリーが持つどの単語に対しても不要単語となる単語割合は 72.1%であった。この数値より、自動構成されたツリー全体の中の多くが不要単語で占められていることが分かった。不要単語が多数存在することにより、ツリーの類似度および分類観点の一貫性を低くし、さらにツリー構造を複雑にしているものと思われる。このような分類に不要な単語を、自動構成処理の時点で取り除くことで、より分類に有用なツリーを構成できるものと思われる。

また､ステップ 2 の重複語切り出しの処理において、元のツリーに複数回出現している単語とそうでない単語を分離してから分類する方法は、各ツリーの不要単語を減少させるのにどの程度効果があるのかを調べた。これを評価するために、自動構成された各ツリーにおいて、不要単語の数を調べ、各ツリーの全単語数に対する割合を算出した。なお、ツリー1 はステップ 2 の単語切り出し処理で重複単語を切り出した後のツリーで､他のツリー2～7 は切り出した重複単語をステップ 3 で分類して生成したツリーである。不要単語率を算出したところ、図 5 に示すような結果となった。ツリー1 では 77.8%、ツリー2～7 では平均で 64.9%であった。ツリー2～7 において分類に有用でない単語の割合は、ツリー1 に比べて 13 ポイント程度低くなった。このことから､ステップ 2 の重複単語の切り出し処理は、不要な単語を取り除くうえで一定の効果があることが分かった。

6. 関連研究

HyperClassifier と同様、複数のツリーを用いてファイルを整理するソフトウェアとしては[5,6]などがある。

[5]は、ファイルの登録に特化した機能の提供に留まっており、OLAP 操作を組み合わせた検索や、ファイルの一括操作などの機能は提供していない。また、既存ツリーとの対応は手作業で設定する方式になっているため､大規模なファイルシステムへの適用は難しい。

ジップインフォブリッジ社の SAVVY/EWAP[6]は、独自開発の全文検索システム SAVVY と多観点ツリーを利用したファイル検索によって企業内ファイルの「見える化」を図っている。SAVVY/EWAP は、基本的に全文検索システムであり、既存のフォルダ階層の各レベルをそのまま分類基準に対応づけているため、ツリーと分類の観点が 1 対 1 には対応せず、系統的なファイル整理が実現できない。また、ファイルの登録は別ソフトで管理者のみが行う。

これに対して、HyperClassifier は、以下に示す各種の機能を提供している。

(1) 観点ごとに定義したツリーを用いたデジタル情報の系統的な整理
(2) OLAP 操作（ダイシング、スライシング、ドリリング）を用いた直観的な情報検索
(3) ドラッグ＆ドロップによる情報の分類＋属性情報に基づく自動分類
(4) 絞り込まれたファイルに対する各種の一括操作（例：アクセス権設定、印刷、圧縮）
(5) 既存の階層フォルダを活用した多次元ツリーの自動生成

また、複数のツリーを用いてファイルを整理することを前提とする発明として[7]がある。[7]では、ドキュメントファイルの管理効率の向上に寄与するため、管理対象であるドキュメントに関連するメタデータを取得し、取得した複数のメタデータそれぞれの属性に基づいて、該複数のメタデータをツリー形式で階層表示する。しかし、本発明は多次元ツリーを自動生成する技術ではなく、何らかの方法で生成されたツリーに対して、メタデータを用いてファイルとツリーを対応付ける技術である。

7. おわりに

今回の評価実験では、多次元ツリー自動構成ツール

の自動構成処理の実行時間および構成されるツリーの品質に関する評価を行った。処理時間に関しては、数線及び数万ファイル規模の入力データに対しては実用には適さないほどの膨大な処理時間がかかることが分かった。ツリーの品質に関しては、生成されるツリーは分類観点の一貫性が低く、また、分類に有用でない単語がツリー内に多く含まれており、ツリーの洗練操作を行うにしても、多くの手間を要することが分かった。

この結果から挙がった課題をもとに、アルゴリズムおよびプログラムの改良を行うことを計画している。処理時間については、より高速に処理を行うためのアルゴリズムの改善を行うことが挙げられる。現在検討している案では、実際の自動構成を行う前に、あらかじめ小規模なツリーを自動構成し、その分類データを学習データとして保持しておき、そのうえで、実際のツリーの自動分類を行うことで、自動構成処理の処理時間短縮につながるものと考えている。また、この手法は、ツリーを自動構成した後、新たにファイルやフォルダを追加する際のツリーの更新にも有用であると考えている。

ツリーの品質については、分類に不要な単語をツリーから取り除くことで、自動構成されるツリーの有用性が高まると考えられるので、分類に不要な単語の定義、およびそれらを取り除くための方法を検討することが課題に挙げられる。

謝辞 本研究は、平成 22 年度佐賀大学奨励研究費の支援を受けている。

参 考 文 献

[1] 掛下 哲郎, 園木 幸寶,「OLAP 操作を活用したファイル整理ツール HyperClassifier」, 第 8 回情報科学技術フォーラム(FIT 2009), 2009.

[2] 山口 章太,「ファイル整理ツール HyperClassifier における移行支援ツールの評価とその改良」, 平成 21 年度佐賀大学理工学部知能情報システム学科卒業論文

[3] 掛下 哲郎, 山口 章太, 「ファイル整理ツール HyperClassifier における多次元ツリー自動構成ツール」,平成 22 年度 電気関係学会九州支部連合大会, 08-2A-07, 2010

[4] MeCab, http://mecab.sourceforge.net/

[5] 八重樫, 白井, 「複数の視点から情報を管理するファイルブラウザの開発」, 情報処理学会 研究報告デジタルドキュメント(DD), 2004-DD-047, 2004.

[6] SAVVY/EWAP, http://www.info-brdg.co.jp/savvy/ewap.html

[7] 富沢 肇, 藤原 彰彦,「ドキュメント管理システム, ドキュメント管理方法, ドキュメント管理プログラム」, 特許公開 2009-110501