係り受け関係を考慮したテキストマイニングのための半構造マイニング手法の提案

Mining Semi-structure for Text with Dependency Structure

佐藤 一誠[1] 中川 裕志[2]

Issei SATO Hiroshi NAKAGAWA

近年，テキストマイニングにおいて，精密な知識をマイニングするために，1文内の単語の共起関係だけではなく，係り受け構造のような単語間の関係性を考慮することで，文が持つ意味構造を含めたマイニングの重要性が高まっている．本研究では，テキストデータから頻出な係り受け構造をもつ単語集合を抽出する手法を提案する．特に，複数のアイテムを節点とする木構造として係り受け構造を表現し，そのマイニング手法を提案することで，従来の半構造マイニング手法では抽出できないパターンの抽出を可能とした．

In text mining, when we need more precise information than word frequencies such as relationships between words, it is important to extract dependency structure in a sentence. This paper proposes a semi-structure mining method extracting frequent words with dependency structure in a large number of text data. Our method identifies dependency structure as tree structure whose node has multiple labels. In this way, our proposed method can extract patterns which the conventional method can not extract.

1. はじめに

近年，テキストマイニングにおいて，精密な知識をマイニングするために，1文内の単語の共起関係だけではなく，係り受け構造のような単語間の関係性を考慮することで，文が持つ意味構造を含めたマイニングの重要性が高まっている．一般に，係り受け構造は，文節単位で係ることから，文節を節点に持つ木構造で表現される．木構造で表現されるので，FREQT[1][3]のような従来の木構造マイニング手法が適用可能である．しかし，係り受け構造を木構造で表現する場合，次のような問題が生じる．例えば， 図1にある(a)(b)2つの木構造に共通する部分構造を抽出する場合を考えると，(c)のような部分構造が抽出される．(c)が抽出されたとしても，節点数が少ないことから，得られる情報量が少ないという問題が生じる．これは，文節を節点とした木構造の場合，助詞や表記上のぶれ(例では，「に」と「を」の違い，「首相」と「総理」の違い)などによって，ラベルの異なり数が増加することに起因する．よって，節点数の少ないパターンが抽出

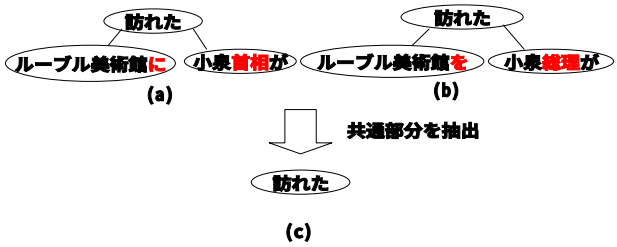

図 1 木構造表現による共通部分木の抽出例
Fig.1 Extracted Common Sub-Tree by Existing Structure

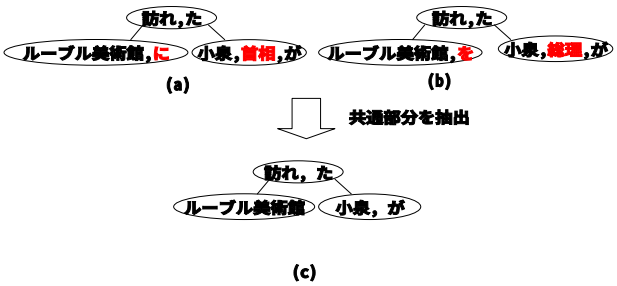

図 2 提案するデータ構造による共通部分構造の抽出例
Fig.2 Extracted Common Sub-Tree by Proposed Structure

され，抽出できない単語間の関係(例では，「ルーブル美術館」「小泉」「訪れる」などの関係)が存在してしまう．

本研究では，係り受け構造を表現する新しいデータ構造を提案し，そのデータ構造に対するマイニング手法を提案することで，上記の問題を解決する．

提案するデータ構造は，単語(形態素)を１つのアイテムとし，木構造の各節点に複数のアイテムを保持する半構造である．図2に具体例を示す．

各節点は，複数のアイテムを保持している．節点間の比較は，文節単位ではなく，形態素単位で，共通する部分構造を抽出する．これにより，図2の(a)(b)からは，(c)のような部分構造が抽出される．ここで重要なのは，節点数が多いパターンが抽出できることである．図1の抽出された部分木の節点数が１であるのに対し，図2では，節点数が3のパターンが抽出される．これにより図1と比べると，図2のように，情報量の多い頻出パターンが抽出されていると考えられる．

提案するデータ構造は，厳密には木構造ではないので，従来の木構造マイニングアルゴリズムが適用できないが，シーケンシャルパターンマイニングアルゴリズムPrefixSpanを改良することで，提案するデータ構造に対する半構造マイニングアルゴリズムを実現する．

2. シーケンシャルパターンマイニング[2][4]

2.1 用語定義

$I=\{i_1, i_2, \cdots, i_n\}$ をアイテムの集合とする．
このアイテムの集合を**エレメント**と呼ぶ．エレメントは，アイテムを'()'で囲むことで表現する．a,bをアイテムとすると，(a,b)などがエレメントである．エレメント中のアイテムの順番は考慮しないため，予めエレメント中のアイテムを辞書順にソートしておく必要がある．しかし，アイテムの元来持っている順序を考慮する必要がある場合，エレメント中のアイテムを辞書順にソートしないでおけば，エレメント中のアイテムの元々の順序を考慮した処理が可能である．

[1] 学生会員 東京大学大学院情報理工学系研究科
sato@r.dl.itc.u-tokyo.ac.jp
[2] 東京大学 情報基盤センター
nakagawa@dl.itc.u-tokyo.ac.jp

シーケンスとは，順序を保持するエレメントの列である．例えば，(a,b)(a,d) ≠(a,d)(a,b)である．
シーケンスsをs=<e_1 ,e_2 ,…,e_l>と表記する．
e_k(k=1,2,…,l)は任意のエレメントである．
例えば，<(a b) (a c) (b d) (a b)>がシーケンスである．
シーケンス中のアイテムの個数をシーケンスの長さとする．
あるシーケンスs_1 中のすべてのアイテムが，別のシーケンスs_2 中に存在し，その順序も保持している場合，s_1をs_2 のサブシーケンスと呼ぶ．s_1とs_2 の関係を $s_1 \subseteq s_2$ と表記する．

シーケンスデータベースSとは，シーケンスID(sid)とシーケンスsのタプル(sid,s)の集合である．

$$S=\{(sid_1,s_1),(sid_2,s_2),\cdots,(sid_n,s_n)\}$$

シーケンスα のシーケンスデータベースSにおける**サポート値**とは，S中のすべてのシーケンスのうち，シーケンスα を含むタプルの数である．

$$support_S(\alpha) = |\{(sid,s)|\ (sid,s)\in S \wedge \alpha \subseteq s\}|$$

2.2 シーケンシャルパターンマイニングの問題定義

シーケンシャルパターンマイニングとは，シーケンスデータベースSから，**最小サポート値**と呼ばれる任意の正の整数ζに対し，$support_S(\alpha) \geqq \zeta$となるような頻出シーケンスαを全て抽出する問題である．

2.3 PrefixSpan[4]

PrefixSpanは，シーケンシャルパターンマイニングの代表的なアルゴリズムである．**射影(Prefix-Projection)**という操作を，深さ優先で再帰的に行うことで、頻出シーケンスを効率的に抽出する．射影とは，射影するアイテムの**Postfix**を新たにシーケンスデータベースとする操作である．例えば，シーケンスs=<(c,d)(b)(a,d)(c,d)(b,c,a)>をaで射影すると，<(_d)(c,d)(b,c,a)>(= sにおけるaのPostfix)となる．つまり，シーケンス中に最初に現れるa以降のアイテム列をPostfixとし、このPostfixを新たにシーケンスデータベース(**射影データベース**と呼ぶ)とすることである．エレメントの内外にあるアイテムを区別するために'_'というPrefixをつける．なお，本論文では，PrefixSpanのアルゴリズムや射影やPostfixなどの正確な定義に関しては元の論文の参照を前提とする．

3. 提案手法

係り受け構造を，文節をラベルとする節点を保持する木構造で表したのでは，助詞や表記上のぶれにより，節点数の少ない部分木が抽出されてしまう．係り受け構造を表現する新しいデータ構造を提案し，そのデータ構造に対するマイニング手法を提案することで，上記の問題を解決する．

3.1 提案手法の概要

提案するデータ構造は，厳密には木構造ではないので，従来の木構造マイニングアルゴリズムが適用できない．

しかし，図 3 に示すように，深さ優先探索で反時計周りに前順走査で各節点を走査し，節点内はアイテムの順序に従って各アイテムを列挙し，index を割り当てれば，シーケンシャルパターンマイニングの適用が可能なデータ構造となる．エレメントは，節点に割り当てられたアイテム集合となる．ただし，各アイテムは，順序を保持していると考え，集合ではなく順序付きの列とする．各エレメントに，木構造における情報(親，長子，次弟)の情報(index)を保持させる．存在しない場合は，index を-1 とする．上記の操作によって得られるシーケンスを**準木構造シーケンス**と呼ぶことにする．なお，ある節点の「長子」とは，一般には複数存在するその子節点のうちの最左節点である．また，ある節点の「弟」とは兄弟節点のうち，自分の右側にある節点であり，「次弟」とは，弟のうちで最左，すなわち自分の隣接する弟である．図 3 の四角で囲まれた部分が，シーケンシャルパターンマイニングのデータ構造となっている．よってシーケンシャルパターンマイニングアルゴリズム PrefixSpan が適用可能となる．ただし，単純に適用したのでは，図 3 の(a,b,c)(c,b)のような非連結なパターン(木構造となっていないパターン)も抽出されてしまうため，制約が必要となる．

本提案手法は，上記のような変換されたデータ構造に対して，連結されたパターン(部分木)のみ抽出するための制約付きの射影(**Tree-Projection**)を行うアルゴリズムである．

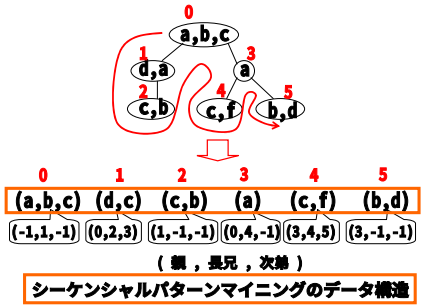

図 3 データ構造の変換例
Fig.3 Transformation of Data Structure

3.2 用語定義

射影したアイテムを積んでいくスタックを**「射影スタック」**と呼ぶ．あるアイテム i で射影したとすると，アイテム i は射影スタックに積まれる．アイテム i での射影が終了すると射影スタックから削除される．射影スタック中のアイテム集合を **Proj-Items** と表現する．この Proj-Items が抽出される頻出パターンとなる．

エレメント I と A の間に経路が存在し，エレメント I の方が A よりも深い位置にいるとする．I と A の経路上のエレメント数(節点数)が k-1 のエレメントであるとき，エレメント A をエレメント I の「k 代前の祖先」と呼ぶ．エレメント I の親は，1 代前の祖先となる．エレメント I の 0 代前の祖先をエレメント I 自身とする．例えば，図 3 の index=2 のエレメント(c,b)の 1 代前の祖先は(d,a)，2 代前の祖先は，(a,b,c)である．

3.3 Tree-Projection

準木構造シーケンス S のアイテム i による Tree-Projection とは，以下の制約を満たす射影である．

制約：射影データベース S<i>に含まれるアイテムは，アイテム i の Postfix の中で，射影するアイテム i または Proj-Items 中のアイテム n との間に枝を持つアイテム，つまり，連結するアイテムのみ射影データベースとすることを意味する．図 4 に具体例を示す．図 4 は，index=0 のエレメント(節点)中のアイテム a による Tree-Projection の動作例を示している．

Tree-Projection は，以下の 3 つの射影からなる．

- element-projection
- child-projection
- Level k sibling-projection

ただし, 各射影において, 次に射影するアイテムを選択し, Tree-Projection が再帰的に呼び出される.

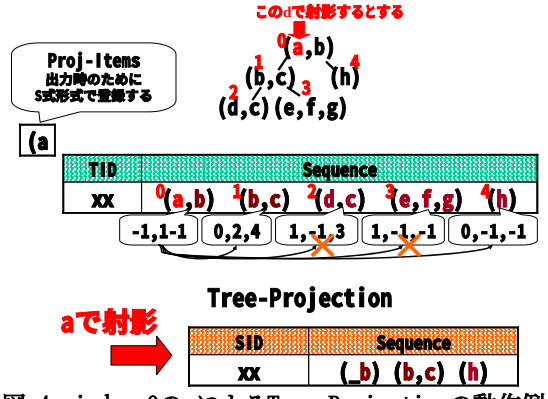

図 4 index=0のaによるTree-Projectionの動作例
Fig.4 Tree-Projection with a (index=0)

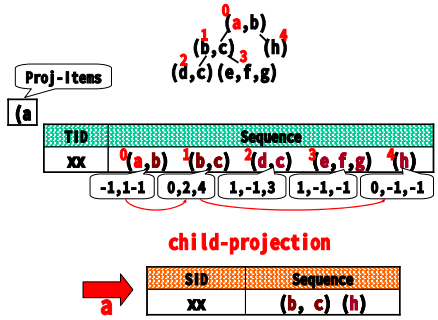

図 5 index=0のaによるchild-projectionの動作例
Fig.5 child-projection with a (index=0)

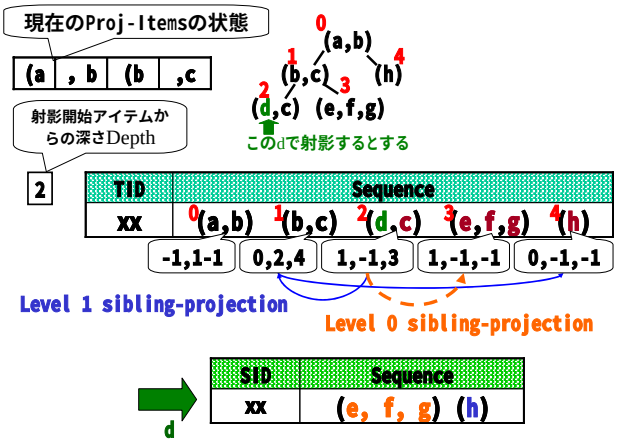

図 6 index=2のb によるsibling-projectionの動作例
Fig.6 sibling-projection with b (index=2)

射影したアイテムは, 射影スタックに積まれていく. 射影が終了すると, スタックから削除される.

以下, それぞれの射影を説明する.

■ element-projection

準木構造シーケンス S のアイテム i による element-projection とは, アイテム i と同一のエレメント(節点)内にあるアイテムのみ射影データベースに含める射影である. PrefixSpan では, '_'という prefix をつけることで, エレメント内外にあるアイテムを区別して抽出するが, このような射影を別に設けることでも実現できる.

■ child-projection

準木構造シーケンス S のアイテム i による child-projection とは, アイテム i の子となるアイテムのみ射影データベースに含める射影である.

具体的には, 以下の操作を行う.

1. アイテム i が属する節点の長兄である子節点の index を取得する.
2. 長兄の次弟の index を取得する.
3. 順次, 次弟の index を取得していく.

以上により, アイテム i と子の関係にあるアイテム集合を取得できる. これらのアイテムのみ射影データベースに含める. 動作例を, 図 5 に示す.

■ Level k sibling-projection

準木構造シーケンス S のアイテム i による Level k sibling-projection とは,

アイテム i の k 代前の祖先の弟となるエレメントのアイテムのみ射影データベースに含める射影である.

具体的には, 以下の操作を行う.

1. k 代前のエレメントの index を取得する.
2. k 代前の祖先の次弟の index を取得する.
3. 順次, 次弟の index を取得していく.

以上により, アイテム i の k 代前の祖先の弟となるアイテム集合を取得できる. これらのアイテムのみ射影データベースに含める. Proj-Items に 1 番最初に積まれたアイテムからの深さを **Depth** とすれば, 上記の操作を k=0 から Depth−1 まで繰り返す. 動作例を, 図 6 に示す.

4. 提案手法の評価

4.1 データセット

データセットは, (株)日本航空インターナショナルで収集されている航空安全レポート[3]を用いた. 1文[4]を単位とし CaboCha[5]を用いて係り受け解析を行った. 単一アイテムを節点とする木構造は, 従来の文節を節点とした係り受け木[6]とし, 複数のアイテムを節点とする木構造は, 文節をエレメント[7]とし, 単語(形態素)をアイテムとした[8].

4.2 評価実験

提案したデータ構造に対するマイニング手法が従来存在しないため, 提案手法の抽出速度は, 厳密には評価できない. しかし, 提案したデータ構造は, 木構造をサブセットとして含む. よって, 係り受け構造を、文節を節点とする木構造として表現したデータに対する速度評価を行う.

便宜上, 本提案手法を pFREQT(projection-based FREQT)と以下呼ぶことにする.

図7に, 文節を節点とする木構造データに対し, 木構造マイニングアルゴリズムFREQT[9][1][3]と提案したマイニングア

---

3 事前に名前等の個人情報は削除して, 個人が特定できないようにしてある

4 「。」で区切られた文を 1 文とする

5 http://chasen.org/~taku/software/cabocha/

6 図 1 参照

7 厳密には集合ではなく順序列

8 図 2 参照

9 http://chasen.org/taku/software/freqt/のものを評価に使用した

ルゴリズムpFREQTを適用させた場合の速度の比較を示す．最小サポート値を2とした．

図7からもわかるように，pFREQTは，FREQTよりも抽出時間が早い．これは，射影を節点間の関係によって分けているためパターンの探索時に，pFREQTのメモリ使用量がFREQTよりも少なくて済むからであると考えられる．

図8は，抽出されたパターンの節点数の統計量である．最小サポート値を3とし，節点が2以上のパターンを抽出した．データセット[10]を5つ作成して評価を行った．

文節を節点としてFREQTを適用させた場合と提案手法を適用させた場合で，抽出されたパターンの節点数の平均値，中央値，最大値の比較を行った．横軸は，各データセットであり，縦軸は，各々の節点数の統計量である．

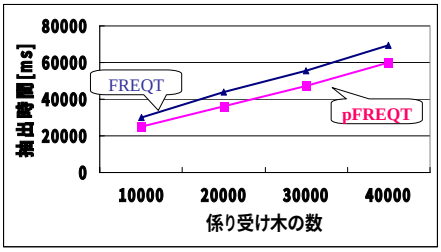

図 7 抽出時間の比較
Fig.7 Comparison of Extracting Time

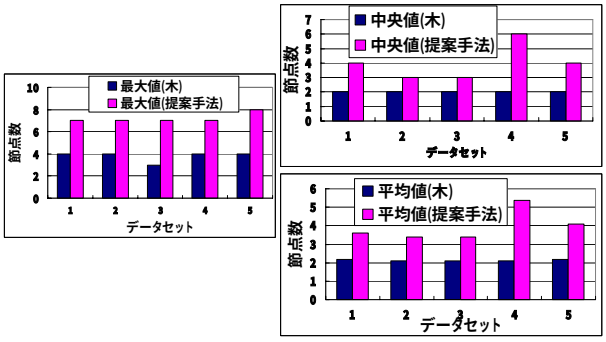

図 8 抽出した節点の統計量
Fig.8 Statistics on the Number of Extracted Node

図8により，本提案手法は，従来手法に比べて，抽出される部分木は，平均値，中央値，最大値すべてにおいて節点数の多い木である．より節点数の大きい部分木が抽出できれば，それだけ係り受け関係を持つ単語の関係を多く抽出できると言える．よって，本提案手法は，従来手法では抽出できないパターンを抽出可能であると言える．

ただし，こうして取り出された節点は多数にのぼるため，知識として意味のある節点を選択する問題に取り組む必要がある．これは知識マイニングの次の段階の処理であるので，本論文では扱わない．

5. 関連研究

工藤らによって，PrefixSpanを，順序木のマイニングへ拡張する研究が行われている[5][6]．工藤らの手法は，関係関数を導入して，PrefixSpanを順序木のマイニングに拡張している．しかし，抽出されるパターンに，連結された部分木以外に，非連結の部分木が抽出されてしまうという問題がある．また，射影時に，各アイテム毎に関係値という値を算出し付加するため，節点数の2乗に比例したメモリ容量を必要とする．本提案手法は，節点同士の関係に応じて分けて射影をしているため，このような関係値の付加は必要ない．また，本提案手法の最大のポイントは，複数のアイテムを節点とする木構造を定義し，そのマイニング手法を提案したことであるため，工藤らの研究とは大きく異なる．

6. まとめと今後の課題

係り受け構造を複数のアイテムを節点とする木構造で表現し，そのマイニングアルゴリズムを提案した．
これにより，従来手法では抽出できないパターンの抽出が可能であることを示した．今後は，他分野(XMLなど)への応用を模索する予定である．

[謝辞]

本研究を進めるにあたり，(株)日本航空インターナショナル 運行本部から提供していただいたデータを利用しました．同社の齋藤隆氏，寺田昭氏に深く感謝いたします．また，この研究は，文科省科学研究費 特定領域研究「情報爆発」の補助を得て行われた．

[文献]

[1] Kenji Abe, Shinji Kawasoe, Tatsuya Asai, Hiroki Arimura, Setsuo Arikawa : Optimized Substructure Discovery for Semi-structured Data, Proc. of PKDD 2002(2002)

[2] R.Agrawal and R.Srikant : Mining Sequential Patterns, In Proc. of ICDE1995, IEEE Press , pp.3-14(1995)

[3] Tatsuya Asai, Kenji Abe, Shinji Kawasoe, Hiroki Arimura, Hiroshi Sakamoto, Setsuo Arikawa : Efficient Substructure Discovery from Large Semi-structured Data : In Proc. of SDM 2002(2002)

[4] J.Pei, J.Han, B.Mortazavi-Asl, H.Pnto,Q.Chen, U.Dayal, and M.Hsu : PrefixSpan:Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, In Proc. of ICDE2001, IEEE Press , pp.215-224(2001)

[5] 工藤拓，山本薫，坪井祐太，松本裕治 ： 言語情報を利用したテキストマイニング， 情報研報(NL)，Vol.2002, No.020, pp.65-72(2002)

[6] ¥bibitem[工藤 02b]{kudo:02b}工藤 拓，山本 薫，坪井祐太，松本 裕治 ： テキストデータベースからの構文構造のマイニング， 情報研報(ICS)，Vol.2002,No.045, pp.139-14(2002)

佐藤 一誠 Issei SATO
東京大学大学院情報理工学系研究科在学中．半構造マイニングや確率的言語モデルの研究に従事．日本データベース学会学生会員．

中川 裕志 Hiroshi NAKAGAWA
1975 年東京大学工学部卒業．1980 年東京大学大学院修了．工学博士．同年より横浜国立大学工学部勤務．1999 年より東京大学 情報基盤センター教授．2003 年より東京大学情報理工学系研究科兼任．現在に至る．人工知能，自然言語処理，WWW の研究に従事．主要な公開ソフトとして，用語抽出システム：言選 Web，用例検索システム：Kiwi がある．

10 1 つのデータセットは 2000 個の係り受け木を持つ