係り受け構造の共起性に基づく日本語コロケーションの自動抽出

高山 宏規 †　　加藤 芳秀 ††　　大野 誠寛 †††　　松原 茂樹 †

† 名古屋大学大学院情報科学研究科
†† 名古屋大学情報連携統括本部
††† 名古屋大学情報基盤センター
takayama@db.ss.is.nagoya-u.ac.jp

1　はじめに

コロケーションとは，言語的，あるいは慣用的な結びつきをもつ単語の系列である．コロケーションを大量に収集し，辞書として整理すれば，自然言語処理における言語資源として，あるいは外国語を学習する際の教材として有用である．コロケーション辞典はいくつか出版されているが，それらに記載されている表現や用例の数は十分に多いとは言い難い．また，コロケーションには流行り廃りがあるため，辞書の定期的な更新が必要である．しかし，辞典の更新コストは非常に高い．

このような問題を解決するために，コーパスから大量にコロケーションを自動収集する方法が数多く提案されている．その多くは，2 単語の結合度を評価し，コロケーションを抽出する．そのため，バイグラムコロケーションしか抽出することができない．

本稿では，係り受け構造が付与されたコーパスから日本語コロケーションを自動抽出する手法を提案する．本手法の特徴は，3 単語以上のコロケーションを抽出できること，及び文中の離れた場所に単語が出現するコロケーションを抽出できることである．本手法は，Takayama らのコロケーション抽出手法 [4] を日本語に適用できるように修正したものである．

2　関連研究

本手法は，Takayama らの手法 [4] を日本語に適用できるように修正したものであるので，まず，Takayama らの手法について述べる．Takayama らは，依存構造に基づきコロケーションを自動抽出する手法を提案している．Takayama らの手法は，文を依存構造に基づく木構造として記述し，その木構造から木構造パターンを抽出することにより，依存関係で連結された単語列を獲得する．獲得された単語列に対して，自己相互情報量に基づき依存構造の結合度を評価し，コロケーションを抽出している．しかし，この手法を日本語に適用する場合，以下の様な問題点がある．

- 木構造として順序木を想定しているため，語順が比較的自由な日本語には適していない．
- 木構造のノードのラベルとして単語を与えているが，日本語においてどのような単位でラベルを与えればよいか不明である．

3　係り受け構造に基づく日本語コロケーションの自動抽出

本節では，係り受け構造が付与されたコーパスから日本語コロケーションを抽出する手法を提案する．本手法は，Takayama らの提案した結合度 [4] を用い，日本語コロケーションを抽出する．日本語の語順の自由さを扱うために，木構造として無順序木を対象とする．提案手法の概略を述べる．前提としてコーパスには係り受け構造が付与されているものとする．この係り受け構造は文ごとに一つの木構造を構成する．本手法では，まず，この木構造に頻出する木構造パターンを抽出する．この木構造パターンは係り受け関係で連結された単語列と対応している．次に，得られた木構造パターンを，Takayama らの提案した結合度により評価し，コロケーションを抽出する．

3.1　係り受け構造に基づく日本語の木構造の構築

文に対する係り受け構造は一つの木構造を構成するので，まず，その点から説明する．

係り受け構文解析器は CaboCha[6] や KNP[1] に代表されるように，文節毎に構文解析を行うものがほとんどである．文中の各文節をノードとし，係り先を親ノード，係り元を子ノードと定めると，文に対して一つの木構造が与えられる．木構造に含まれる木構造パターンを抽出すると，係り受け関係で連結された文節列が得られる．文節単位ではコロケーションを抽出する際の粒度としては粗い．例えば，「問題　として ...

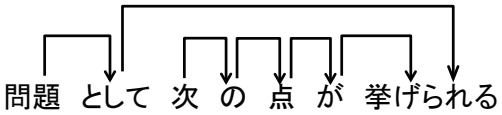

図 1: 変更後の係り受け関係の例

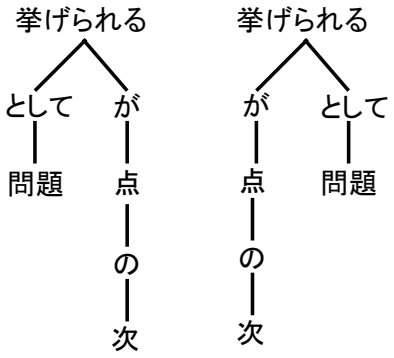

図 2: 二つの文に対する木構造

が　挙げられる」といったようなコロケーションを抽出することができない．以上のことを踏まえ，本手法では，以下の様な変更を加える（図 1 に変更後の係り受け関係の例を示す）．

- 文節 B が助詞を含むとき，助詞の部分 (A) とそれ以外の部分 (C) に分割する．
- C の係り先は A とする．
- A の係り先は B の係り先とする．
- 係り先が分割されていた場合は，助詞以外の部分を係り先とする．

また，英語は語順に厳しい制限があるのに対し，日本語の語順は比較的自由である．例えば,「問題として次の点が挙げられる」と「次の点が問題として挙げられる」の二つの文は語順は異なるが，同じ意味を持つ．この二つの木構造を図 2 に示す．

この木構造を Takayama らの手法 [4] と同様に，順序木としてとらえると，二つの木構造は異なるものとして扱われる．この二つの木構造は「として」と「が」のノードの兄弟関係が逆になっている点だけが異なる．兄弟関係のない無順序木として二つの木構造をとらえると，二つの木構造の木構造は同じものを表す．そのため，以下，日本語の係り受け構造を無順序木として考える．

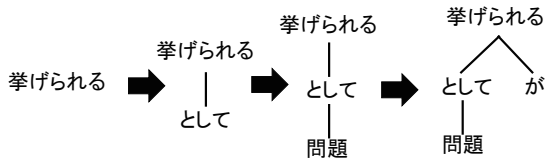

図 3: 拡張の例

3.2　木構造パターンの抽出

係り受け関係で連結されたコロケーションを得るため，本手法では，コーパスの文を前節で述べた木構造で記述し，この木構造から木構造パターンを抽出する．本手法では，効率的にパターンを抽出するために，従来手法と同様に，木構造パターンの抽出において閾値を設定し，出現頻度が閾値以下のパターンを抽出しない．これにより，効率的にパターンを抽出する．

木構造パターンの抽出は，Takayama らの手法では，ツリーマイニングアルゴリズムである FREQT[2] を使用して，木構造パターンの抽出を行っている．FREQT は順序木集合から，木構造パターンを抽出している．しかし，前節で述べたように，本手法では無順序木を扱うため，無順序木に対するツリーマイニングアルゴリズムである UNOT[3] をベースとして木構造パターンを抽出する．UNOT は木構造集合から，ある閾値以上出現する無順序木パターンを抽出する手法である．UNOT では，ラベル集合から任意のラベルを一つ選択し，サイズが 1，つまり単一のノードからなるパターンを生成する．出現頻度があらかじめ定められた閾値以上のものであれば，次の操作に進む．次に，サイズが 1 のパターンにノードを一つ付加することにより，サイズが 2 の頻出パターン候補を列挙する．頻出パターンにノードを付加する操作を拡張と呼ぶ．拡張は正規形表現と呼ばれる構造を維持するように制限されるが，これにより，無順序木パターンを重複なく効率的に列挙することができる（図 3 に拡張の例を示す）．拡張によりサイズが 2 のパターンの候補が得られるが，これらの候補のうち，出現頻度が閾値以上のパターンのみを頻出パターンとして残す．以降，サイズを 1 ずつ増やしながら拡張を行い，頻出パターンが得られなくなるまで繰り返す．これにより，出現頻度が閾値以上の無順序木パターンを効率的に得ることができる．UNOT では，パターンの全てのノードに対応するコーパス中のノードを漸増的に計算しており，これに基づき出現頻度を計算している．

3.3　無順序木の結合度の評価

UNOT により，木構造集合において閾値以上の頻度で出現するパターンを得ることができる．これらのパ

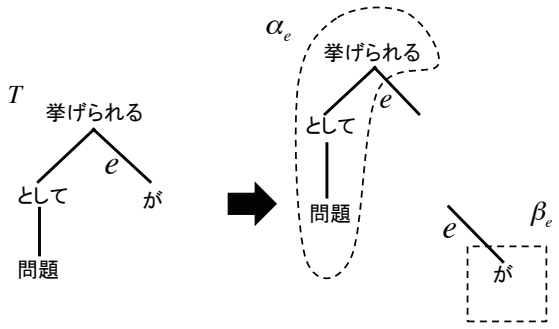

図 4: 分割の例

ターンのうち，対応する単語列がコロケーションであるか否かを評価する必要がある．コロケーションの評価尺度として，Takayama らが提案した結合度 [4] を用いる．Takayama らの提案した結合度は，依存構造を分割して自己相互情報量を計算し，その自己相互情報量の平均をとっている．

最初に係り受け構造の分割について説明する．係り受け構造を分割するには，係り受け構造中のエッジを一つ選択し，親ノードに連結した部分と，子ノードに連結した部分に分ければよい．以下では，e をエッジとするとき，親ノードに連結した部分を α_e，子ノードに連結した部分を β_e と書く．図 4 に係り受け構造の分割の例を示す．

次に，係り受け構造に対する結合度の計算について説明する．基本的には，Takayama らの提案した結合度と同じである．係り受け構造をエッジ e で分割したときの α_e と β_e の自己相互情報量は以下の式のようになる．

$$PMI(\alpha_e, \beta_e) = \log \frac{f(\alpha_e, \beta_e)N}{f(\alpha_e)f(\beta_e)} \quad (1)$$

ここで，N は係り受け構造コーパス中のエッジの総数である．$f(\cdot)$ は出現頻度を表す．$f(\alpha_e, \beta_e) = f(T)$ であるので，UNOT の結果をそのまま利用できる．$f(\alpha_e)$，及び $f(\beta_e)$ については別途求める必要があるが，これについては次節で述べる．

最後に，この係り受け構造の結合度は，係り受け構造 T のエッジ集合 E とするとき，以下のように定義されている．

$$Assoc(T) = \frac{1}{|E|} \sum_{e \in E} PMI(\alpha_e, \beta_e) \quad (2)$$

3.3.1 依存構造パターンとエッジの共起頻度

前節で述べた結合度を計算するためには $f(\alpha_e)$，及び $f(\beta_e)$ を求める必要がある．これを求めるために，Takayama らの手法と同様に，特殊な記号＊を導入し，$f(\alpha_e)$ を求める．そのために，UNOT に変更を加える．この記号は任意の単語にマッチすることを表している．パターンを拡張するとき，＊をラベルにもつノードを付加することにより，$f(\alpha_e)$ を計算することができる．UNOT では，無順序木パターンを重複なく効率的に列挙するために，ラベルの大小関係を定義する必要がある．本手法では，＊を任意のラベルより小さいラベルと定義する．＊を使用した拡張を無制限に行うと効率的ではないため，Takayama らの手法と同様に，以下のような制約を設ける．

- ＊をラベルに持つノードに対して，いかなるノードも付加しない．
- パターンが＊をラベルに持つノードを含むとき，＊をラベルに持つノードを付加しない．

次に，$f(\beta_e)$ について述べる．UNOT では，パターンの全てのノードに対応するコーパス中のノードを全て保持している．したがって，β_e のルートに対応するコーパス中のノードが係り受け構造のルートかどうかをチェックすれば，$f(\beta_e)$ を求めることができる

3.3.2 コロケーションの抽出

本節では，結合度を用いてコロケーションを抽出する手法について述べる．基本的には，結合度の高いパターンを選択し，それらを単語列に復元するが，Takayama らの手法と同様の，極大性と閉包性の制約を導入する．

- すべての $T' \in Sub(T) \cup Super(T)$ に対して，$Assoc(T) > Assoc(T')$.
- すべての $T' \in Super(T)$ に対して，$f(T) - f(T') > m$.

ここで，$Sub(T)$ は T から葉ノード，あるいは根ノードを一つ取り除いたパターン集合，$Super(T)$ は T の任意のノードに子ノードを追加したパターン，あるいはルートに親ノードを追加したパターンの集合である．また，m は閾値を表す．

4 実験

提案手法の有効性を確認するために，実験を実施した．実験には，人工知能学会全国大会論文集 2005 年から 2012 年までの論文から取り出した 241909 文 [5] を使用し，提案手法によりコロケーションを抽出した．係り受け構造は CaboCha[6] により与えた．また，句読点はすべて取り除いた．出現頻度の閾値を 10，閉包性の制約パラメータを $m = 1$ とした．

3 つ以上の単語から構成され，文中において単語が離れて出現しているパターンのうち，結合度が上位

表 1: 提案手法により抽出された単語列

「りんご が … あります
予定 の … 実施率
2 章で は … 述べる
4 章で は … 述べる
3 章で は … 述べる
出演者 による 推薦 … による 推薦
5 章 で … 述べる
必要十分条件 は … ことである
時点 で … 終了する
筆者ら が … 進めている
重要なの は … ことである
詳細 は … で 述べる
図 1 に … 概念図 を 示す
今後 は … 検討していく
選択する と … 表示される
行為 と … 行為
このよう に … 可能 と なる
また … 研究 も 行われている
今後 は … 予定である
今後 は … 検討したい

表 2: Takayama らの手法と重複していない単語列

出演者 による 推薦 … による 推薦
また … 研究 も 行われている
今後 は … 予定である
から ランダム に … 選択する
では … 概要 を 述べる
では … 提案 を 行った
について 説明し … について 述べる
最後 に … 述べる
ここ で … と 仮定する
最後 に … まとめる
利用する ことで … 可能 と なる
では … 手法 を 提案した
実験 には … が 参加した
また … 問題 も ある
から … 可能性 が 示唆された
では … 効果 を 検証する
では … 抽出 を 試みた
これ により … 可能 と なる
では … 手法 を 提案する
我々 は … と 考える

20 であったものを表 1 に示す．また，Takayama らの手法も同様に実験を行い，本手法と同一の単語列が抽出されているかを調べた．本手法により抽出された単語列のうち，長さが 3 単語以上かつ，単語同士が離れて出現する単語列は 1269 個であった．一方，Takayama らの手法は 3680 個であった．2 つの手法が共通して抽出している単語列の数は 577 個であった．また，Takayama らの手法が抽出した単語列と重複していない単語列のうち，結合度が上位 20 までのものを表 2 に示す．Takayama らの手法と重複していない単語列は，本手法の抽出数の半数以上あり，抽出結果も論文でよく使われている言い回しが抽出できたことは，本手法が，日本語コロケーションの抽出に有効であることを示している．

5 おわりに

本稿では，係り受け構造に基づき日本語コロケーションを抽出する手法を提案した．本手法では，日本語文を係り受け構造に基づく無順序木として記述し，その無順序木から無順序木パターンを抽出することにより，係り受け関係で連結された単語列を獲得する．獲得された単語列に対して，Takayama らの提案した結合度を用いて評価し，コロケーションを抽出する．

今後，提案した手法の詳細な評価をする予定である．今回の実験では，論文データをコーパスとして用いたが，語順の入れ替えがより頻繁に生じるデータを用いた実験を行い，本手法の性質について詳細に検討したい．

参考文献

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?knp.

[2] Tatsuya Asai, Kenji Abe, Shinji Kawasoe, Hiroki Arimura, Hiroshi Sakamoto, and Setsuo Arikawa. Efficient Substructure Discovery from Large Semi-Structured Data. *Proc. of 2nd SIAM Inter. Conf. on Data Mining*, pp. 158–174, 2002.

[3] T.Uno T. Asai, H.Arimura and S.ichi Nakano. Discovering frequent substructures in large unordered trees. *In Discovery Science, volume 2843 of Lecture Notes in Artificial Intelligence*, pp. 47–61, 2003.

[4] Hiroki Takayama, Yoshihide Kato, Tomohiro Ohno, Shigeki Matsubara, and Yoshiharu Ishikawa. Collocation Extraction Using a PMI-Based Association Measure for Dependency Tree Patterns. *The Tenth Symposium on Natural Language Processing*, pp. 138–143, 2013.

[5] 馮思萌, 井上慧, 松原茂樹, 長尾確. 語の共起頻度の提示と用例文の検索に基づく論文執筆支援システム. 情報処理学会第 76 回全国大会, 2014. （掲載予定）.

[6] 工藤拓, 松本裕治. チャンキングの段階適用による日本語係り受け解析. Vol. 43, No. 6, pp. 1834–1842, 2002.