自然言語処理 121－11
情報学基礎 47－3
（1997. 9. 11）

単語の出現位置情報を用いたコーパスからのコロケーションの自動抽出

小田 裕樹　北 研二

徳島大学 工学部

コロケーションは単語間の共起情報を与える言語学的に重要な知識源であり，機械翻訳をはじめとする自然言語処理において，重要な意味をもっている．本稿では，コーパスからコロケーションを自動的に抽出する新しい手法を提案する．提案する手法では，コーパス中の各単語の位置情報を用いて，任意の文中のコロケーションを連続型・不連続型の別に抽出する．また，提案した自動抽出法を用いて，ATR 対話コーパスからコロケーションを抽出する実験を行った．本実験で得られた結果は，連続型・不連続型コロケーションともに重要な表現が抽出されており，提案した抽出法の有効性を示すことができた．

Automatically Extracting Collocations Using Words Position Information in Corpora

Hiroki ODA　Kenji KITA

Faculty of Engineering, Tokushima University

Collocations, which are cohesive and recurrent word clusters, play an important role in many natural language application systems as well as in linguistics. In this paper, we present a set of new techniques for automatically identifying or extracting collocations from corpora. These techniques produce a wide range of collocations, including continuous or discontinuous collocations, and are based on words position information. The effectiveness has been confirmed by evaluation experiments using the ADD (ATR Dialogue Database) corpus.

1　はじめに

コロケーション (Collocation) は単語間の共起情報を与える言語学的に重要な知識源であり，機械翻訳をはじめとする自然言語処理において，重要な意味をもっている．コロケーションは言語的あるいは慣用的な表現であることから，様々な形態が考えられる．その例として，“*Thank you very much*” や “*I would like to*” のような単語が連続している表現と，“*not only ～ but (also) ～*” や “*not so much ～ as ～*” のように単語間にギャップを持つ不連続な表現が存在する．これらの表現はそれを一つのまとまった単位として処理する必要があり，その知識は機械翻訳への適用を主として，音声・文字認識において認識結果を補正する手段とする方法 [1] や，第二外国語を学習する際の手助けとするような言語学習や言語教育の分野にも適用できる [2, 3].

以上のように，コロケーションの収集・整理は言語学的にも機械処理の面からも有益であるため，その収集の仕方は自然言語処理における重要な課題である．しかし，人手による収集では膨大な時間と手間が必要となり，かつコロケーションの定義が曖昧であるためにその網羅性・一貫性にも問題が生じる．

これらの点から，コロケーションを自動的に抽出・収集する方法として，相互情報量を用いた方法 [4]，仕事量基準を用いた方法 [5, 6]，n-gram を用いた方法 [7]，2 つの単語の位置関係の分布を考慮する方法 [8] をはじめとして様々な方法が提案されている．しかし，従来の方法の多くは連続したコロケーションを抽出の対象としており，不連続なコロケーション

の抽出に関する研究はごく少数であった [1, 9].

本稿では，単語の位置情報に基づき，連続型および不連続型の二種類のコロケーションをコーパスから自動的に抽出する方法を提案する．提案する手法は，コーパス全体からコロケーションを抽出するだけではなく，指定された任意の範囲 (たとえば，何番目の文，または何番目から何番目の文の中) にあるコロケーションを同定することができる．また，言語に依存しない (言語独立の) 方法であり，機械翻訳等への様々な活用が期待できる．

2 コロケーションの自動抽出の考え方と特徴

本稿で自動抽出の対象とするコロケーションは，文法的，意味的にまとまりのある単語の組み合わせであるため，コーパスを文字列ではなく単語列として扱う．また，コロケーションは同一文中に共起する単語の組み合わせであり，複数文にまたがるようなものはないとする．

2.1 連続型および不連続型コロケーション

コロケーションを，以下のように，形態的に “連続型” と “不連続型” の二種類に分類する．

- 連続型コロケーションとは連続する単語の組み合わせ (単語列) である．
- 不連続型コロケーションとは単語 (列) が離れて共起する組み合わせである．

また，コロケーションはそれ自身が意味的なまとまりであるので，一度，コロケーションとみなされた単語の組み合わせが，それ以後に分割されて処理されることは避ける必要がある．そこで，池原ら [9] の考えを応用して，次のように処理を行う．

1. 頻繁に出現するパターンとして，単語列 (連続型コロケーション) と，常に決まった単語を隣接して伴わない単語を抽出する．
2. 上記 1 の単語列または単語がギャップを持って (離れて) 共起する組み合わせが不連続型コロケーションである．

以上のように，連続型コロケーションと不連続型コロケーションを別々に抽出して集計する．この集計方法は，不連続型コロケーションが連続型コロケーションに比べて，その存在数 (出現回数) が少ないことからも有効である．

2.2 単語の出現位置情報

まず，本手法の基礎となる，単語の “位置情報” について説明する．コーパス中のある特定の場所にある単語 w の位置は，2 項組 (i, j) によって表される．ここで，i は文番号 (コーパス中の何番目の文であるか) を，また j は単語番号 (文中の何番目の単語であるか) を示している．単語 w はコーパス中の複数箇所に出現しえるので，このような 2 項組のリストを考えることにより，単語 w がコーパス中のどこに出現するかを決めることができる．以下では，2 項組 (i, j) のリストのことを，単語 w の出現位置表と呼ぶことにする．

一つ注意すべきことは，ある単語 w に対し，必ずしも出現位置表を一つだけ考える必要はないという点である．もし，単語 w が異なったコンテキストで用いられていれば，コンテキストごとに単語 w の出現位置表を用意してもよい．従って，同じ単語 w であっても，異なった出現位置表を持つ場合がある．本稿では，単語 w の出現するコンテキストをとらえるために，有限オートマトンを用いる．まず，コーパス中のすべての文を受理するような有限オートマトンを構成する．このような有限オートマトンにおいて，単語 w による状態遷移はオートマトン中の複数箇所に現れる可能性があるので，各状態遷移ごとに単語 w の出現位置表を作成する．なお，本稿では，コーパスから有限オートマトンを構成するために，ALERGIA アルゴリズム [10] を用いる．

2.3 基本的な考え方

コロケーションの自動抽出の基本的な考え方について説明する．コロケーションの自動抽出において，最も重要視すべきことは単語の組み合わせの出現頻度である．また，同一文中に共起する単語の組み合わせを抽出することから，各々の文に対する処理を行えばよい．そこで，“ある文中の単語の組み合わせが他の文にも出現するものをその文に含まれるコロケーションとする” という考えが，本稿の自動抽出

法の基本である．上で述べた単語の出現位置表により，単語間の位置関係をとらえることができるので，ある単語がコーパス中の複数の文に出現している場合，その各々の文の中での組み合わせを考慮するだけで，複数の文に対して同じ組み合わせで出現しているものを知ることができる．

本手法の処理の特徴を次に示す．

1. 各単語間の距離値そのものを扱うのではなく，各単語の関係を“連続”か“不連続”かのみとして考える．
2. 意味的にまとまりのない断片的な単語列の抽出を防ぐために，池原ら[9]のように最長一致の原則により，長い表現を優先して抽出する．
3. 多分木を構築することで一度の処理で，すべての任意の長さの不連続型コロケーションを抽出する．
4. コロケーションの抽出は，各単語のコーパス中の位置情報(整数)の比較のみで行うことができるため，計算的な負荷は非常に小さい．
5. コーパス全体に含まれるコロケーションだけではなく，任意の文(またはコーパス中の任意の範囲内の文)に含まれるコロケーションを知りたい場合にも適する．
6. 処理するコーパスを大きくしたい場合，位置情報を追加または新規作成するだけでよい．

3 単語位置情報に基づく自動抽出アルゴリズム

前節で述べた考えに基づいた，コロケーションの自動抽出アルゴリズムを提案する．まずコーパスを$W = s_0 s_1 \ldots s_N$と表す．s_nはコーパス中のn番目の文であり，$N+1$がコーパス中の文の総数である．また，$s_n = w_n^0 \ldots w_n^{T(n)}$は$s_n$が$T(n)+1$単語の単語列であることを示す．

(1) 単語の出現位置表の作成

ALERGIAアルゴリズム[10]を用いて，コーパス$W = \{s_n; 0 \leq n \leq N\}$を受理する決定性確率有限オートマトンを構成する．決定性確率有限オートマトンでは，各文s_nに対して状態遷移が一意に定まる．各状態遷移に対して，単語の出現位置表を作成する(図1参照)．なお，出現位置表とは各単語の

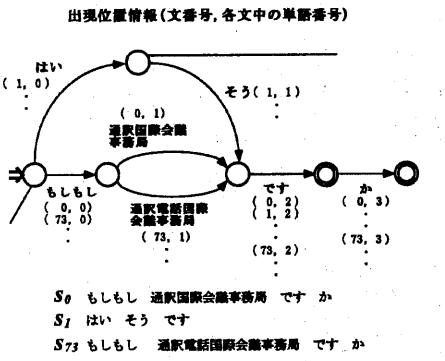

図 1: コーパスW中の各単語の出現位置の記録例

コーパス中の出現位置をまとめたものであり，出現位置は文番号と単語番号の2項組である．位置情報は$n = 0, 1, \ldots N$，$t(n) = 0, \ldots T(n)$の順に記録する．図2は，図1の処理によって作成された(W中の)単語の出現位置表を示している．各表は文s_0の単語“もしもし”，“通訳国際会議事務局”，“です”，“か”に対応している．

また，ALERGIAアルゴリズムの等価判定の信頼範囲によって，構築されるオートマトンの規模は変化する．様々な信頼範囲で構築されるオートマトンによって，上の手順で作成される出現位置表の総数(= 状態遷移の総数)と各表の内容が異なる．各表に記録される出現位置の2項組の数は状態遷移を行った回数である．

(2) 単語共有表の作成

単語の出現位置表から，文s_nの単語共有表を作成する(図3参照)．単語共有表とは，文s_n中のすべての単語$w_n^0, \ldots w_n^{T(n)}$(の状態遷移)の出現位置表を文番号の小さい順にマージして一つの表にまとめたものである．なお，図3において“位置番号”とあるものは，マージする前に，各単語の出現位置表に文s_nの単語番号$t(n)$を付けて，単語共有表の情報として付加したものである．つまり，単語共有表の中の情報は“出現する文の番号m”，“文s_m中の位置”，“文s_n中の位置”の3項組となる．ただし，次のものは

もしもし

文番号	単語番号
0	0
73	0
116	0
169	0
236	0
270	0
353	0
429	0
484	0
512	0
⋮	

です

文番号	単語番号
0	2
1	2
2	5
4	2
7	8
13	9
14	8
15	3
16	22
17	4
18	9
19	9
22	10
23	27
31	6
33	6
33	12
⋮	
73	2
⋮	

か

文番号	単語番号
0	3
3	9
6	19
10	2
12	7
14	9
15	4
18	10
21	33
22	19
27	15
31	7
33	13
⋮	
73	3
⋮	

通訳国際会議事務局

文番号	単語番号
0	1

図 2: 単語の出現位置表の例 (W中の文 s_0の単語に関する)

単語共有表

文番号	単語番号	位置番号
14	8	2
14	9	3
15	3	2
15	4	3
18	9	2
18	10	3
31	6	2
31	7	3
33	6	2
33	12	2
33	13	3
⋮		
73	0	0
73	2	2
73	3	3
⋮		

下記のものは単語共有表には含めない

文番号	単語番号	位置番号
0	0	0
0	1	1
0	2	2
0	3	3
1	2	2
2	5	2
3	9	3
4	2	2
6	19	3
7	8	2
10	2	3
12	7	3
13	9	2
16	22	2
17	4	2
19	9	2
21	33	3
22	10	2
23	27	2

図 3: コーパス W中の文 s_0の単語共有表 (左表) と単語共有表には含めない出現位置 (右表)

単語共有表に含めない.

1. 文番号が n であるもの.
2. マージしたときにその文番号を持つ単語が一つしかない文番号. たとえば, 図2の出現位置表において, 文番号1や2が該当する.

(3) 二文中に共通して出現するパターンの抽出

手順 (2) で作成した文 s_nの単語共有表を, 文番号 m の同じものごとに分割する. 分割後の表に対して, 次の条件により, 文 s_nと文 s_mで共に出現するパターンを抽出する.

条件 1: ある単語と別の単語の位置番号 (文 s_n中の単語の位置) が連続であり, かつ単語番号 (文 s_m中の単語の位置) が連続な場合は, それらの単語列を一つの出現パターンとして扱う.

条件 2: 条件 1 を満たさない単語はそれ自身を単独で一つの出現パターンとする.

上記の条件により得られる出現パターンはコーパス中に最低でも 2 回出現するために, 文 s_nに含まれるコロケーションとなる可能性がある. 図4に, 図3の単語共有表から抽出される出現パターンの例を示す.

(4) 単語パターンの収集

次の条件により, 手順 (3) で抽出されたすべての出現パターンから重複をなくし, 文 s_nの単語パターンを収集する. ここで単語パターンとは, 文 s_n中の単語 (列) が一つのまとまりとして他の文にも出現するものである.

条件: 出現パターンの位置番号 (文 s_n中の単語の位置) の組み合わせが一致するものは, 文 s_nにおける同一の単語パターンである.

出現パターン a の最小の位置番号を $\mathcal{P}_{min}(a)$, 最大の位置番号を $\mathcal{P}_{max}(a)$ とすると, 出現パターン a と出現パターン b は $\mathcal{P}_{min}(a) = \mathcal{P}_{min}(b)$ かつ $\mathcal{P}_{max}(a) = \mathcal{P}_{max}(b)$ の場合, 同じパターンと判定される. たとえば, 図4では, パターン (a), (c), (e) は, 文 s_0の同じ単語パターンとなる.

上の条件により, すべての文 s_mとの出現パターンを収集することで得られた文 s_nの単語パターンの集合を P_{s_n}とする. また, 文 s_nの単語パターンには位置番号のみを情報として持たせる.

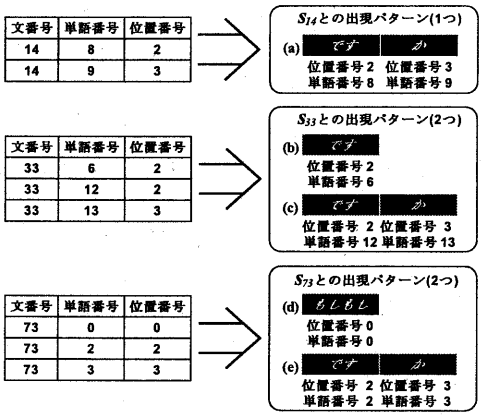

図 4: コーパス W 中の文 s_0 から抽出された出現パターンの例

(5) 部分単語パターンの削除と連続型コロケーションの抽出

手順 (4) の文 s_n の単語パターンの集合 P_{s_n} の中から，次の条件により，コロケーションの抽出時に問題となる“意味的にまとまりのあるパターンに包含される断片的なパターン”を削除する．なお，以下の条件において，抽出された場所とは文 s_n での各単語パターンの位置で知ることができる．

条件: 文 s_n 中で，ある単語パターンが抽出された場所からは，その部分単語パターンを抽出しない．

上の条件を満たすために問題となるのは，文 s_n の単語パターンが複数個抽出されたときに，単語パターン間で同じ位置番号の単語が重複する場合である．その場合，もしある単語パターン a の位置番号 (列) が最大値・最小値の比較によって他の単語パターン b の位置番号列に包含される ($\mathcal{P}_{min}(b) \leq \mathcal{P}_{min}(a)$ かつ $\mathcal{P}_{max}(a) \leq \mathcal{P}_{max}(b)$) ならば a を削除する．図5に簡単な例を示す．P_{s_n} 中で削除されずに残された単語パターンの集合を P'_{s_n} とし，P'_{s_n} 中の単語列を文 s_n 中の連続型コロケーションとして抽出する．たとえば，コーパス W の文 s_0 の P'_{s_n} からは“です か”が抽出される．また，P'_{s_n} 中の単語パターンは，文 s_n 中の不連続型コロケーションの要素となる．

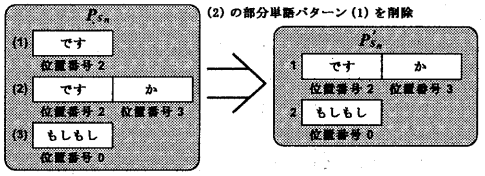

図 5: コーパス W 中の文 s_0 の単語部分パターンの削除例

(6) 不連続型コロケーションの抽出

P'_{s_n} 中の単語パターンはコーパス中の他の文にも出現するパターンであり，文 s_n における単語の意味的なまとまりと考えることができる．不連続型コロケーションとは，文 s_n の単語パターンが文 s_n の中と同じ組み合わせ (同順) で他の文中でも共起するものである．

まず，文 s_n の単語共有表の中から抽出される出現パターンが，P'_{s_n} 中の単語パターンと一致する文 $s_{m'}$ を探す．そして，文 $s_{m'}$ との間で複数の (P'_{s_n} 中の) 単語パターンが同じ順で出現する組み合わせを持つ場合，その単語パターンの組み合わせを文 s_n に含まれる不連続型コロケーションとする．以下で詳細な処理の説明を述べる．

(6.1) 要素の認定

出現パターンの抽出処理は手順 (3) と同様である．ただし，ここでは，抽出された文 $s_m (m \neq n)$ との間の出現パターンから，次の条件を満たさないパターンを削除する．

条件 1: P'_{s_n} 中のある単語パターンと位置番号 (列) が一致するパターンであること．

条件 2: 条件 1 を満たさない出現パターンをすべて削除した後，出現パターンが，“位置番号”と“単語番号”ともに不連続な関係が成立する他の出現パターンを持つものであること．

条件 2 の不連続な関係は，条件 1 を満たす出現パターンが複数ある場合のみ，次の条件により判定する．ここで，条件 2 の判定の対象となる出現パターンを a，比較対象の他の出現パターンを b とする．また，$\mathcal{W}_{min}(a)$ を a の最小単語番号，$\mathcal{W}_{max}(a)$ を a の最大単語番号とする．

不連続条件 1:

$\mathcal{P}_{min}(a) - 1 > \mathcal{P}_{max}(b)$ かつ $\mathcal{W}_{min}(a) - 1 > \mathcal{W}_{max}(b)$

不連続条件 2:

$\mathcal{P}_{max}(a) + 1 < \mathcal{P}_{min}(b)$ かつ $\mathcal{W}_{max}(a) + 1 < \mathcal{W}_{min}(b)$

不連続条件 1 または不連続条件 2 の一方を満たす場合，a と b の間に位置番号と単語番号とで共に不連続な関係が成立する．二つの不連続条件を用いて条件 2 を満たさない出現パターンを削除する．

条件 1 と条件 2 をすべて満たす出現パターン (P'_{s_n} 中の単語パターン) が，文 s_n に含まれる不連続型コロケーションの要素となる．つまり，そのパターンを共有する他の文 ($s_{m''}$ とする) と，文 s_n には同じ不連続な単語の組み合わせが出現する．

(6.2) 多分木の構築

手順 (6.1) で抽出された出現パターンのうちの一つを a としたときに，不連続条件 1 を満たす b を持たない出現パターンが，文 s_n と文 $s_{m''}$ との間で共有する不連続型コロケーションの先頭の要素である．先頭の要素の出現パターンを各々別の木構造の根として多分木を構築する．各々の根に対してつなぐことのできる出現パターンは，根を a としたときに不連続条件 2 が成立する出現パターン b のみである．ここで，b は根 a の子孫である．つまり，木構造の各々の節の出現パターンを a としたときに，

- 不連続条件 2 を満たす出現パターン b が節 a の子孫である．
- あるいは，不連続条件 1 を満たす出現パターン b が節 a の祖先である．

という条件を満たし，木構造が根のみである場合は単純に根の子として直接つなげればよい．しかし，既に根以外の節がある場合は，次の条件を，1，2，3 の順に適用して，多分木につなげていく．

1. ある節 A の子孫であり，かつ節 A が子パターンを持っていない場合，節 A の子としてつなげる (追加)．
2. ある節 A の子孫であり，かつ節 A の子の祖先でもある場合，節 A と節 A の子の間に挿入する．
3. ある節 A の子孫であり，かつ節 A の子すべてに対して子孫でも祖先でもない場合，節 A の子としてつなげる (追加)．

以上の処理に基づいて，すべての出現パターンを各々多分木につなげる．構成されたすべての多分木を根からすべての葉までたどった各々の出現パターン (P'_{s_n} 中の単語パターン) の組み合わせが，文 s_n と文 $s_{m''}$ の間で共有する文 s_n 中の不連続型コロケーションである．たとえば，W の文 s_0 は文 s_{73} との間から不連続型コロケーション“もしもし ～ です か”が抽出される．

(7) 不連続型コロケーションの収集

手順 (6) をすべての文 $s_{m''}$ に対して行い，抽出される文 s_n と文 $s_{m''}$ で共有する不連続型コロケーション (P'_{s_n} 中の単語パターンの組み合わせ) から，文 s_n に含まれる不連続型コロケーションを重複しないように収集する．

まず，抽出された各々の不連続型コロケーション α の先頭の要素である単語パターンと，不連続型コロケーション $\beta(\neq \alpha)$ の先頭の要素 (単語パターン) を位置番号列の “一致” か “不一致” かにより比較する．そして，一致した場合は，次の要素同士を比較するという処理を繰り返す．すべての単語パターンが一致すれば，α と β を文 s_n の同じ不連続型コロケーション，一致しない要素が一つでもあれば，α と β は文 s_n 中の異なる不連続型コロケーションとする．

(8) コロケーションの集計

コーパス中のすべての文に含まれる連続型および不連続型コロケーションを手順 (5) と手順 (7) で抽出し，各々単語の組み合わせそのもの (文字列) の比較により，同じ表現が抽出された (出現する) 回数を求める．ここで，集計されたコロケーションの出現回数は，本手法で単語の組み合わせが意味的なまとまりであると判定された回数である．

以上，手順 (1) から手順 (8) で，コロケーションの抽出法を述べてきた．手順 (1) と手順 (8) のみで若干の文字列処理を必要とするほかは，すべて整数 (位置情報) の比較演算のみで行うことができる．このため，計算的な負荷は少なく，高速な処理ができる．また，各文 s_n $(0 \leq n \leq N)$ が含むコロケーショ

ンの抽出処理は，各々独立したものであり，分割処理を行うことで，さらに計算機の負荷を軽くすることができる．

4 コロケーションの抽出実験

前節で述べた抽出法を用いて，コロケーションの抽出実験を行った．使用したデータは，ATR 対話コーパスの国際会議に関するキーボード会話データである．データの大きさを表 1に示す．

表 1: 抽出実験に用いたデータの大きさ

言語	文の数	異なり単語数	延べ単語数
日本語	6,025	3,799	71,780
英語	5,984	3,158	64,088

4.1 実験結果

実験結果を表 2に示す．実験結果より，連続型コロケーションと不連続型コロケーションともに意味的にまとまりのある単語の組み合わせを抽出している．また，名詞間の共起関係 (複合名詞句等) よりも，述語型の定型表現や慣用表現を多く抽出している．ただし，不連続型コロケーションは，全体として出現回数が少ないために，一部にノイズ的なものを含んでいる．この原因の一つとして，今回の実験で使用した言語データの規模が小さかったことがあげられる．抽出された不連続型コロケーションの出現回数は，連続型コロケーションの出現回数に比べ圧倒的に少なく，最も多い場合でも，日本語の場合に 13 回，英語の場合に 5 回に過ぎない．このため，偶然共起したような単語の組み合わせが，ノイズとして混入してしまった．より大規模の言語データを用いれば，ノイズ的な単語の組み合わせの混入を抑えることができると考えられる．

不連続型コロケーションの出現回数が少ないという点に関しては，今回実験に用いた言語データの性質も関係していると思われる．抽出実験に用いた言語データは会話，すなわち話し言葉であり，話し言葉の性質として，断片的で不完全な表現や省略が多いことをあげることができる．その結果として，話し言葉中には不連続型コロケーションの絶対数が少ないのではないかと考えられる．この点に関しては，新聞記事等の言語データを対象とした抽出実験を行い，今回の結果と比較してみる必要があるだろう．これは，今後の課題である．

4.2 位置情報の変化によるコロケーションの違い

第3節で述べた抽出方法では，ALERGIA アルゴリズムにより得られた決定性確率有限オートマトンを用いて，単語の出現位置表を作成している．ALERGIA アルゴリズムの信頼範囲を変化させることにより，異なった出現位置表が作られる．追加実験として，ALERGIA アルゴリズムの信頼範囲を変化させた場合の抽出されるコロケーションの違いを調べた．

状態の等価判定を厳しくするほど，出現回数の多い単語同士の共起関係が強調される傾向にあり，出現回数の少ない長い表現の抽出は抑制された．これにより，単語数の少ないコロケーションが多く抽出された．特に，上位に抽出されたものの多くは 2 単語のみの表現であった．たとえば，日本語の場合には，格助詞間の共起 (“～ の ～ に”, “～ を ～ に” 等) が上位に抽出され，英語の場合には，前置詞と冠詞などの共起が上位に抽出された．上位の表現の出現回数は等価判定を緩くした場合に比べて多くなった．

逆に，等価判定を緩くするほど短い表現の抽出が抑制され，単語数の多い長い表現が優先して抽出された．等価判定が厳しかった場合に抽出されていた短い表現は，より単語数の多い表現に吸収され，抽出されるコロケーションの種類が多くなった．

ただし，あまりに等価判定が厳しすぎる場合は，接頭木アクセプタ (木構造) に近いものとなるために，各文での先頭からの距離値に左右されやすく，また，出現回数の多い単語が，その前後の単語を無視して共起することが増える．このため，コロケーションとして不適切なものが上位に抽出される場合がある．

5 おわりに

本稿では，連続型および不連続型コロケーションの自動抽出法として，単語の出現位置表を用いて，単語の連続・不連続の関係により，出現頻度の高い単語の組み合わせを長さ優先の条件で抽出する方法を提案した．また，ATR 対話コーパスを用いた実験

表 2: 抽出されたコロケーションの例 (日本語データと英語データ)

連続型コロケーションと出現回数		不連続型コロケーションと出現回数	
種類数 10,994　延べ出現回数 26,308		種類数 8,293　延べ出現回数 9,139	
そう ですか	111	はい ～ ですね	13
ですか	95	はい ～ です	13
わかり ました	81	の ～ です	10
そう ですね	81	の ～ が	7
ですね	73	はい ～ が	7
には	46	の ～ の	6
はい わかり ました	46	を ～ に	6
あ そう ですか	43	もしもし ～ ですか	5
ですから	43	はい ～ で ございます	5
はい かしこまり ました	34	はい ～ に	5
失礼 します	32	ええ ～ が	5
でしょう か	32	え ～ ですか	4
種類数 10,127　延べ出現回数 24,692		種類数 5,910　延べ出現回数 6,414	
is that so	82	i see ～ right	5
i see	70	from ～ to	4
and the	62	and ～ are	4
thank you very much	58	as far as the ～ is concerned	4
oh is that so	53	of ～ and	4
okay goodbye	52	professor ～ professor	4
for the	46	whether the ～ or not	3
yes goodbye	43	mr ～ right	3
to the	38	professor ～ of tokyo university and	3
of the	36	yes ～ speaking	3
will be	31	directly ～ he comes	3
thank you	31	from the ～ of the	3

を行い，提案した方法の有効性を示した．

本手法はコーパス中の各々の文に対する処理であることから，任意の文 (またはコーパスの一部分) に含まれるコロケーションを知りたい場合にも有効である．今後の課題としては，本稿で提案したアルゴリズムのよりよい処理対象を調べることや，従来の評価方法と組み合わせることを考えている．

参考文献

[1] 尾本貴志, 北研二 (1996). "距離反比例型スコアを導入したコロケーションの自動抽出法", 情処研報, Vol. 96, No. 27, pp. 75–82.

[2] 北研二, 加藤安彦, 尾本貴志, 矢野米雄 (1994a). "コーパスからのコロケーションの自動抽出: 相互情報量および仕事量基準 – 言語学習の観点からの比較 –", 電子情報通信学会技術研究報告, ET93-128, pp. 31–38.

[3] Kita, K. and Ogata, H.(1997). "Collocations in Language Learning: Copus-Based Automatic Compilation of Collocations and Bilingual Collocation Concordancer", *Computer Assisted Language Learning*, Vol. 10, No. 3, pp. 229–238.

[4] Church, K. W. and Hanks, P.(1990). "Word Association Norms, Mutual Information and Lexicography", *Computational Linguistics*, Vol. 16, No. 1, pp. 22–29.

[5] 北研二, 小倉健太郎, 森元逞, 矢野米雄 (1993). "仕事量基準を用いたコーパスからの定型表現の自動抽出", 情報処理学会論文誌, Vol. 34, No. 9, pp. 1937–1943.

[6] Kita, K., Kato, Y., Omoto, T. and Yano, Y.(1994b). "A Comparative Study of Automatic Extraction of Collocations from Corpora: Mutual Information vs. Cost Criteria", *Natural Language Processing*, Vol. 1, No. 1, pp. 21–33.

[7] Nagao, M. and Mori, S.(1994). "A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese", *The Proceedings of the 15th International Conference on Computational Linguistics*, pp. 611–615.

[8] Smadja, F.(1993). "Retrieving Collocations from Text: Xtract", *Computational Linguistics*, Vol. 19, No. 1, pp. 143–177.

[9] 池原悟, 白井諭, 河岡司 (1995). "大規模コーパスからの連鎖型および離散型共起表現の自動抽出法", 信学技報, Vol. 95, No. 29, NLC95-3, pp. 17–24.

[10] Carrasco, R. C. and Oncina, J.(1994). "Learning Stochastic Regutar Grammars by Means of a State Merging Method", *Grammatical Inference and Applications*, Carrasco, R. C.and Oncina, J. (Eds.), Springer-Verlag, pp. 139–152.