語彙的結束性に基づく語彙的連鎖の計算

望月 源[†]　奥村 学[‡†]　岩山 真[§]

2000 年 8 月 17 日
IS-TM-2000-002

[†] 北陸先端科学技術大学院大学情報科学研究科
[‡] 東京工業大学 精密工学研究所
[§] 日立製作所 中央研究所

目 次

図 目 次

表 目 次

はじめに

本稿では，語彙的結束性に基づく語彙的連鎖の説明と，その計算方法について示し，我々が作成した語彙的連鎖計算プログラムについて説明する．また，他の研究における語彙的連鎖計算の方法についてもまとめて示す.

1　語彙的連鎖

本節では，語彙的連鎖について説明する．まず，1.1節では，談話構造の解析に重要な役割をする一貫性と結束性という，2 つの似ているが異なる性質について説明する．次に，1.2節では，結束性の中で語彙的連鎖を構成する基となる語彙的結束性について具体的な分類を示し説明する．1.3節では，語彙的連鎖について説明し，語彙的連鎖が談話構造を理解する上でどのような役割を果たすかについて述べる．

1.1　一貫性と結束性

文書は単なる文の集まりではなく，文書中の各文は互いに何らかの意味的関係を持っている．特に意味的関係の強い文が集まって談話セグメントと呼ばれる単位を形成する．文が互いに意味的関係を持つように，これらの談話セグメント間にも意味的な関係が存在する．文書の全体的な談話構造はこの談話セグメント間の関係によって形成される．こうした文間やセグメント間の結びつきを扱う重要な言語学的性質として，一貫性 (coherence) と結束性 (cohesion) がある．

一貫性は，文書中の全体的な談話構造を表わす性質である．Hobbs[8] は，一貫性を原因 (occasion)，評価 (evaluation)，背景-描写および説明 (ground-figure and explanation)：ground-figure，explanation，拡張 (expansion)：並行 (parallel)，詳細化 (elaboration)，例示 (exemplification)，対照 (contrast)，反対の予想 (violated expectation) という意味的な関係に分類している．

文書中の各文がこのような意味的な関係によって結びつき一貫した談話セグメントを形成する．また，各セグメント間の関係にも意味的な関係が存在することによって，文書全体としての談話構造である序論，本論，結論や起承転結というような構造，および Hahn[4] の示すような主題-副主題 (theme-rheme) 構造を形成する．

一貫性は談話構造の表現力が高いが，計算機で扱うことは困難である．一貫性を認識するためには，文書内容の完全な理解とセグメント間の関係についての複雑な推論が必要となるためである．また，意味的関係の分類基準が曖昧な点も問題になる．例えば，‘John can open the safe. He knows the combination’ [7] という 2 つの文間の関係は，詳細化か説明のどちらかであると考えられるが分類が難しい [9]．分類基準は，文脈，知識，および分類者の信念に基づく場合が多くシステマティックに関係を決定することができない.

一方，結束性は，文と文の表層的な関係による結びつきのことを言う．Halliday と Hassan は，結束性を理論的な面から，代名詞 (pronoun)，省略 (ellipsis)，接続 (conjunction)，代入 (substitution)，語彙的結束性 (lexical cohesion) の 5 つに分類している [5].

結束性を自動的に計算するという面からは，代名詞と省略は指示 (reference) 関係にまとめることができる．接続は，接続詞が出現する前後の文やセグメント間の関係を結びつける関係である．また，代入は既出の語を他の語に言い換える照応のことを言う (例えば，‘a new one’ の ‘one’，‘do so’ の ‘do’ など) ため，指示関係に含めて考えることができる．従って，結束性は指示，接続詞，語彙的結束性の 3 つに大きく分けられる [10]．

この 3 つの分類の内，接続詞以外の 2 つである指示関係と語彙的結束性は，意味的に同じあるいは関連する 2 つの語によって示される文と文の関係であるという点で類似していると考えられる．しかし，語彙的結束性の場合は，関連する 2 つの語が両方とも文書内に出現するため自動的な計算が比較的容易であるのに対して，指示関係では，関連する 2 つの語の片方は，代名詞であったり省略されているため，結束性関係を自動的に計算する場合の明示的な情報が語彙的結束性に比べて少ないという違いがある．指示関係を計算するためには，照応解析などによって省略された語や指示対象を特定する必要がある．

このようなことから本研究では，結束性の中で計算機によって比較的容易に自動的に認識できる語彙的結束性について扱う．次節で語彙的結束性について説明する．

1.2 語彙的結束性

Morris と Hirst は，語彙的結束性を次の 5 つに分類している [9]．

1. 同一の参照の繰り返し (Reiteration with identity of reference)

   例 1 *Mary bit into a peach. Unfortunately the peach wasn't ripe.*

2. 異なる参照の繰り返し (Reiteration without identity of reference)

   例 2 *Mary ate some peaches. She likes peaches very much.*

3. 上位の概念を持つ語による繰り返し (Reiteration by means of superordinate)

   例 3 *Mary ate a peach. She likes fruit.*

4. 規則性のある意味的な関係 (Systematic semantic relation)

   例 4 *Mary likes green apples. She does not like red ones.*

5. 規則性のない意味的な関係 (Nonsystematic semantic relation)

   例 5 *Mary spent three hours in the garden yesterday. She was digging potatoes.*

例 1から 3は Halliday と Hassan[5] の分類では，反復 (reiteration) と呼ばれる．例 1では，2 つの *peach* がどちらも同じ peach を指すが，例 2では Mary が食べた *peaches* と後の *peaches* は同じものを指してはいない．また，例 3では，*fruit* と *peach* は概念的に上位下位の関係にある．このように，reiteration には，同じ語の繰り返しだけでなく上位 (superordinates) 関係，下位 (subordinates) 関係および類義語 (synonyms) が含まれる．

例 4と 5は Halliday と Hassan の分類では，collocation と呼ばれる．例 4は，規則性のある意味的な関係による collocation である．このタイプの関係は反対語 (antonyms)，順位性のある語 (one,two,three)，順位性のない語 (white,black,red)，全体部分関係 (eyes, mouth, face) に分類できる．例 5は，規則性のない意味的な関係による collocation である．このタイプの関係は計算が難しい．例えば例 5の，*garden* と *digging* には意味的な関係があるが規則性はない．このタイプの結束性は，類似した状況や内容を記述する場合に共起する傾向の高い語と語のあいだに存在すると考えられる．例えば，*teacher, school, examination* は学校についての内容を持つ文書中では，結束性を持つ関係であると考えられる．

しかし実際に語彙的結束性を計算する上では，1 と 2 を厳密に区別することは難しい．どちらも表層上は同一の語で構成されるためである．同一の語の繰り返しについて，それが 1 の結束性であるか 2 の結束性であるかを決定するためには，他の語による情報を取り入れる必要がある．しかし，同一の文書中に出現した同じ語は参照が同一か異なるかを別にすれば同じ意味で使用されていると考えることができる．そのため，本研究では語彙的結束性を構成する基準として，1 と 2 を区別せず，同一の語の繰り返しに基づく語彙的結束性として分類する．

また，3 と 4 はどちらも語彙的な知識を記述したシソーラスなどの知識ベースをあらかじめ用意することにより計算できる．そのため，本研究では 3 と 4 を区別せず，シソーラスに基づく語彙的結束性として分類する．

5 は，あらかじめ用意した知識ベースでは計算できず，類似した状況や内容を記述する際に共起する傾向が高い語を計算する必要がある．従って，語の共起関係に基づく語彙的結束性として分類する．

語彙的結束性の構成基準の違いによる分類をまとめると次のようになる．

1. 同一の語の繰り返しに基づく語彙的結束性
2. シソーラスに基づく語彙的結束性
3. 語の共起関係に基づく語彙的結束性

1.3 語彙的連鎖

例として次の文 (例 6) を考える．

例 6 膨張を続ける 宇宙$_1$ の中で多くの 星$_1$ が 誕生$_1$，消滅$_1$ を繰り返しました．そして 宇宙$_2$ の 誕生$_2$ から約 *100* 億年後，他の 星$_2$ と同じ様にして，原始$_1$太陽$_1$ を中心にして 原始$_2$太陽系$_1$星雲$_1$ を作りました．

1.2節で述べた 3 つの語彙的結束性の分類に照らして考えると，例 6には次のような語彙的結束性が見られる．

1. 同一の語の繰り返しに基づく語彙的結束性
   星$_1$ と 星$_2$，宇宙$_1$ と 宇宙$_2$，誕生$_1$ と 誕生$_2$，原始$_1$ と 原始$_2$ の間に語彙的結束性がある．

2. シソーラスに基づく語彙的結束性
   日本語の代表的なシソーラスである角川類語新辞典 [17] を調べると,「星」と「星雲」,「誕生」と「消滅」をそれぞれ含む分類がシソーラス上に存在する．そのため，星$_1$ と 星$_2$ と 星雲$_1$，誕生$_1$ と 消滅$_1$ と 誕生$_2$ がそれぞれ語彙的結束性を持つ．また，同一の語も当然シソーラス上で同じ分類に含まれるため，宇宙$_1$ と 宇宙$_2$，原始$_1$ と 原始$_2$ の間にも語彙的結束性がある．

3. 語の共起関係に基づく語彙的結束性
   例文は宇宙に関する話題で星の誕生や消滅について述べた内容であるから，{宇宙$_1$，星$_1$，宇宙$_2$，星$_2$，太陽$_1$，太陽系$_1$，星雲$_1$}，{誕生$_1$，消滅$_1$，誕生$_2$}，{原始$_1$，原始$_2$} がそれぞれ共起しやすい語であり語彙的結束性を持つと考えられる．

この例のように，文書中では 2 つの単語間だけでなく，互いに語彙的結束性のある単語が連なることが良くある．この文書内での語彙的結束性を持つ語の連続のことを語彙的連鎖 (lexical chain) と呼ぶ [9]．語彙的連鎖は文書中で意味的にまとまった部分を示している．一般に，文書中には複数の語彙的連鎖が存在する．1 つの語彙的連鎖が出現する範囲では，その連鎖を構成する語に関連する話題が述べられていると考えられる．文書中で話題が変化する場合，そこに出現する語彙的連鎖も変化すると考えられる．そのため，語彙的連鎖は次の重要な特徴を持つ．

1. 語彙的連鎖は文書中の局所的な文脈情報を提供する．

2. 語彙的連鎖は文書中の一貫性や談話構造を決定するための手がかりを提供する．

1.3.1 語彙的連鎖を構成する基準

語彙的連鎖は語彙的結束性を持つ語の連続であるから，語彙的連鎖を構成する基準は，語彙的結束性を構成する基準と同じである．

1. 同一の語の繰り返しに基づく語彙的連鎖
   表層形式の同じ語は互いに意味的に関連のある語と考え連鎖を構成する．

   {星$_1$，星$_2$}，{宇宙$_1$，宇宙$_2$}，{誕生$_1$，誕生$_2$}，{原始$_1$，原始$_2$}

2. シソーラスに基づく語彙的連鎖
   シソーラス上の同一概念に属する語を互いに意味的に関連のあるタームと考え，連鎖を構成する．

   {星$_1$，星$_2$，星雲$_1$}，{誕生$_1$，消滅$_1$，誕生$_2$}，{宇宙$_1$，宇宙$_2$}，{原始$_1$，原始$_2$}

3. 語の共起関係に基づく語彙的連鎖
   このタイプの語彙的連鎖は，既存のシソーラスを使用しない．文書コーパスの共起情報から単語間の共起の強さである共起スコアを計算する．このスコアを用いて語のクラスタを構成する．1 つのクラスタ内の語の連続が 1 つの語彙的連鎖となる．

   {宇宙$_1$，星$_1$，宇宙$_2$，星$_2$，太陽$_1$，太陽系$_1$，星雲$_1$}，{誕生$_1$，消滅$_1$，誕生$_2$}，{原始$_1$，原始$_2$}

1 の同一の語の繰り返しに基づく語彙的連鎖は，最も単純な連鎖構成基準である．表層形式が同一の語を認識することで語彙的連鎖が構成できるため，容易に計算ができる．一方，実際には結束性を持っている語であっても，表層形式が異なる語は同一の語彙的連鎖に含まれないという問題もある．

2 のシソーラスに基づく語彙的連鎖では，既存のシソーラスに記述された概念構造から語と語の間の同義語，類義語，上位下位関係などを発見し，互いに意味的関連がある語と判断された場合に語彙的連鎖を構成する．これらの関係の内，どの関係が抽出可能かは，使用するシソーラスの構造に依存する．現在計算機で利用可能なシソーラスには，WordNet[2]，EDR 概念体系辞書 [15]，分類語彙表 [19]，角川類語新辞典 [17]，Roget' International Thesaurus[11] などがあるが，使用するシソーラスの構造を考慮した語彙的連鎖の計算手法が必要になる．

また，シソーラスでは同一の語が複数の概念に分類されていることが多く，語義曖昧性の問題が発生する．語がどの概念に対応するかによって語義が異なるため，シソーラスを使用する場合には，語義の決定が必要になる．このことは逆に，語彙的連鎖を構成する過程で語義の曖昧性を解消することも可能にしている．実際の例は 2節で示す．

3 の語の共起関係に基づく語彙的連鎖は，最も計算が複雑な手法である．シソーラス上には記述されていないが，意味的に関連のある語を計算するためには，一般に大規模なコーパスを用いた語の共起関係に関する統計情報を使用する．2 つの語がコーパス中の多くの文書で共起する場合には，互いに意味的な関連が高い語であると推測される．また，別の見方として，2 つの語の一方が出現する時に共起する語の集合と，もう一方が出現する時に共起する語の集合が類似している場合にも，2 つの語は非常に類似した場面で使用されている可能性が高い．このような関係を共起情報を基に計算する．

2　語彙的連鎖の計算

本節では，我々の語彙的連鎖計算プログラムのアルゴリズムについて述べる．本研究では，連鎖構成基準の違いにより，同一の語の繰り返しに基づくもの，シソーラスに基づくもの，語の共起関係に基づくものの 3 種類の語彙的連鎖を実装する．ここでは，具体的な実装方法について順に説明する．このうち，語の共起関係に基づくものは我々が新たに提案した手法に基づいている．また，他の研究における代表的な語彙的連鎖計算手法についても説明する．

2.1　前処理

3 種類全ての連鎖構成手法において，以下の前処理を行ない，連鎖を構成する候補語を決定する．

まず，文書を形態素解析 [18] し，動詞，名詞，形容詞を候補語として選択する．ただし，表 1にあげた「よる」,「する」などの動詞,「こと」などの形式名詞,「(〜の) ため」,「まま」などの副詞的名詞および，ひらがな 1 文字の名詞は除く．また,「〜だ」で終る形容詞は「だ」を取り除いた形に統一する．例文 6から選択した候補語の例を表 2に示す．

表 1: 除去する動詞一覧

する	ある	なる	よる
よって	より	できる	つれて
ついて	という	とする	した

2.2　同一の語の繰り返しに基づく語彙的連鎖

同一の語の繰り返しに基づく語彙的連鎖は，最も単純な手法である．表層形式の同じ語は互いに意味的に関連のある語と考え連鎖を構成する．

2.1節で説明した方法により，文書中から選択した候補語について，表層形式の同じものをまとめることで語彙的連鎖を構成する．

表 2を例とすると，構成される語彙的連鎖は次のようになる．

{宇宙$_1$,宇宙$_2$}，{星$_1$, 星$_2$}，{誕生$_1$，誕生$_2$} {原始$_1$,原始$_2$}

表 2: 候補語の例

文番号	候補語
1	膨張$_1$, 続ける$_1$, 宇宙$_1$, 中$_1$，多く$_1$, 星$_1$, 誕生$_1$，消滅$_1$, 繰り返す$_1$
2	宇宙$_2$, 誕生$_2$, 100$_1$，他$_1$，星$_2$，同じ，原始$_1$，太陽$_1$, 中心$_1$, 原始$_2$, 太陽系$_1$，星雲$_1$, 作る$_1$

2.3 シソーラスに基づく語彙的連鎖

このタイプの語彙的連鎖は，シソーラス上の同一概念に属する語を互いに意味的に関連のある語と考え，連鎖を構成する．

シソーラスを使用する場合，1 つの語が複数の分類に含まれる場合には，語義曖昧性の問題が発生する．そのため，本研究では Okumura と Honda の語義曖昧性を解消しつつ語彙的連鎖を構成する手法 [10] により語彙的連鎖を構成する．

以降で説明する我々の実装アルゴリズムでは，語彙的結束性を計算するシソーラスとして分類語彙表と角川類語新辞典 [17] のどちらも使用できるが，ここでの説明は角川類語新辞典を例にとっておこなう．

角川類語新辞典では，図 1に示すように，概念体系を大分類・中分類・小分類の 3 階層に分類し，各階層はそれぞれ 10 個に分類されている．また，分類によっては小分類の下に細分類があるものもあるが，全て小分類を使用する*.

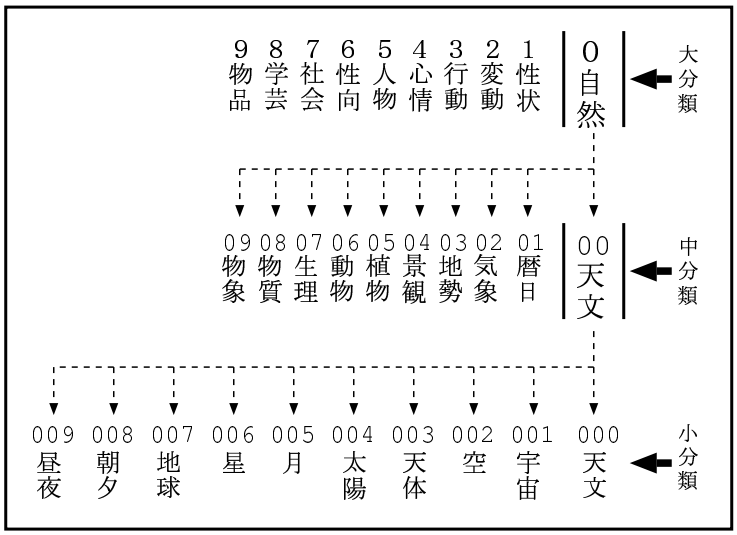

図 1: 角川類語新辞典の分類階層

2.1節で説明した方法により文書中から選択した候補語について，語彙的連鎖を次のように計算する．まず，文書から 1 文を取り出し，1 文内の候補語の語彙的結束性を調べるため，各候補語についてシソーラスを索き，分類番号と候補語のマトリックスを作成する．マトリックスの例を図 2に示す.

次に，このマトリックスを参照し，ある候補語の属す分類と同じ分類に属す他の候補語が存在する場合には，一致する分類のみを残し部分的な曖昧性解消を行なう．この際，次の方針で部分的な曖昧性解消を行なう．

1. ある分類 (分類 A) が，他の分類 (分類 B) の全候補語と分類 B に含まれない候補語を持つ場合，
   分類 B を削除する (部分的な語義曖昧性の解消).

*小分類は $10^3 = 1000$ 分類あるが，語彙的連鎖構成の基準として考慮しない方が良いと考えられる {508：接尾辞，509：接辞，828：単位，829：助数詞および 834：接辞 } の 5 つの分類を除外する．

2. 分類 A と分類 B がまったく同じ候補語を持つ場合，
   分類 A，分類 B とも残す．

3. 上記の条件に該当しない分類は全て残す．

	候補語								
分類番号	膨張1	続ける1	宇宙1	中1	多く1	星1	誕生1	消滅1	繰り返す1
262	●								
243	●								
284		●							●
001			●						
192				●					
157				●					
153				●					
107				●					
103				●					
126					●				
006						●			
111						●			
260							●	●	
071							●		
272								●	

図 2: 候補語と分類番号のマトリックスの例

例えば，図 2では，候補語「誕生$_1$」は 260 と 071 に属し，「消滅$_1$」は 260 と 272 に属すが，260 が 2 つの候補語に共通の分類なので，1 の基準により，260 を残し 071 と 272 は削除され，語彙的連鎖の候補として，{誕生$_1$，消滅$_1$} が生成される．この処理により，文レベルの部分的な語義曖昧性解消が行なわれ，語彙的連鎖の候補が計算される．文レベル終了後の語彙的連鎖候補は，図 3の (A) のようになる．

この段階で多義性の解消が行なえなかった候補語からなる語彙的連鎖候補は，複数の分類を持つ．例えば，図 3の語彙的連鎖候補 {膨張$_1$}，{中$_1$}，{星$_1$} は，多義性が解消されていないため，それぞれ，{243,262}，{192,157,153,126,107,103}，{111,006} という複数の分類を持つ．

次に，今，計算した文レベルの各語彙的連鎖候補と，既に計算されスタックに存在する，文書レベルの語彙的連鎖候補を比較し，スタックの更新をする．ただし，1 文目を処理する時点ではスタックが空なので，文レベルの各語彙的連鎖候補を以下の基準によって順番に取り出し，スタックに積む．

- 語彙的連鎖候補は，各連鎖候補に含まれる語の文中での出現位置の順に取り出す．
- 語彙的連鎖候補内に，語が複数存在する場合は，文中で一番最後に出現した語の出現位置をその連鎖候補を取り出す順位とする．

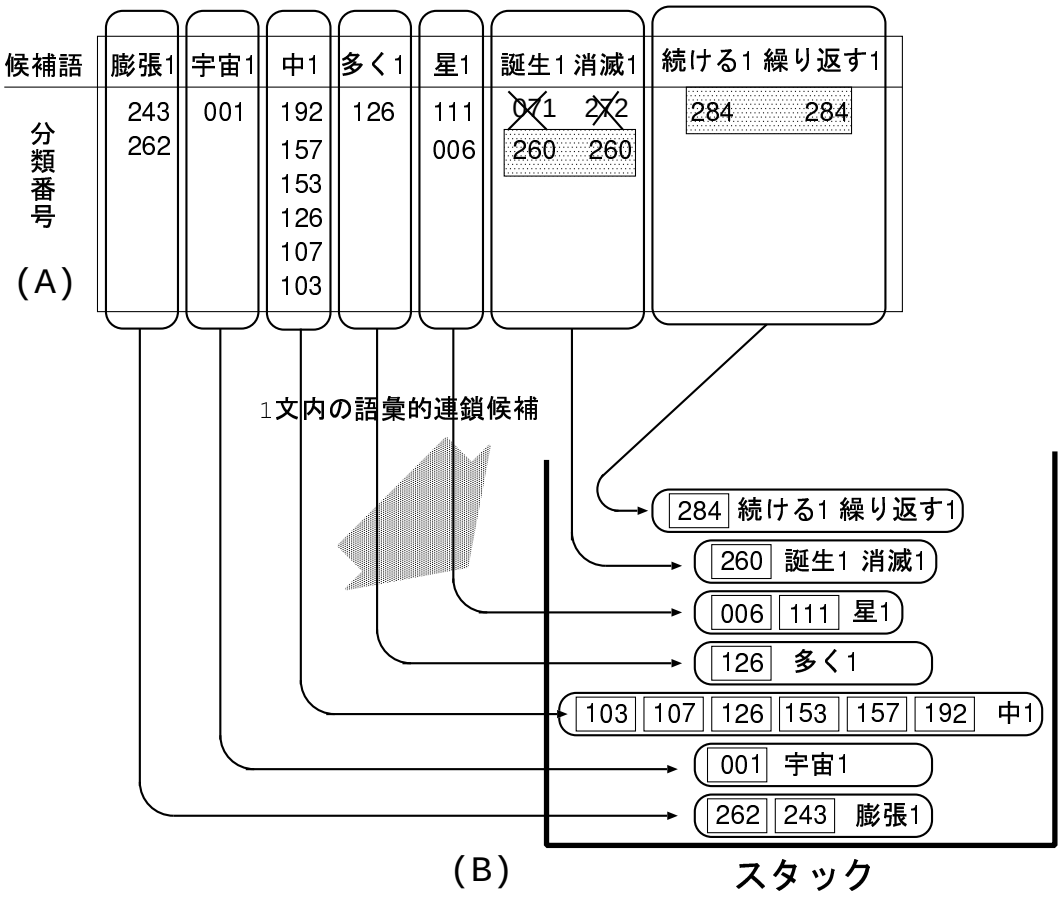

図 3: 語彙的連鎖候補 (A) と 1 文目終了後のスタックの状態 (B)

1 文目を処理した後の例を図 3の (B) に示す．図 3の (B) では，連鎖候補内の語の出現が早いものほどスタックの下に積まれていることがわかる．

2 文目以降の処理では，スタックが空ではないので，次のようにスタックの更新を行なう．まず，文レベルの各語彙的連鎖候補を，1 文目の時と同じ基準で順番に取り出す．ここで，取り出した文レベルの連鎖候補を S_Chain とする．次に，スタック内に存在する文書レベルの各語彙的連鎖候補をスタックの上位から順に参照し，連鎖候補の持つ分類番号によって，S_Chain とのマッチングを行なう．文レベルの連鎖候補 S_Chain と同一の分類を持つ文書レベルの連鎖候補が発見された場合 (ここでこの文書レベルの連鎖候補を T_Chain とする)，T_Chain を S_Chain の内容によって更新する．まず，T_Chain に S_Chain 内の候補語を追加する．次に，T_Chain をスタックの最上位に移動する．この時，T_Chain と S_Chain が共通に持つ分類以外の分類は全て削除する．これにより，文書レベルによるさらなる語義の曖昧性解消が行なわれる．

一方，スタック内に文レベルの連鎖候補 S_Chain とマッチする連鎖候補が存在しない場合は，S_Chain をスタックの最上位に積み，新たな文書レベルの連鎖候補とする．

この手法では，最近更新された語彙的連鎖の候補が，常にスタックの最上位になる．また，S_Chain とのマッチングはスタックの上位から順に行なわれるので，S_Chain の近傍の文脈を得られる構造になっている．

スタックの更新過程の例を図 4に示す.

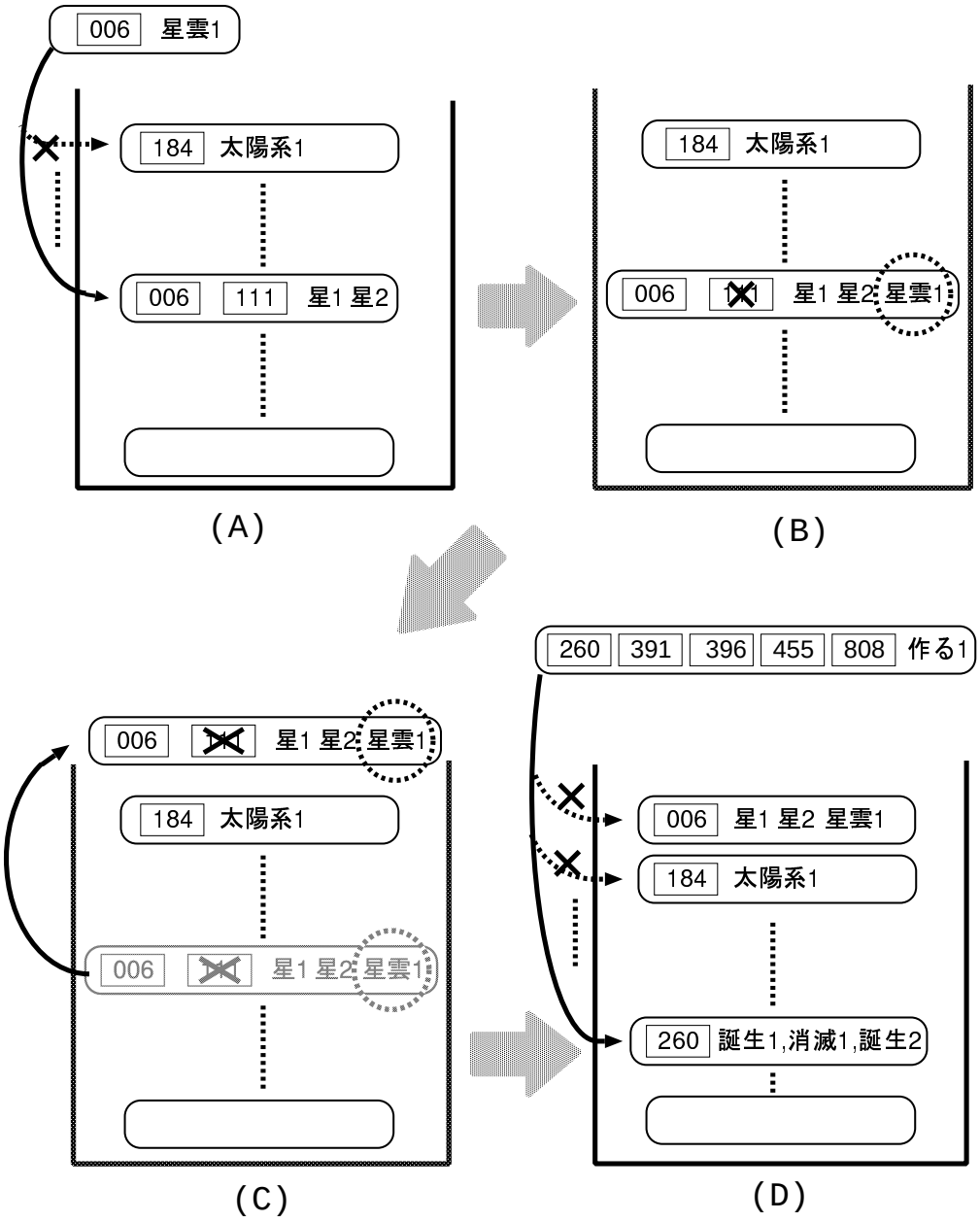

図 4: スタックの更新

図 4(A) では，文レベルの連鎖候補 S_Chain (分類番号 006, 候補語 「星雲$_1$」) とスタック内の連鎖候補のマッチングが行なわれている．次に，マッチする連鎖候補 T_Chain が発見され，S_Chain 内の候補語「星雲$_1$」が追加される (B)．続いて，更新された連鎖候補 T_Chain をスタックの最上位に移動し，S_Chain と T_Chain で一致しない分類である 111 が T_Chain から削除される (C)．ここまでで現在参照中の文レベルの語彙的連鎖候補 S_Chain の解析が終了し，次の連鎖候補 S_Chain' により，処理が繰り返される (D)．この文書レベルの処理は文レベルの語彙的連鎖の候補がなくなるまで繰り返される．

以上の文レベルと文書レベルの処理を文書末まで繰り返す．最終的にスタックに存在する語彙的連鎖の候補を取り出して，それぞれ 1 つの語彙的連鎖とする．この段階で，連鎖候補の中に分類が複数存在する場合は，その連鎖に関しては多義性の解消が行なえなかったことになる．

なお，シソーラスに存在しない未知語に関しては，便宜的に分類番号をその語の表層形式とすることで，語彙的連鎖の計算に含める．そのため，未知語の場合は，同一の語の繰り返しによる手法と同様の語彙的連鎖が構成される．

シソーラスに基づく語彙的連鎖のアルゴリズムをまとめると次のようになる．

ステップ 1.
if (文書から取り出す文がない)
スタックにある連鎖を出力して終了
else
文書から 1 文を取り出し，**ステップ 2** へ
ステップ 2. 1 文内の語彙的連鎖候補を決定
ステップ 2-1.
シソーラスを調べ，1 文内の全候補語の分類番号を取得する
ステップ 2-2.
分類番号と候補語のマトリックスを作成する
ステップ 2-3.
マトリックスを調べ，分類ごとに可能な語彙的連鎖候補を作る
ステップ 2-4.
if (未参照の連鎖候補が存在しない)
ステップ 2-5 へ
else
連鎖候補を 1 つ取り出し，同じ候補語を含む他の連鎖候補を全て探す
if(他の候補が存在する)
候補語の数が最も多い連鎖候補を残し，他の連鎖候補は削除する
ステップ 2-4 へ
ステップ 2-5.
同じ候補語を含む連鎖候補を 1 つの連鎖候補にマージし，**ステップ 3 へ**
ステップ 3. 文書全体のスタックの更新
ステップ 3-1.
if (スタックが空)
候補語の出現順に文レベルの語彙的連鎖候補をスタックに積み，**ステップ 1** へ
else
ステップ 3-2 へ
ステップ 3-2.
候補語の出現順に文レベルの語彙的連鎖候補を並べる．
ステップ 3-3.
if (未参照の文レベルの語彙的連鎖候補が存在しない)
ステップ 1 へ

else

文レベルの連鎖候補を 1 つ取り出す

ステップ 3-4.

文書レベルの連鎖候補の分類番号とのマッチングをスタックの上から順に行なう

ステップ 3-5.

if (分類番号の一致する連鎖候補がスタック内に存在する)

連鎖候補を更新し，スタックの最上位に積み，**ステップ 3-3** へ

else

新しい連鎖候補をスタックの最上位に積み，**ステップ 1** へ

2.4 語の共起関係に基づく語彙的連鎖

本研究では，3 番目の語彙的連鎖構成基準として，語の共起関係に基づく語彙的連鎖の計算手法を提案し，実装する．このタイプの語彙的連鎖は，既存のシソーラスの代わりに，大規模な文書コーパスを使用し，文書コーパス内の語の共起情報から共起スコアを計算する．この共起スコアを語と語の結束性の強さとして，文書中の語によるクラスタを構成する．同一のクラスタ内の語の連続が 1 つの語彙的連鎖となる．

文書数 N のコーパスを使用することで，コーパス内の文書を次元とし，各文書にその語が出現する頻度を要素とする N 次元のベクトルを考えることができる (図 5)．このベクトルを使用すると，任意の語 X と語 Y の共起の強さを，それぞれのベクトル間の類似度によって計算できる．

	文書										
語	1	2	3	4	5	6	7	8	9	$\cdots$	N
語 1	0	5	0	2	8	0	0	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
語 X	2	0	8	1	0	2	1	0	4	$\cdots$	1
語 Y	0	0	5	2	0	2	0	2	2	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
語 m	0	5	0	2	8	0	0	0	0	$\cdots$	0

図 5: 文書ベクトルの例

本研究では，語 X と Y の共起スコアを式 (1) のコサイン距離によって計算する．

$$coscr(X,Y) = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{N} x_i^2} \times \sqrt{\sum_{i=1}^{N} y_i^2}} \tag{1}$$

ここで，x_i と y_i はそれぞれ文書 i に語 X と Y の出現する数 (tf) を表わし，N はコーパスの全文書数を表わす．なお，本研究ではコーパス中の語の出現頻度の取得に，(株) 日

立製作所中央研究所において開発された汎用連想検索エンジン GETA(Generic Engine for Transposable Association)[16] を使用している．

また，クラスタリングにおけるクラスタ間の類似尺度には次の最短距離法を用いる．

$$sim(C_i, C_j) = \max_{X,Y} coscr(X \in C_i, Y \in C_j) \tag{2}$$

ここで X, Y はそれぞれクラスタ C_i 内，C_j 内の語である．

最短距離法では，2 つのクラスタ間の類似度を各クラスタ内の要素の内，最も類似度の高い (距離の近い) 要素ペアの類似度 (距離) とする (図 6)．

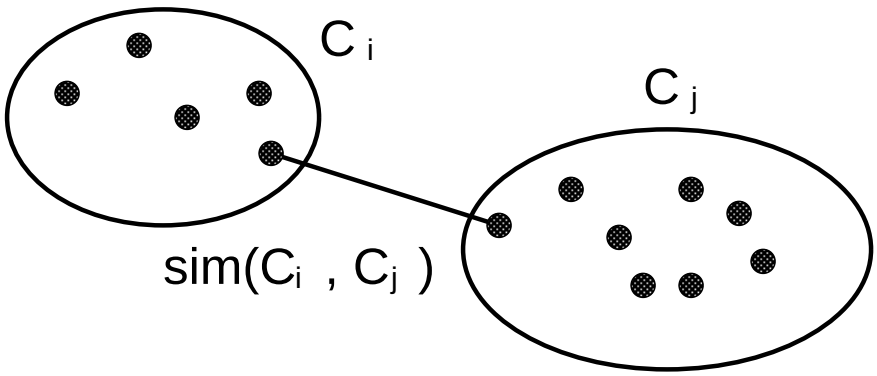

図 6: 最短距離法

この共起スコアを使用した文書ごとの語彙的連鎖の計算は，次のように行なう．まず，文書から 1 文を取り出し，式 (1) によって，1 文内の語の全てのペア間の共起スコアを計算する．1 語 1 クラスタから開始し，クラスタ間の類似度を語の共起スコアを基に式 (2) の最短距離法により計算する．類似度が最も高いクラスタ間の類似度が閾値以上の場合，そのクラスタペアをマージする．類似度が最も高いクラスタ間の類似度が閾値以上である間マージを繰り返すことで，文レベルのクラスタリングを行なう．クラスタリングの例を図 7に示す.

図 7では，1 文内に候補語が $W1, W2, W3, W4, W5$ の 5 語存在する．まず，1 語 1 クラスタとしてクラスタ間の距離を計算する (A)．この時，最も類似度の高いクラスタペア $C1$ と $C3$ の類似度が 0.8 であり，閾値を 0.25 とすると，閾値以上であるためクラスタ $C1$ と $C3$ がマージされる (B)．マージ後に最も類似度が高いペアは $C2$ と $C5$ であり，類似度は 0.7 であるため，次に $C2$ と $C5$ がマージされる (C)．次に類似度が高いペアは $C1$ と $C4$ であるが，類似度が 0.2 であり，閾値を下まわるため，クラスタリングを終了する (D)．結果的に図 7では，$C1$,$C2$,$C4$ の 3 つの語彙的連鎖候補が得られる．

1 文内での処理が終了した後に，その時点までに作成された文書全体でのクラスタと今計算した 1 文内でのクラスタによる 2 段階目のクラスタリングを 1 文内の場合と同様に行なう．これを文書内の文がなくなるまで繰り返す.

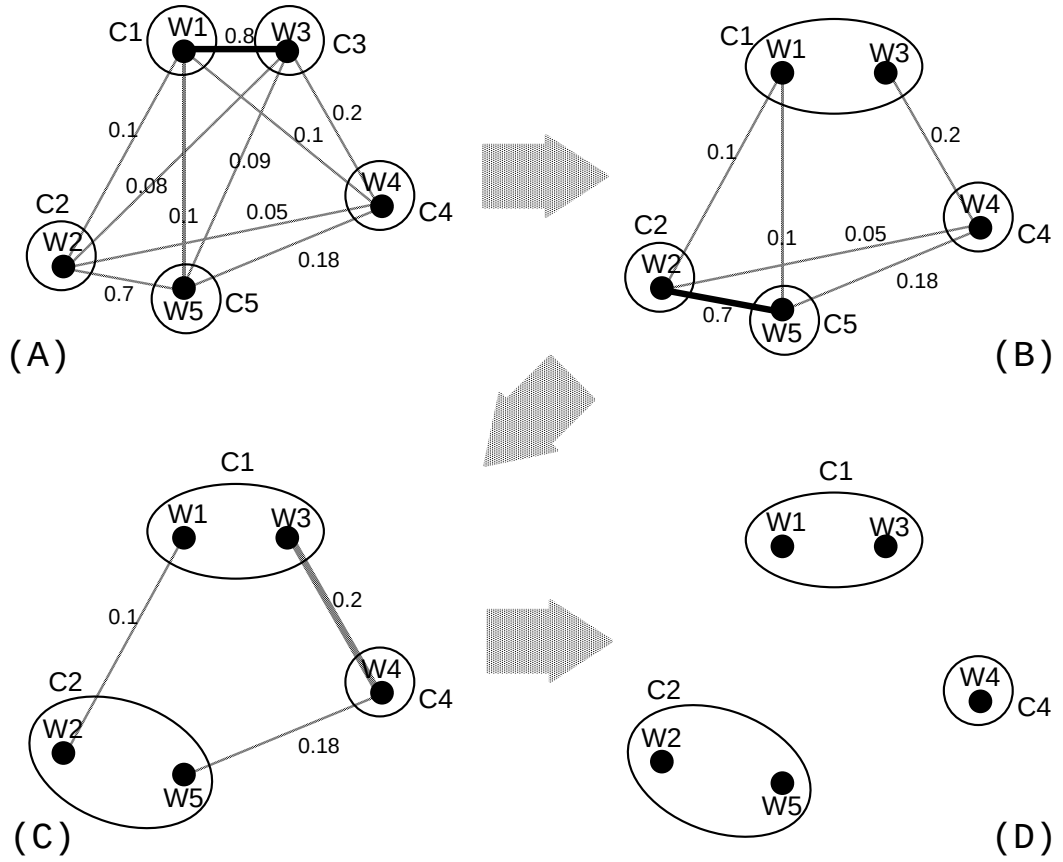

図 7: クラスタリングの例

共起関係に基づく語彙的連鎖構成アルゴリズムは次のようになる.

ステップ 1.
if (文書から取り出す文がない)
終了
else
文書から 1 文を取り出し，**ステップ 2** へ
ステップ 2. 1 文内の語のクラスタリング
ステップ 2-1.
1 文内のすべての語ペア間の共起スコアをコサイン距離 (式 (1))
により計算する
ステップ 2-2.
1 語 1 クラスタとしてクラスタを形成する
ステップ 2-3.
if (クラスタが 1 つである)
ステップ 3 へ
else
すべてのクラスタペア間の類似度を最短距離法 (式 (2)) により計算する

ステップ 2-4.
if (類似度の最も高いクラスタペアの類似度が閾値以上である)
クラスタペアをマージし，**ステップ 2-3** へ
else
ステップ 3 へ

ステップ 3. 文書全体の語のクラスタリング
ステップ 3-1.
if (文書全体のクラスタが存在しない)
ステップ 2 で計算したクラスタを文書全体のクラスタとする
else
ステップ 2 で計算した 1 文内の各クラスタ内の各語と，
文書全体の各クラスタ内の
語との共起スコアをコサイン距離 (式 1)) により計算する
ステップ 3-2.
if (マージするクラスタがない)
ステップ 1 へ
else
1 文内のクラスタと文書全体のクラスタの
全ペア間の類似度を最短距離法 (式 (2)) により計算する
ステップ 3-3.
if (類似度の最も高いクラスタペアの類似度が閾値以上である)
クラスタペアをマージし，**ステップ 3-2** へ
else
ステップ 1 へ

2.5 他の研究における語彙的連鎖計算方法

本節の最後に，他の研究における語彙的連鎖の構成方法について述べる．現在，語彙的連鎖の構成基準として非常に多く使用されているのは，同一語の繰り返しによるものと，シソーラスに基づくものであるが，同一の語の繰り返しは手法として単純であり，基本的に本研究の手法と同じである．そこで，他の研究におけるシソーラスに基づく語彙的連鎖計算手法について述べる．

シソーラスとしては，日本語の場合，分類語彙表 [19] を使用するもの [10] と，角川類語新辞典 [17] を使用するもの [14] があり，英語の場合は WordNet[2] を使用するもの [12, 3, 13, 1] が多い．

分類語彙表と角川類語新辞典では，概念階層の深さが均一になるように考慮されており，見出し語は最下層の分類に配置されている．このタイプのシソーラスを用いる研究には，意味的な類似度を計る基準として，概念階層のどの深さを用いるかという違いはあるが，同一概念に属す語に語彙的結束性があるとする点は同じである．

佐々木ら [14] は，角川類語新辞典の中分類を使用して†，本研究とは異なるアルゴリズムで，語彙的連鎖‡を計算する．候補語として，シソーラスの見出し語になっている語を選択し，同じ分類に属す候補語をまとめることで語彙的連鎖を計算する．この際，多義の候補語に対しては，以下に示す 2 種類の曖昧性解消手法を提案している．

- 形式段落を語彙的連鎖を計算する単位とし，形式段落の先頭から現在解析中の語までに出現した分類の累積頻度を計算する．多義性の解消は，多義語が属する分類の内，その時点で累積頻度が最大の分類を選択することで行なう．

- 文書全体を語彙的連鎖を計算する単位とし，文書の先頭から現在解析中の語までに出現した分類の累積頻度を計算する．多義性の解消は，多義語が属する分類の内，その時点で累積頻度が最大の分類を選択することで行なう．

また，どちらの手法においても，最大累積頻度を持つ分類が複数存在する場合は，分類番号の若い方を選択する．佐々木らはこの 2 種類の手法の内，語義決定に用いるデータ数の問題で，文書全体を用いた方が良い結果が得られると報告している．

一方 WordNet は，同一概念を表わす単語および複合語 (例えば，school も private school も 1 つのエントリとなる) の集合 (synset) をノードとし，各 synset 間がリンクされたネットワーク構造になっている．同じ単語が複数の synset に含まれている場合が多く，各 synset を 1 つの語義と考えることができる．synset 間を結ぶリンクには表 3に示す意味的な属性が付加されている．

表 3: WordNet のリンク属性

hypernym(y)	is-a に相当するリンク (上方向へのリンク)
hyponym(y)	is-a に相当するリンク (下方向へのリンク)
holonym(y)	全体-部分関係の部分に相当するリンク (下方向へのリンク)
meronym(y)	全体-部分関係の全体に相当するリンク (上方向へのリンク)
antonym(y)	反対の意味を表わすリンク
also see	意味的に関連しないが，その語を含む表現へのリンク (水平方向へのリンク)
attribute	密接な関係にある名詞と形容詞間のリンク (水平方向のリンク)
cause	他の行動の結果 (下方向のリンク)
entailment	他の行動の含意 (下方向のリンク)
pertain	名詞-形容詞間，および形容動詞から形容詞への形態的な関係を表わすリンク (水平方向へのリンク)

WordNet はこの意味的なリンク属性を参照することでシソーラスとして利用できる．ただし，各リンクによってどのくらい概念的な距離があるかは定かでない．そのため，WordNet を利用する研究では，リンクの属性に基づいて，単語 (synset) 間の結束性関係の強さを独自に定めている．以降では，代表的な 3 つの語彙的連鎖構成手法である，St-Onge の手法 [12]，Stairmand の手法 [13]，Barzilay と Elhadad の手法 [1] について説明する．

†本研究では小分類を使用している．

‡佐々木らは「結束チャート」と呼んでいる．

St-Onge[12] は，WordNet の見出し語になっている語を候補語とする．語と語の結束性関係は，強さの順に extra-strong, strong,medium-strong の 3 種類に分類し，以下のように定義している．

- extra-strong
  同一の語が繰り返す場合

- strong
  - 2 つの語間に共通の synset が存在する場合
  - 各語を含む synset 間に水平方向のリンクが存在する場合
  - 一方の語がもう一方の語に複合的に含まれ，synset 間に何らかの意味的関係 (リンク) が存在する場合

- medium-strong
  2 つの語，それぞれを含む synset 間に (8 パターンの) 特別なパスが存在する場合

語彙的連鎖の構成手法は次のようになる．まず，空の連鎖リストを用意する．次に候補語を一語づつ取り出し，連鎖リストに加えていく．その際，既にリストに入っている語との語彙的結束性を調べ，語彙的結束性があればその関係の強さ (extra-strong,strong,medium-strong) を記録する．また，語が複数の synset に出現する場合，語彙的結束性のある synset だけを残すように更新する．現在の連鎖リストにある語と語彙的結束性がない場合は，新しい連鎖リストを作る．この処理を一文ごとに行ない．一文での処理が終了した時点で，文書全体での連鎖のスタックを更新する．このスタック更新の手法は基本的に**??**節で説明した本研究の手法 [10] と同じである．つまり，連鎖リストを最近出現した連鎖が上位に来るようにスタック状に管理し，スタックの上位にある連鎖リストから順に解析中のリストとの語彙的結束性を調べる．一致するものがあれば，更新しスタックの最上位に積み，なければ新しくスタックの上位に積む．解析が終了した時点でスタック上に存在する連鎖リストが語彙的連鎖となる．

Stairmand[13] は，名詞と形容詞を候補語とする．ただし，形容詞は相当する名詞で置き換える．以下のパターンに合う語と語に語彙的結束性が存在すると定義している．

- 2 つの語が同じ synonym set に出現する場合

- 各語を含む synset 間が hyponym(あるいは hypernym) 関係をたどることでリンクされる場合 (上位下位関係)

- 各語を含む synset 間が meronym(あるいは holonym) 関係をたどることでリンクされる場合 (全体部分関係)

- 各語を含む各 synset から hypernym 関係をたどって，あるノードでぶつかる場合．ただし，約 400 ノードの概念的に一般的過ぎるノードについては除く．

語彙的連鎖の構成手法は次のようになる．

WordNet を調べ，各候補語 t と上の条件に合う関係のある他の候補語を調べ t に関連する候補語リスト (RELATED_TERMS(t)) を作成する．次に，RELATED_TERMS(t) 内の

各候補語ペア間の関係を調べ関連のある候補語のペアで結束性リンク (COHESION_LINKS) を作成する．次に同じ候補語を持つ候補語をマージすることで語彙クラスタを作成する．例えば，{car/1,vehicle/1} と {break/1,car/1} という 2 つの COHESION_LINK がある場合，どちらも car/1 が共通であるので，{car/1,vehicle/1,break/1} という語彙クラスタを作成する．ここで，car/1 の car は単語を表わし，1 は概念番号を表わす．次に作成されたクラスタを参照し，同じ語を持つクラスタをマージする．例えば，
{jury-court-jurors}，{court-supreme court-appeals court}，{juror-foreman-panelist}
の 3 つのクラスタはマージされ，
{jury - court - supreme court - appeals court - juror - foreman - panelist} となる．

続いて各語彙クラスタから語彙的連鎖を取り出す．その際次の規則を適用する．

1. 候補語間の位置が 80 words 以上離れたら，ギャップとし部分的な連鎖に分ける．

2. 部分的な連鎖には，最低 3 つの要素を必要とし，2 つ以下は連鎖と認めない．

ここまでの計算では，同一の位置にある候補語が複数の語彙的連鎖に重複して含まれる可能性がある．そのため，各語彙的連鎖を抽出した元のクラスタのスコアを計算し，重複単語がある場合には最もスコアの高いクラスタに属す連鎖を優先し，低いクラスタに属す連鎖内の重複単語を取り除いて連鎖を更新する．更新された連鎖に再度，上の規則 1,2 を適用し，語彙的連鎖を再計算する．例えば，重複単語が削除され，要素が 2 つ以下になった連鎖は 2 番目の規則により連鎖と認められず削除される．こうして最終的に残ったものを語彙的連鎖とする．

Barzilay と Elhadad[1] は，名詞，複合名詞，名詞句を候補語とする．ただし，複合名詞や名詞句の場合，そのヘッドワードとなる名詞のみを使って語彙的連鎖を計算する．例えば，‘quantum computing’ を一つの候補語とするが，連鎖の計算は ‘computing’ だけを用いて行なう．

語と語の語彙的結束性関係を強さの順に extra-strong,strong,medium-strong の 3 種類に分け，それぞれの関係を以下のように定義している．

- extra-strong
  同一の単語が繰り返す場合

- strong
  WordNet 上で同じ synset にある単語もしくは直接リンクする synset 間にある単語で，単語間の距離が 7 文以内の場合

- medium-strong
  パスの長さが 2 以上の関係にある synsets 間に含まれる単語で，単語間の距離が 3 文以内の場合

Barzilay と Elhadad は，まず Hearst の手法 [6] により文書を談話セグメントに分割する．その後，各セグメントごとに語彙的連鎖を計算し，セグメント間にまたがる語彙的連鎖をマージするという 3 段階で語彙的連鎖構成を行なう．

セグメントに分割した後の語彙的連鎖の構成手順は次のようになる．まず，各セグメントごとに上の語彙的結束性関係に適合する語の可能な全ての組み合わせを生成し，語彙

的連鎖の候補とする．この処理によって非常に多くの語彙的連鎖の候補が生成され，同じ単語が複数の語彙的連鎖の候補に含まれることになるので，各連鎖候補について結合度を計算し，重複する連鎖候補内で最も強い結合度を持つ候補を選択する．結合度の強さは，WordNet のリンクの種類に応じて決定し，各リンクの点数の合計をその語彙的連鎖の結合度とする．Barzilay と Elhadad は，経験的な値として，同一の語の繰り返しと synonym 関係を 10，antonym 関係を 7，hyperonym と holonym 関係を 4 に設定している．この処理により，最終的に残った候補を談話セグメント内の語彙的連鎖とする．

最後に，隣接する 2 つのセグメント内の語彙的連鎖を調べ，2 つの語彙的連鎖が，表層形式の同一の候補語を同一概念として含んでいる場合は，その語彙的連鎖をマージする．これにより文書内の最終的な語彙的連鎖が決定される．

3 語彙的連鎖計算プログラム

本節では我々が開発した語彙的連鎖計算プログラムの使用方法について説明する．

3.1 配布プログラムと動作環境

本プログラムの配布パッケージは語彙的連鎖の構成基準に用いる情報の違いにより 3 種類あり，それぞれのパッケージに含まれるプログラムは次のようになる．

パッケージ 1. (chainers_th.tgz)
同一の語の繰り返し，または，シソーラスに基づいて語彙的連鎖を計算するプログラム：chainer_th, jum_chainer_th,

パッケージ 2. (chainers_co.tgz)
語の共起関係に基づいて語彙的連鎖を計算するプログラム:chainer_co, jum_chainer_co

パッケージ 3. (chainers_all.tgz)
フルパッケージ．chainer_th, jum_chainer_th, chainer_co, jum_chainer_co

それぞれのパッケージには，同じ構成基準を用いる場合でも，使用する形態素解析器の違いで，chasen1.51 に対応したものと，juman3.61 に対応したものの 2 種類のプログラムが含まれる．プログラム名との対応を表 4にまとめて示すので各自の目的に合わせてパッケージを選択して欲しい．

表 4: 使用条件とプログラム名の対応表

連鎖構成基準	形態素解析器	プログラム	パッケージ
同一の語の繰り返しに基づく語彙的連鎖	chasen1.51	chainer_th	1,3
同一の語の繰り返しに基づく語彙的連鎖	juman3.61	jum_chainer_th	1,3
シソーラスに基づく語彙的連鎖	chasen1.51	chainer_th	1,3
シソーラスに基づく語彙的連鎖	juman3.61	jum_chainer_th	1,3
語の共起に基づく語彙的連鎖	chasen1.51	chainer_co	2,3
語の共起に基づく語彙的連鎖	juman3.61	jum_chainer_co	2,3

また，各プログラムごとに，必要なソフトウェアが異なるので，注意する必要がある．

1. chainer_th (シソーラスや同じ語の繰り返し：chasen1.51 を使用)
   - chasen1.51 のライブラリが必要である．
   - また，シソーラスを使用する場合は,「分類語彙表」が必要である．
2. jum_chainer_th (シソーラスや同じ語の繰り返し：juman3.61 を使用)
   - juman3.61 がインストールされている必要がある．
   - また，シソーラスを使用する場合は,「分類語彙表」が必要である．

3. chainer_co (語の共起関係：chasen1.51 を使用)

   - chasen1.51 のライブラリが必要である．
   - geta がインストールされている必要がある．
   - また，共起計算に使用するコーパスが必要である．

4. jum_chainer_co (語の共起関係：juman3.61 を使用)

   - juman3.61 がインストールされている必要がある．
   - geta がインストールされている必要がある．
   - また，共起計算に使用するコーパスが必要である．

いずれのプログラムも，UNIX 系 OS 上で動作する．現在のところ動作が確認されている OS は，Solaris2.6,2.7, FreeBSD 2.2.8, FreeBSD 3.0, SunOS 4.1.x である．また，現在のところプログラムのコンパイルには，BSD 系の make,ar,ranlib と gcc が必要なため，各自の環境で用意されていない場合には，別途用意する必要がある．

3.2 インストール

各パッケージ毎にインストール方法を説明する．

パッケージ 1. ファイル名：*chainers_th.tgz*
シソーラス，同一の語の繰り返しに基づく語彙的連鎖
(chainer_th, jum_chainer_th) の場合

パッケージ 2. ファイル名：*chainers_co.tgz*
語の共起に基づく語彙的連鎖 (chainer_co, jum_chainer_co) の場合

パッケージ 3. ファイル名：*chainers_all.tgz*
両方を同時に使用する場合

3.2.1 chainers_th.tgz の場合

1. chainers_th.tgz を適当なディレクトリにコピーする．

2. chainers_th.tgz を解凍・展開する．
   例：
   % zcat chainers_th.tgz | tar xvf -

3. chainers ディレクトリが作成されるので，そこに移動する．
   % cd chainers

4. 以降このディレクトリを $CHAINTOOLS_ROOT とする．

5. $CHAINTOOLS_ROOT/Makefile.inc を修正する．
   以下の箇所を設定する (必須)．

   - OS，CC，AR，RANLIB
   - CHAINTOOLS_ROOT

   chasen 1.51 を使用する場合，

   - CHALIB= に chasen のライブラリ libchasen.a のある場所を指定する．

   分類語彙表を使用する場合，

   - BGH= に分類語彙表の索引 sakuin のある場所を指定する．

6. make を実行する．
   % make

7. $CHAINTOOLS_ROOT/bin に以下のプログラムが作成される．
   chainers_th, jum_chainers_th

以上でインストール終了．

3.2.2 chainers_co.tgz の場合

1. chainers_co.tgz を適当なディレクトリにコピーする．

2. chainers_co.tgz を解凍・展開する．
   例：
   % zcat chainers_co.tgz | tar xvf -

3. chainers ディレクトリが作成されるので，そこに移動する．
   % cd chainers

4. 以降このディレクトリを $CHAINTOOLS_ROOT とする．

5. $CHAINTOOLS_ROOT/Makefile.inc を修正する．以下の箇所を設定する (必須)．

   - OS，CC，AR，RANLIB
   - CHAINTOOLS_ROOT

   chasen 1.51 を使用する場合，

   - CHALIB= に chasen のライブラリ libchasen.a のある場所を指定する．

6. $CHAINTOOLS_ROOT/Makefile.geta を修正する．以下の箇所を設定する (必須)．

   - GETAROOT=

7. make を実行する．
   % make

8. $CHAINTOOLS_ROOT/bin に以下のプログラムが作成される．
   chainers_co, jum_chainers_co

以上でインストール終了．

3.2.3 chainers_all.tgz の場合

1. chainers_all.tgz を適当なディレクトリにコピーする．

2. chainers_all.tgz を解凍・展開する．
   例：
   % zcat chainers_all.tgz | tar xvf -

3. chainers ディレクトリが作成されるので，そこに移動する．
   % cd chainers

4. 以降このディレクトリを $CHAINTOOLS_ROOT とする．

5. $CHAINTOOLS_ROOT/Makefile.inc を修正する．以下の箇所を設定する (必須)．

   - OS，CC，AR，RANLIB
   - CHAINTOOLS_ROOT

   chasen 1.51 を使用する場合，

   - CHALIB= に chasen のライブラリ libchasen.a のある場所を指定する．

   分類語彙表を使用する場合，

   - BGH= に分類語彙表の索引 sakuin のある場所を指定する．

6. $CHAINTOOLS_ROOT/Makefile.geta を修正する．以下の箇所を設定する (必須)．

   - GETAROOT=

7. make を実行する．
   % make

8. $CHAINTOOLS_ROOT/bin に以下のプログラムが作成される．
   chainers_th, jum_chainers_th,
   chainers_co, jum_chainers_co

以上でインストール終了．

3.3 使用方法

4 種類のプログラム (chainers_th, jum_chainers_th, chainers_co, jum_chainers_co)，それぞれの使用方法を示す.

3.3.1 chainers_th

プログラムは以下のように使用する.
% ./chainers_th [options] < text_file

オプションの説明：

-i <num> ：
シソーラスの概念階層の 何レベルまで 使用するかを指定する.
0 の場合は，シソーラスを使用しないで，同じ語の繰り返しのみになる.

-j <num> ：
シソーラスの概念階層の 何レベルを 使用するか指定する.
-i とは異なり，指定したレベルのみを使う.
0 の場合は，このオプションを無効にする.

-t <num> ：
何語以上連なったら chain として出力するかを指定する.

-s ：
シンプルな出力

-1 ：
多義性が完全に解消できない場合に，最初の 1 語義だけを表示

-F <num> ：
filter mode 連鎖を作る品詞を指定する．組み合わせは以下のとおり
0 . 名詞 動詞 形容詞
1 . 名詞 —- 形容詞
2 . 名詞 —- ——
3 . 名詞 動詞 ——
4 . —- 動詞 ——
5 . —- 動詞 形容詞
6 . —- —- 形容詞

-D <dicidx> ：
使用するシソーラスを指定する．ただし事前にシソーラスを加工しておく必要がある.

-h ：
help を表示する；

出力フォーマット：

出力の 1 行目は，{ 総文数 総候補語数 } を表わす．2 行目以降は，1 レコードが 1 単語の情報を表し，複数レコードによって 1 種類の語彙的連鎖を表わしている．1 つの語彙的連鎖は空行の次のレコードから空行までである．1 レコードのフォーマットは以下のようになる．

- 同じ語の繰り返しに基づく場合 (-i 0 で chainer_th を使用)
  { 文番号 語番号 通し番号 見出し語 見出し語 付属する助詞 (または品詞)}
  例：
  5 9 63 要求 要求 wo
  19 11 249 要求 要求 ga
  (空行)

- シソーラスに基づく場合
  { 文番号 語番号 通し番号 概念番号 見出し語 付属する助詞 (または品詞)}
  例：
  3 7 45 165 難しい you
  19 13 251 165 困難 you
  (空行)

各フィールドの内,「通し番号」は文書の先頭から現在までに出現した候補語の通し番号を表す．また,「付属する助詞 (または品詞)」は，その単語が名詞である場合は直後の助詞の種類を示し，動詞か形容詞である場合には you(用言) と記述されている．表記と意味の対応を表 5に示す.

表 5: 付属する助詞または品詞一覧

	副助詞			格助詞				
表記	ha	mo	fuku	ga	wo	kaku	no	you
意味	「は」	「も」	他の副助詞	「が」	「を」	他の格助詞	なし	用言

3.3.2 jum_chainers_th

プログラムは以下のように使用する．

% $JUMAN -e < text_file | ./jum_chainers_th [options]

ここで「$JUMAN -e」は，juman3.61 を-e オプションで実行することを意味する．

オプションの説明：

i, j, t, s, 1, F, D, h の各オプションは，chainer_th と同じ．その他のオプションは以下のようになります.

-knp ：

knp の -tab 形式の出力を入力として使用する．

-W ：

語彙的連鎖の候補語以外の語もカウントし出力に反映させる．

出力フォーマット：

出力フォーマットは基本的に，chainer_th と同じで以下のようになります．

出力の 1 行目は，{ 文数 ターム数 } を表わす．2 行目以降は，1 レコードが 1 単語の情報を表し，複数レコードによって 1 種類の語彙的連鎖を表わしている．1 つの語彙的連鎖は空行の次のレコードから空行までである．1 レコードのフォーマットは以下のようになる．

- 同じ語の繰り返しに基づく場合 (-i 0 で jum_chainer_th を使用)
  { 文番号 語番号 通し番号 見出し語 見出し語 付属する助詞 (または品詞)}
  例：
  5 9 63 要求 要求 wo
  19 11 249 要求 要求 ga
  (空行)

- シソーラスに基づく場合
  { 文番号 語番号 通し番号 概念番号 見出し語 付属する助詞 (または品詞)}
  例：
  3 7 45 165 難しい you
  19 13 251 165 困難 you
  (空行)

各フィールドの説明は chainer_th と同じ．

3.3.3 chainers_co

プログラムは以下のように使用する．

% ./chainers_co -handle handle_name [options] < text_file

オプションの説明：

-handle <str> ：

geta で使用するコーパスの handle name を指定する．

-calc_mode <num> ：

共起スコアの計算式を指定する．

1 . TF.IDF： $df(x,y) \times log(N/(df(x) \times df(y)))$

2 . 相互情報量： $df(x,y) \times N/(df(x) \times df(y))$

10 . コサイン距離：$\sum(xi \times yi)/\sqrt{x_i{}^2} \times \sqrt{y_i{}^2}$

-threshold <num> ：

クラスタ (連鎖) 作成時の共起スコアの閾値を指定する．

-N <num> ：
　　共起スコア計算に使用するコーパスの総文書数.

-F <num> ：
　　filter mode
　　1 . 動詞を抜かす
　　2 . 名詞だけにする
　　3 . 形容詞を抜かす
　　4 . 動詞だけにする
　　5 . 名詞を抜かす

-h ：
　　help を表示

出力フォーマット：
　出力の 1 行目は，{ 文数 ターム数 } を表わす．2 行目以降は，1 レコードが 1 単語の情報を表し，複数レコードによって 1 種類の語彙的連鎖を表わしている．1 つの語彙的連鎖は空行の次のレコードから空行までである．1 レコードのフォーマットは以下のようになる．

- 語の共起関係に基づく場合
  { 文番号 語番号 通し番号 見出し語 付属する助詞 (または品詞)}
  例：
  1 10 10 携帯 no
  1 12 12 電話 no
  (空行)

　各フィールドの説明は chainer_th と同じ.

3.3.4 jum_chainers_co

　プログラムは以下のように使用する.
　% $JUMAN -e < text_file | ./jum_chainers_co -handle handle_name [options]
ここで「$JUMAN -e」は，juman3.61 を-e オプションで実行することを意味する.

オプションの説明：
handle, calc_mode, threshold, N, F, h の各オプションは，jum_chainer_co と同じ．その他のオプションは以下のようになる.

-knp ：
　　knp の -tab 形式の出力を入力として使用する.

-W ：
　　語彙的連鎖の候補語以外の語もカウントし出力に反映させる.

出力フォーマット：

出力フォーマットは，chainer_co と同じで以下のようになる．

出力の 1 行目は，{ 文数 ターム数 } を表わす．2 行目以降は，1 レコードが 1 単語の情報を表し，複数レコードによって 1 種類の語彙的連鎖を表わしている．1 つの語彙的連鎖は空行の次のレコードから空行までである．1 レコードのフォーマットは以下のようになる．

- 語の共起関係に基づく場合
  { 文番号 語番号 通し番号 見出し語 付属する助詞 (または品詞)}
  例：
  1 10 10 携帯 no
  1 12 12 電話 no
  (空行)

各フィールドの説明は chainer_th と同じ．

おわりに

本稿では，語彙的結束性に基づく語彙的連鎖の説明とその計算方法について示し，我々の作成したプログラムについて説明した．また，他の研究における語彙的連鎖計算の方法についてもまとめた．

なお語の共起関係に基づく語彙的連鎖を計算するプログラムでは，コーパス中の語の出現頻度の取得に，(株) 日立製作所中央研究所において開発された汎用連想検索エンジン GETA(Generic Engine for Transposable Association)[16] を使用している．

参考文献

[1] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proc. of the ACL Workshop on Intelligent Scalable Text Summarization*, pp. 10–17, 1997.

[2] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[3] S. J. Green. *Automatically Generating Hypertext by Computing Semantic Similarity*. PhD thesis, University of Toronto, Graduate Department of Computer Science, 1997.

[4] U. Hahn. On Text Coherence Parsing. In *Proc. of the 14th International Conference on Computational Linguistics*, pp. 25–31, 1992.

[5] H.A.K. Halliday and R. Hasan. *Cohesion in English*. Longman, 1976.

[6] M.A. Hearst. Multi-Paragraph Segmentation of Expository Texts. Technical Report UCB/CSD-94-790, UC Berkeley Science Technical Report, 1994.

[7] J. R. Hobbs. Coherence and coreference. Technical Report 168, SRI International Technical note, 1978.

[8] J. R. Hobbs. *literature and cognition*. Center for the study of Language and Information, 1990.

[9] J. Morris and G. Hirst. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, Vol. 17, No. 1, pp. 21–48, 1991.

[10] M. Okumura and T. Honda. Word sense disambiguation and text segmentation based on lexical cohesion. In *Proc. of the 15th International Conference on Computational Linguistics*, pp. 755–761, 1994.

[11] R. Roget. *Roget's International Thesaurus, Fourth Edition*. Harper and Row Publishers Inc., 1977.

[12] D. St-Onge. Detecting and Correcting Malapropisms. Master's thesis, University of Toronto, 1995.

[13] M. A Stairmand. Textual Context Analysis for Information Retrieval. In *Proc. of 20th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, pp. 140–147, 1997.

[14] 佐々木一朗, 増山繁, 内藤昭三. 結束チャートの自動生成と日本語文章の語彙的結束構造解析へ の応用. 情報処理学会研究会資料 NL95-8, pp. 57–64, 1993.

[15] (株) 日本電子化辞書研究所. EDR 電子化辞書仕様説明書, 1993.

[16] 高野明彦, 丹羽芳樹, 西岡真吾, 岩山真, 久光徹, 今一修, 櫻井博文, 徳永健伸, 奥村学, 野本忠司. 汎用連想計算エンジンの開発と大規模文書分析への応用. IPA 独創的情報技術育成事業研究成果報告書, 1999.

[17] 大野晋, 浜西正人. 角川類語新辞典. 角川書店, 1981.

[18] 松本裕治, 北内啓, 山下達雄, 平野善隆, 今一修, 今村友明. 形態素解析システム『茶筌』version 1.5 使用説明書, 1997.

[19] 中野洋（編）. 分類語彙表. 国立国語研究所, 1990.