係り受けの 2 部グラフと共起関係を利用した同義表現抽出

上野 友司 [†] 森 辰則 [‡] 木戸 冬子 [††] 中川 裕志 [‡‡]

[†] 横浜国立大学大学院 環境情報学府 [‡] 横浜国立大学大学院 環境情報研究院
{tomot,mori}@forest.eis.ynu.ac.jp

[††] マイクロソフト株式会社 [‡‡] 東京大学 情報基盤センター
fkido@microsoft.com nakagawa@dl.itc.u-tokyo.ac.jp

1 はじめに

テキストマイニングの基本機能の一つに概念の抽出があるが、そこでは語の多義性の解消とともに異表記同義語の認定が重要となる。例えば、テキストマイニングの基本ツールとしてよく用いられる情報検索システムでは、利用者の入力する検索質問中の表現と文書に実際に登場する表現との相違が検索効率に影響を与える。利用者が「A をインストールしたい」と尋ねたときに、文書中にある「A をセットアップする方法」を検索することは容易ではなく、利用者が自分の語彙を元に順次言い換えを行っていくのが現実である。そこで検索システム側の前処理として異表記同義語の認定を行い、検索語を同様の現象を示すと思われる索引語へ自動的に変換することで、個人個人の語彙に左右されない検索システムを作ることができると考えられる。

システム側で異表記同義語を認定するには同義語辞書を用いる。その同義語辞書は通常、人手によって作られているがその整備には多くの手間と時間がかかる。さらに人手で作られていることで、どうしても作り手の背景知識に偏ったものとなってしまい、不特定多数の利用者の背後にある膨大な背景知識に対応することは難しい。そこで本稿では同義語辞書を一から作成する際の支援を目的として、同義表現の候補をその分野に属する大量のコーパスから抽出する手法を示す。大量のコーパスから機械的に抽出することで、様々な背景知識を網羅した表現を抽出でき、また辞書作成のコストも削減できると考えられる。

同義表現抽出と関連のある研究としては、仁井ら [1] や山本ら [2]、中渡瀬ら [3] の研究がある。これらは名詞や動詞に限定した類義語もしくは換言知識の抽出手法を提案しているが、本研究では抽出の対象を名詞や動詞などに限定せず、基本的に全ての品詞を対象とする手法を検討する。なぜならば本研究の目的である同義語辞書作成の支援の際には、分野固有の名詞はもちろんのこと、それらの名詞と関連して出現する分野固有の動詞や形容詞、形容動詞なども網羅する必要があるからである。

2 同義表現の抽出手法

2.1 処理の概要

本研究では膨大な背景知識を網羅するため、大量のコーパスから全ての品詞を対象とした同義表現を抽出する。まずはじめに、文書中に出現するすべての係り受けを抽出し、その係り受けの連接関係から 2 部グラフを作成する。この 2 部グラフの中で完全 2 部グラフを形成している部分を見つけることで、同義表現と思われる候補を大まかに抽出することを行う (2.2節)。次に、完全 2 部グラフによって抽出された文節組において、その組の表現の近さの度合を類似度として算出することを行う。類似度の算出手法としては、2 部グラフ作成時に解析した係り受け情報をもとに、計算対象となる文節に係る、もしくは文節が係っている文脈の頻度を情報ベクトルに変換し、そのベクトル間の内積で求める (2.3節)。ここまでが本稿で提案する同義表現抽出の基本的手法である。加えて抽出精度を向上させる手法もいくつか検討する (3章)。

2.2 完全 2 部グラフ

まずはじめに、コーパス中に出現する文節間の係り受け関係をすべて抽出する処理を行う。この処理を効率的に行うために拡張版 PrefixSpan[4] を用いる。なお PrefixSpan の前処理として、それ単体では具体的な述部を構成しない、「できる」「行う」「出る」「ある」「なる」などの動詞およびその活用形は、表層表現上でこれらの直前に出現している文節とまとめたものを一つの文節として取り扱う。すべての係り受け関係が得られたら、その情報を用いて各文節を頂点、係り受け関係を辺とする 2 部グラフを作成することができる。作成される 2 部グラフの例を図 1に示す。

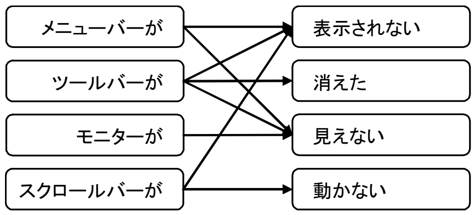

図 1: 係り受け関係の 2 部グラフ

この 2 部グラフ中で完全 2 部グラフを形成しているところに注目してみる。完全 2 部グラフとは 2 部グラフの頂点集合を X、Y とするとき X(または Y) の任意の点が Y(または X) の全ての点と接続されているものである。図 1において完全 2 部グラフとなっているのは図 2に示す部分である。

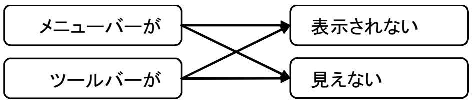

図 2: 完全 2 部グラフ 1

完全 2 部グラフを形成している頂点集合は互いに共通の要素と係り受け関係を持っているため、左側 (始頂点集合) および右側 (終頂点集合) の文節組それぞれが言い換えができる可能性のある文節の組、すなわち、同義文節の候補の組となる。図 2の場合においては (メニューバーが ツールバーが) の組、(表示されない 見えない) の組がそれぞれ同義文節の候補の組にあたる。これらの始頂点集合、終頂点集合を同義表現の候補として 2 部グラフから抽出していく。

ただし係り受けの関係においては、ある 1 つの文節に対して複数の文節が係る場合がある。それによって異なる格を持つ文節同士で完全 2 部グラフを構成してしまうことがある[1]。このようにして作られる完全 2 部グラフは同義表現の候補となっているとはいい難い。そこで完全 2 部グラフを探索する際に係り受けの種類 (格、接続形態など) も考慮し、同じ係り受けの種類を持つもの同士を頂点集合とするようにする。

図 3に完全 2 部グラフによって抽出された、同義表現の候補の一部を示す。

(動作が 動きが) (遅くなる 悪くなる)
(cdrom を cd-rom を) (挿入しても 入れても)
(パソコンに マシンに) (移したい 入れたい)
(パソコンに pc に) (導入して インストールして)
(setup が セットアップが) (行えない 進まない)
(強制終了が エラー msg が) (起こる 発生する)

図 3: 同義表現の候補

完全 2 部グラフを形成する頂点集合の位数は 2 以上であるので、完全 2 部グラフの最小部分は $(a\ b)(x\ y)$ という 2-2 の形である。$(a\ b)(x\ y)$ と $(a\ b)(y\ z)$ という 2 つの完全 2 部グラフがあった場合、これらを併合することで $(a\ b)(x\ y\ z)$ という 1 つの完全 2 部グラフにまとめることができるが、本稿では 2.3節で行う類似度計算を考慮して頂点集合の位数 2 のものを順時出力していく形式を取っている。

2.3 文脈の共起情報を用いた類似度の計算

2.2節で同義表現の候補となる文節の組 (候補文節の組) が抽出された。本節では候補文節に係る、もしくは候補文節が係っている文節郡の頻度情報を用いて類似度を算出し候補間の順位付けを行う手法を述べる。候補として $(a\ b)(c\ d)$ が抽出された場合、a と b、c と d の類似度を計算し降順に出力させる。

類似度を計算するにあたり、ある文節 a に注目したとき、その文節 a に対して直接係る文節 pre_i の集合を前文脈、a が直接係る文節 $post_j$ の集合を後文脈と考え、文脈中の各文節の重みを成分にした次のベクトルを考える。

$$CV_a \quad = \quad \{w^a_{pre_1}, \dots, w^a_{pre_m}, w^a_{post_1}, \dots, w^a_{post_n}\} \qquad (1)$$

文節間の文脈の類似度は対応するベクトルの内積で求める。一般的にベクトル間の類似度は $cosine$ 値を用いるが、本研究では文脈中に出現している各文節の頻度も考慮するために内積を採用した。

$$sim(a, b) = CV_a \cdot CV_b \qquad (2)$$

上記ベクトルの各成分の値は次のように求める。

$$\begin{aligned} w^a_{pre_i} &= tf(a, pre_i) icf(pre_i) && (3)\\ w^a_{post_j} &= tf(a, post_j) icf(post_j) && (4)\\ tf(a, pre_i) &= log_2(freq^a_{pre_i}) + 1 && (5)\\ tf(a, post_j) &= log_2(freq^a_{post_j}) + 1 && (6)\\ icf(pre_i) &= log_2 \frac{pre\_N}{cf(pre_i)} && (7)\\ icf(post_j) &= log_2 \frac{post\_N}{cf(post_j)} && (8) \end{aligned}$$

ただし

$freq^a_{pre_i}$	:	文節 a に係っている文節 pre_i の出現頻度
$freq^a_{post_j}$	:	文節 a が係っている文節 $post_j$ の出現頻度
$cf(pre_i)$	:	文節 pre_i が前文脈に出現する頻度
$cf(post_j)$	:	文節 $post_j$ が後文脈に出現する頻度
pre_N	:	全前文脈数
$post\_N$	:	全後文脈数

式 (7)、(8) で示した icf は $idf(Inverse\ \ Document\ Frequency)$ の文書 $(Document)$ を文脈 $(Context)$ に置き換えたものである。係り受け関係の頻度における、ある文節が出現している割合の逆数で求める。つまり多くの文節の文脈に出現している文節ほどその値は小さくなり、逆に少ないと値は大きくなる。本研究のように全品詞の係り受け関係からベクトルを作成した場合、多種類の文節が係り易い「こと」「方法」などの名詞、同じ対象に対する複数の動作表現 (「ファイルを」に対する「コピーする」「削除する」「消す」「作成する」「張り付ける」など) が多数存在し、またその頻度も大きい。これらの係り受けし易い文節は特定の文脈を判別する時の手がかりとはならないので、その頻度の影響を抑える指標として icf を用いた。

ここまでが同義表現抽出の基本手法である。次章ではその抽出精度を向上させる手法をいくつか検討する。

3 抽出精度向上の向上に関する検討

3.1 類似度計算における文脈となる文節の制限

文節によっては比較的ゆるやかな係り受け関係しか構成しないものもある。そのため類似度計算に用いる文脈に制限をかけることで、抽出の精度が向上することが期待される。ここでは一つの節中における係り受

[1] (excel を PC に) (セットアップする インストールする) など

け関係よりも、節を越えた係り受けを重視することを行った。具体的に「名詞＋の」や、副詞、形容詞などを文脈から除いて、動詞ならびに「名詞＋ノ以外の格助詞」で限定したものを文脈として扱うことにする。

3.2 上位文節組の併合を用いたフィードバック

類似度計算によって文節間の類似度が数値として表される。この数値が高く算出された文節の組を同義文節として同一視し、一連の計算を再度行うことで抽出の精度を上げる手法がいくつか考えられる。

まず一つめは、類似度計算にのみフィードバックをかける手法である。文節 x と文節 y が類義文節と判定された場合において文節 a の前文脈にそれらが存在していたとき、

$$CV_a = \{w^a_{pre_x}, w^a_{pre_y}, w^a_{pre_z}, \ldots\} \qquad (9)$$

を

$$CV_a = \{w^a_{pre_{x+y}}, w^a_{pre_z} \ldots\} \qquad (10)$$

と書き換える。この処理を行ったあとに類似度を再計算することによって、文脈に同義表現として判定された語を含む文節組の類似度が上昇すると考えられる。

次に類似度計算の結果を完全 2 部グラフ形成にまでフィードバックをかける手法がある。

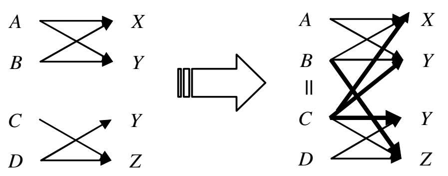

図 4: 新しい完全 2 部グラフの発見

図 4左で $(A\ B)(X\ Y)$ は完全 2 部グラフを形成しているが、$(C\ D)(Y\ Z)$ は辺が一つ足らず完全 2 部グラフではない。ここで類似度計算の結果、文節組 B と C が近いものであると判定されたとする。

この時文節組 B と C を同じものとみなし、B が繋がっていたものは C も繋がっているとみなす。このように類似度計算の結果を完全 2 部グラフの発見過程にフィードバックすると図 4右に示す太線が新たに追加され、これで新しい完全 2 部グラフ $(B\ C)(Y\ Z)$、$(C\ D)(Y\ Z)$ が作られる。類似度の再計算の際、これらの新しい候補も加えられ類似度計算の対象とする。新しい候補が追加されることにより、再現率が向上すると考えられる。

4 評価実験

2章、3章で述べた手法を実装して実験を行い、その結果の評価を行った。手法の実装には Perl を用いた。

4.1 使用データ

コーパスにはマイクロソフト製品の問い合わせ情報 128 日分 (17 万件) を用いた。これらは一つ一つの案件を 1 文でまとめたものであり、同じ現象の案件でも書き手ごとに表記が異なる場合がある。

4.2 評価基準

本手法による同義文節組の抽出がどの程度正しく行われたかを評価するために、実験結果に対する検査を行った。抽出結果の上位から 700 件に対して、コーパスの関連する分野に精通している人間が同義か否かの判定を以下の三段階で評価した。[2]

1. 前後にどんな文脈がきても同義として扱える
2. 前後に出現する文脈によっては同義として扱える
3. どんな文脈でも同義となり得ない

なお、類義語や反意語に対しては「3」と判定している。ただし本稿では「類義語」ならびに「反意語」をそれぞれ、「シソーラス上で兄弟関係など距離の近い場所に存在する語[3]」、「対となる事物を指し示す語[4]」と定義する。

4.3 完全 2 部グラフを用いた同義表現の候補抽出

完全 2 部グラフを構成した文節組の候補となるものは全部で 81319 組であった。類似度計算では最も精度が高くなると考えられる「文脈に制限を加えた類似度計算」の手法を採用し、前述の評価基準に従って判別した。それぞれの評価 1 と評価 2 の例を下記に示す。

表 1: 評価 1 の文節組の例

起動できない	起動しない
知りたい	教えてほしい
関して	ついて
強制終了	フリーズ
マシンに	PC に
無くなった	消えた
使用する	使う
winxp に	xp に

表 2: 評価 2 の文節組の例

開くと	クリックすると
表示される	要求される
画像を	図を
データが	ファイルが
変更する	切り替える
間隔を	行間を
方法を	手順を
値を	文字を

評価 1 と判定されたもので特に上位には、(教えて欲しい 教えてほしい) などの漢字 / かなの表現の差異や (表示される 表示されてしまう) などの補助動詞の差異による文節組が多数見られた。

評価 2 と判定されたものには (画像を ファイルを) などのように is-a 関係にある組が多く見られた。これらは前後に出現する文脈によって対象の範囲が限定されることで、同義のものと同様の働きを持つものだと考えられる。

また評価 3 の中には 4.2で述べた類義語、反意語がいくつか見られた。これらの例を表 3と表 4に示す。

表 3: 類義語の例

コピーしたい	張り付けたい
フォルダを	ファイルを
fd に	cd-r に
図を	表を
fax の	メールの

表 4: 反意語の例

表示したい	消したい
起動できない	終了できない
表示したい	非表示にしたい
開く	閉じる
削除したい	保存したい

[2]具体的には第三著者が評価を行っている。
[3]例えば「文字」と「数字」
[4]例えば「開く」と「閉じる」

前後に出現する文脈の情報だけで類似度を計算しているため、どうしても類義語や反意語、固有名詞を含む名詞句などは類似度が高くなる。本稿では「どんな文脈でも同義となり得ない」と評価したが、一方で、これらは有用な知識になることがある。例えば「～を表示したい」といった要求があった場合には、その関連知識として対になる操作である「～を消す」ための方法や手段も併せて提示すると利用者の理解を一層深めることができると考えられる。

取り出した文節組の件数とその精度の変化を図 5に示す。

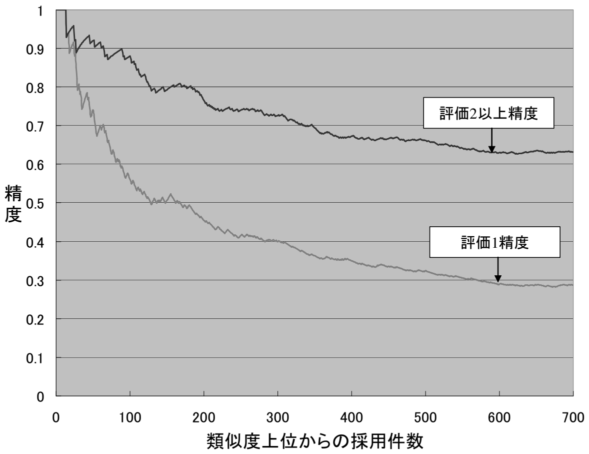

図 5: 抽出精度

上位の 200 組においては文脈に依存せずに同義といえる文節組が 45.50%、文脈によって同義といえる文節組も合わせると 76.50% の精度となった。

4.4 上位文節組の併合を用いたフィードバック

この節では 3.2節で提案した 2 つのフィードバックによる効果を評価してみる。類似度計算が終了するごとに類似度が最上位の 1 文節組を同義文節とし、それぞれのフィードバックの処理を行う。これを 10 回繰り返した。なお 2 回目以降の繰り返しでは、一度同義文節と判定されたものは同義文節の候補から除外する。

フィードバックをかけたことにより類似度の上位 700 単語組に新しく加わった文節組を 4.3節と同様の手法で評価し、上位 700 以内の評価の構成比の変化を求めた。それぞれのフィードバックの効果の結果を表 5に示す。

どちらのフィードバックにおいても評価 3 の数が減っており、適合率の向上を目的とした類似度計算へのフィードバックでは評価 1、評価 2 がともに増え、再現率の向上を目的とした 2 部グラフ構成へのフィードバックでは評価 1 は減少したが、評価 2 は増加している。よって効果の方向性は正しいと考えられるが、それぞれの追加事例件数が 700 件に対して、 14 件と 27 件というように、その効果の大きさは非常に小さいものとなった。

表 5: 各フィードバックによる構成比の変化

類似度計算へのフィードバック			
	新規追加単語組数 14		
	評価 1	評価 2	評価 3
件数 (件)	+1	+4	-5
構成比の差 (ポイント)	+0.2	+0.6	-0.8

2 部グラフ構成へのフィードバック			
	新規追加単語組数 27		
	評価 1	評価 2	評価 3
件数 (件)	-3	+4	-1
構成比の差 (ポイント)	-0.4	+0.6	-0.2

5 まとめと今後の課題

本稿では、特定分野に属する文書群から完全 2 部グラフを用いて同義表現となる文節を大まかに抽出し、前後に出現する文脈の情報を用いて表現の近さの度合を類似度として求める手法を提案した。さらにその上位の結果を一連の処理にフィードバックし抽出の精度を上げる手法も提案した。完全 2 部グラフの構成から抽出した同義表現の候補にはノイズも数多く残っているが、指標 $tf{\cdot}icf$ を用いて文脈の類似度を求めることで、ランクの上位から 200 件においては比較的近い表現といえるものが 76.50% の精度で取り出すことができる。これによって一般的な同義表現および特定の文脈を持つことで同義表現になり得るものを容易に集めることができ、同義語の集合を求める際の支援として適当だと考えられる。

今後の課題としては、まず係り受けの情報を細分化し類似度計算に反映させる手法の検討がある。さらに、複数の文節を対象とした同義文節の抽出を行う手法の検討も挙げられる。

6 謝辞

拡張版 PrefixSpan のプログラムを提供して頂いた奈良先端科学技術大学院大学の工藤拓さんに深く感謝致します。

参考文献

[1] 仁井康夫, 酒井浩之, 吉田辰巳, 増山繁. 動詞間の換言知識の自動獲得. 言語処理学会第 9 回年次大会発表論文集 B3-2, 言語処理学会, 3 月 2003.

[2] 山本和英. テキストからの語彙的換言知識の獲得. 言語処理学会第 8 回年次大会発表論文集 A6-5, 言語処理学会, 3 月 2002.

[3] 中渡瀬秀一. 複合語からの類義語抽出法. 情報処理学会デジタル・ドキュメント研究会, DD-32-6, pp. 39–46, 3 月 2002.

[4] 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. 情報処理学会自然言語処理研究会, NL-148-10, pp. 65–72, 3 月 2002.