

用言と直前の格要素の組を単位とする格フレームの自動獲得

河原大輔 黒橋禎夫
京都大学大学院 情報学研究科
〒 606-8501 京都市左京区吉田本町
{kawahara,kuro}@pine.kuee.kyoto-u.ac.jp

あらまし

本稿では、生コーパスから格フレームを自動的に獲得する手法を提案する。まず、生コーパスを解析し、解析結果から確信度の高い係り受けを収集する。次に、用言の意味的曖昧性に対処するために、用言と直前の格要素の組を単位として、生コーパスから用例を収集し、それらのクラスタリングを行う。また、得られた格フレームを用いて格解析を行い、その結果を示す。

Case Frame Construction by Coupling the Predicate and its Adjacent Case Component

Daisuke Kawahara Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 JAPAN
{kawahara,kuro}@pine.kuee.kyoto-u.ac.jp

Abstract

This paper describes a method to construct a case frame dictionary automatically from a raw corpus. First, we parse a corpus and collect reliable examples from the parsed corpus. Secondly, to deal with semantic ambiguity of a predicate, we distinguish examples by a predicate and its adjacent case component and cluster them. We also report on an experimental result of case structure analysis using the constructed dictionary.

1 はじめに

日本語には語順の入れ替わり、格要素の省略、表層格の非表示などの問題があり、単純な係り受け解析を行っただけでは文の解析として十分とはいえない。例えば、「ドイツ語も話す先生」という文の場合、係り受け構造を解析しただけでは、「ドイツ語」と「話す」、「先生」と「話す」の関係はわからない。このような問題を解決するためには、用言と格要素の関係、例えば、「話す」のガ格やヲ格にどのような単語がくるかを記述した格フレームが必要である。さらに、このような格フレームは文脈処理(照応処理、省略処理)においても必須の知識源となる。

格フレーム辞書を構築する方法のひとつは、人手による作成である[3, 6]。しかし、その場合には非常に大きなコストがかかり、カバーレージの大きな辞書を作成、保守することは難しい。また、これまでの人手による辞書では、格と同じ振る舞いをする「によって」、「として」などの複合辞や、「～が～に人気だ」のように名詞+判定詞の格フレームを取り扱っているものはほとんどない。

そこで、格フレーム辞書をコーパスから自動学習する方法を考える必要がある。しかし、格フレームの学習には膨大なデータが必要となり、現存するタグ付きコーパスはこのような目的からは量的に不十分である。そこで、本論文では、格フレーム辞書をタグ情報が付与されていない大規模コーパス(生コーパス)から自動的に構築する手法を提案する。

格フレーム辞書を生コーパスから学習するためには、まず、生コーパスを構文解析しなければならないが、ここで解析誤りが問題となる。しかし、この問題はある程度確信度が高い係り受けだけを学習に用いることでほぼ対処することができる。むしろ問題となるのは用言の意味の曖昧性である(これはタグ付きコーパスから学習する場合にも問題となる)。つまり、同じ表記の用言でも複数の意味をもち、意味によってとりうる格や体言が違ふことがあるので、用言の意味ごとに格フレームを作成することが必要である。本論文では、これに対処するために、

1. 用言とその直前の格要素の組を単位として用例をまとめ、
2. さらに、それらのクラスタリングを行った。

2 格フレーム学習の種々の方法

生コーパスからの格フレーム辞書の構築の過程は以下のとおりである(図1の点線で囲まれた部分)。

1. コーパスのテキストに対して、KNP[5]を用いて構文解析を行い、その結果から、ある程度信頼できる用言・格要素間の関係を取り出す。ここで取り出すデータを**用例**と呼ぶ。
2. 抽出した関係を用言と直前の格要素の組ごとにまとめる。このようにして作成したデータを**用例パターン**と呼ぶ。
3. シソーラスを用いて、用例パターンのクラスタリングを行う。この結果できたものを**用例格フレーム**と呼び、本研究ではこれが最終的に得られるものである。以下では「記事」、「本」、「先」などの格要素になる単語を**格用例**、用例格フレームにおけるある格の格用例の集合、例えば「読む」のひとつめの用例格フレームのヲ格の格用例集合、{「記事」、「本」}を**格用例群**と呼ぶ。

次に、格フレームに関連するさまざまなデータ処理を図1に沿って議論する。

まず、図1のIの用例をそのまま個別に使うことが考えられるが、この場合データスパースネスが問題になる。例えば、

- (1) a. 図書館で本を読む
b. 家で新聞を読む

という2つの用例がコーパスにあったとしても、「図書館で新聞を読む」という表現が妥当であるかどうかはわからない。

一方、用例を二項関係に分割すると、図1のIIのような共起データを作ることができる。これは統計パーサによって用いられているデータ形式であり、データスパースネスの問題を回避することができる[1]。しかし、その副作用として用言の意味の曖昧性の問題が生じる。例えば、

- (2) a. 彼が天を仰ぐ
b. 彼が先生に指示を仰ぐ

この2つの用例から「天を仰ぐ」、「先生に仰ぐ」、「指示を仰ぐ」という共起データが得られるが、これらのデータだけでは「先生に天を仰ぐ」のような間違った表現を許すことになる。

また、図1のIIIのように用例を単純にまとめたものも、もっている情報は共起データと同じであ

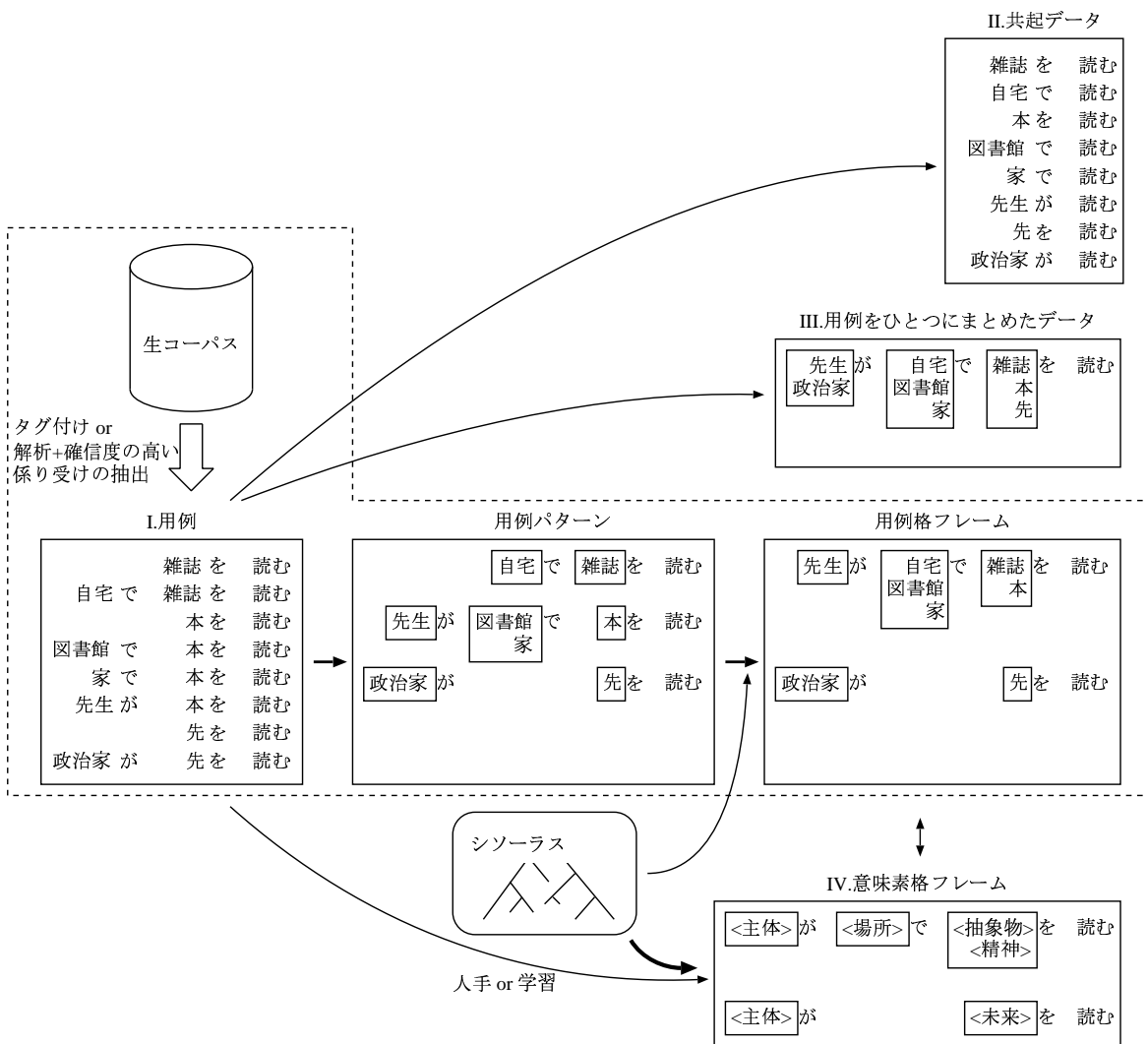


図 1: 格フレームに関連するさまざまなデータ処理

り、やはり用言の意味の曖昧性が問題となる。

本手法は、用言とその直前の格要素を組にして扱うという方法で、意味の曖昧性の問題を解消しつつ、データスパースネスにも対処している。

一方、用例格フレームの格要素を意味素に抽象化することにより、図 1 の IV に示したような**意味素格フレーム**を考えることができる。しかし、これを人手で作成する場合は、1 章で述べたような問題がある。宇津呂らは意味素格フレームをコーパスから学習している [4]。本研究との違いは、学習にタグ付きコーパスを用いていること、また、用言の意味の曖昧性を扱っていないことである。

また、意味素格フレームの作成はシソーラスに深く依存してしまう。一般に、シソーラスにおいて、同義語、類義語を定義するには問題ないが、それを

大きな体系にまとめる部分では大きく主観に依存してしまう。意味素格フレームの作成は、そのような大きな体系に左右されるので、シソーラスの不整合な問題の影響を受けることになる。本研究では、シソーラスの比較的安定した部分だけを用いており、シソーラスによる悪影響はほとんど出ない。

3 用例の収集

コーパスを構文解析した結果から、図 1 に示したような用例の収集を行う。単語を個別に扱うことにあまり意味がなく、意味が明確な格要素はクラスとして扱う。また、質の高い用例を収集するために、コーパスの解析結果から確信度の高い係り受けを抽出するというを行う。

3.1 収集に関する条件

用例を収集するときに、格要素、用言のそれぞれに以下のような条件を設定する。

格要素の条件

収集する格要素は、ガ格、ヲ格、ニ格、ト格、デ格、カラ格、ヨリ格、無格とする。また、次のものを新たな格として扱う。

時間格 ニ格、無格、カラ格、マデ格で、意味素「時間」(3.2節で述べる)をもっている格要素はまとめてひとつの格にする。これは、格フレームを作るときには、その用言が時間に強く関係しているかどうか重要であり、そのような格要素は表層格をそのまま扱うよりも、表層格をまとめてひとつの格にしたほうが望ましいからである。

例: 3時に、来年から

複合辞 格と同じように振る舞う複合辞も、ひとつの格として扱う。

例: ～をめぐって、～によって、
～について、～として

次のような格要素は収集に用いない。

- 提題助詞をもつ格要素と用言の連体修飾先は、表層格が明示されていないので収集に用いない。

例: ～を提案している 議員 が～
その 議員 は～を提案した。

- ニ格、デ格で副詞的に使われる格要素は、係る用言との関係が任意的であるので収集から除外する。

例: ために、無条件に、うえで、せいで

- KNP では、「～では」、「～でも」はデ格、「～には」、「～にも」はニ格の格要素として扱われるが、副助詞、あるいは従属節の場合もあるので収集の対象から除外する。

例: 足の1本 でも 折ってやろうかと思った。
育成しないこと には 世界で通用しない。

格要素として、文節内の最後の自立語を収集に用いる。格要素が複合名詞の場合には、そのなかで最後の自立語がもっとも意味的に重要であると考えられるからである。

用言の条件

収集する用言は動詞、形容詞、名詞+判定詞とする。名詞+判定詞として収集する用言には体言止めの名詞も含む。ただし、以下のような用言は収集に用いない。

- 用言が受身、使役、「～もらう」、「～たい」、「～ほしい」、「～できる」の形であれば、格の交替が起こり、格と格要素の関係が通常の場合と異なるので収集に用いない。

- 「～で」は、判定詞かデ格かの自動判定が難しいので、KNP が判定詞と認識しても、用言として収集に用いない。

例: 彼は 京都で、試験を受け… (助詞)
彼が好きな町は 京都で、… (判定詞)

3.2 格要素の汎化

個別の単語を扱うことにあまり意味がなく、明確な意味を考えることできる格用例はクラスとしてまとめて扱う。この汎化したクラスを以下のように3種類設定した。この場合、格用例として単語のかわりにクラスを記述する。

時間

- 品詞細分類が時相名詞
例: 朝, 春, 来年
- 時間助数辞を含む文節
例: 年, 月, 日, 時, 分, 秒
- 「前」、「中」、「後」という接尾辞をもち、自立語がシソーラス上の「場所」の意味属性をもたない文節
例: 会議中, 戦争後, 書く前に

数量

- 数詞を含む文節
例: 1, 2, 一, 二, 十, 百
- 数詞と、「つ」、「個」、「人」のような助数辞を含む文節については、「<数量>つ」、「<数量>個」、「<数量>人」のように数量クラスと助数辞のペアにして学習する。
例: 1つ → <数量>つ
2個 → <数量>個

補文

- 引用節「～と」、連体修飾+形式名詞 またはそれに準ずる表現 (～の～, ～くらい～,)

例: 書くと、書いたことを、書くのを、
書くくらいが

例えば、

(3) 30日に総理大臣がその2人に賞を贈った。
という文からは、

<時間>:時間格 大臣:が

<数量>人:に 賞:を 贈る

という用例を学習する。

3.3 確信度の高い係り受けの抽出

コーパスを構文解析した結果から用例を収集するときに問題となるのは、解析結果に誤りが含まれていることである。そこで、誤りの影響を軽減するために、解析の精度が低い係り受けは捨てて、ある程度確信度が高い係り受けを格フレームの収集に用いる。

KNPでは、次のような優先規則によって文節の係り先を決定している。

1. 文中の強い区切りを見つけることによって、係り先の候補の絞り込みを行う(ここで候補がひとつになるなら、係り先をそれに決定する)。
2. 係り先の候補の用言のうち、格要素の係り先にならないことが多い用言を候補から除外する。
3. “読点のない文節はもつとも近い候補に係り、読点のある文節は2番目に近い候補に係る”という優先規則に従って、候補の中から係り先を決定する。

用例の収集では、1は信頼し、2と3は信頼しない(多くの場合正しいが、誤っていることもある)こととする。つまり、1で候補がひとつになり決定される係り受けは用例の収集に用い、2や3の処理が適用された係り受けは収集に用いない。例えば、

- (4) その著者は買いたい本をたくさん見つけたので、東京へ送った。

この例において、まず、まったく曖昧性なしに取り出すことができる用例は、文末にある「東京へ送った」だけである。ここでさらに、「～ので」はKNPによって強い区切りであると認識され、「本を」の係り先の候補は「見つけた」の1つしかないので、この用例を取り出す。

次に、2において、係り先の候補から除外した用言は、場合によっては係り先になる可能性があるので、このときの用例は収集しないことにする。例えば、次の例のように、形容詞のすぐ後に強い用言がある場合、このような形容詞は格要素の係り先になりにくいために、係り先の候補から除外される。

- (5) 長女が気づき、家族とともに二人を助けようとしたが火の回りが早く救い出せなかった。

この例では、「回りが」は形容詞「早く」に係るのが正解であるが、「早く」は係り先の候補から除外されており解析が誤っている。

また、3の処理の例を次に示す。

- (6) 商工会議所の会頭が、質問に先頭を切って答えた。

KNPは、「質問に」の係り先の候補として、「切って」、「答えた」の2つの可能性を考慮する。この場合、“より近くに係る”という優先規則に従って係り先は「切って」に決定されるが、この解析は誤りである。この例のように、係り先の候補が複数存在すると、係り先に曖昧性があり確信度が低いので、このような用例は収集しない。

4 用例格フレームの作成

2章の例文で示したように、用言の意味の異なる用例をひとつの格フレームとしてまとめて学習すると、誤った表現を許す格フレームを作ってしまう。従って、格パターンの異なる格フレームは別々に学習する必要がある。

用言の意味を決定する重要な格要素は用言の直前にくることが多い。また、用言とその直前の格要素をペアにして考えると、用言の意味の曖昧性はほとんどなくなる。そこで、用例を、**用言とその直前の格要素の組を単位としてまとめる**という処理を行い、用例パターン(図1)を作る。用例パターンの用言の直前の格要素を**用例パターンの直前格要素**と呼ぶ。

用例パターンは、ひとつの用言について、直前格要素の数だけ存在している。そのため、次の例のように、意味がほとんど同じパターンまで個別に扱われている。

- (7) a. 自宅:で 雑誌:を 読む

b. 先生:が {図書館, 家}:で 本:を 読む

そこで、ほとんど意味や格パターンが同じ用例パターンをマージするために、用例パターンのクラス

タリングを行う。以下では、このクラスタリングの詳細について述べる。

4.1 用例パターン間の類似度

用例パターンのクラスタリングは、格用例群間の類似度と格の一致度をもとにした用例パターン間の類似度を用いて行う。

まず、日本語語彙大系のシソーラスを利用し、意味属性 x, y 間の類似度 $sim(x, y)$ を次のように定義する。

$$sim(x, y) = \frac{2L}{l_x + l_y}$$

ここで、 l_x, l_y は x と y の意味属性のシソーラスの根からの階層の深さを表し、 L は x と y の意味属性で一致している階層の深さを表す。類似度 $sim(x, y)$ は 0 から 1 の値をとる。

次に、単語 e_1, e_2 間の類似度 $sim_e(e_1, e_2)$ は、 e_1, e_2 それぞれの意味属性間の類似度の最大値と考え、

$$sim_e(e_1, e_2) = \max_{x \in s_1, y \in s_2} sim(x, y)$$

と定義する。 s_1, s_2 はそれぞれ e_1, e_2 の日本語語彙大系における意味属性の集合である（日本語語彙大系では、単語に複数の意味属性が与えられている場合が多い）。

また、格用例群 E_1, E_2 間の類似度 sim_E は、格用例の類似度の和を正規化したもので、

$$sim_E(E_1, E_2) = \frac{\sum_{e_1 \in E_1} \sum_{e_2 \in E_2} \sqrt{|e_1||e_2|} sim_e(e_1, e_2)}{\sum_{e_1 \in E_1} \sum_{e_2 \in E_2} \sqrt{|e_1||e_2|}}$$

とする。 $|e_1|$ などの絶対値は頻度を表している。ただし、実際には、 $sim_e(e_1, e_2)$ の上位から 1/5-best だけを足すことにする。

ここで、用例パターン F_1, F_2 の格の一致度 cs は、格用例の頻度を重みとした、共通格の割合の平方根とし、

$$cs = \sqrt{\frac{\sum_{i=1}^n |E_{1cc_i}| + \sum_{i=1}^n |E_{2cc_i}|}{\sum_{i=1}^l |E_{1c1_i}| + \sum_{i=1}^m |E_{2c2_i}|}}$$

と定義する。ただし、用例パターン F_1 中の格を $c1_1, c1_2, \dots, c1_l$ 、用例パターン F_2 中の格を $c2_1, c2_2, \dots, c2_m$ 、 F_1, F_2 の共通格を cc_1, cc_2, \dots, cc_n とする。また、 E_{1cc_i} は F_1 内の格 cc_i に含まれる格用例群で、他も同様である。

用例パターン F_1, F_2 間の類似度 $score$ は、格の一致度 cs と F_1, F_2 の共通格の格用例群間の類似度の積とし、

$$score = cs \cdot \frac{\sum_{i=1}^n \sqrt{w_i} sim_E(E_{1cc_i}, E_{2cc_i})}{\sum_{i=1}^n \sqrt{w_i}}$$

と定義する。ただし、 w_i は、格用例群間の類似度に対する重みであり、格用例の頻度の平方根をもとにして、

$$w_i = \sum_{e_1 \in E_{1cc_i}} \sum_{e_2 \in E_{2cc_i}} \sqrt{|e_1||e_2|}$$

とする。

4.2 用例パターンの意味属性の固定

用例パターン間の類似度は、用例パターンの直前格要素の意味属性が大きく影響する。すると、用例パターンの直前格要素に多義性があるときに問題がある。例えば、「合わせる」の用例パターンのクラスタリングにおいて、用例パターンの組 (手, 顔)¹が意味属性 <動物 (部分)>、(手, 焦点) が意味属性 <論理・意味等> でマージされるときに、<動物 (部分)> と <論理・意味等> はまったく類似していない意味属性であるのにもかかわらず、(手, 顔, 焦点) というマージが起きてしまう。

この問題に対処するために、もっとも類似度が高い用例パターンの組から意味属性を固定する処理、すなわち用例パターンの意味の曖昧性解消を行う。この処理は、用例パターンの直前格要素の意味属性を固定することによって、次のような手順で行う。

1. 類似度が高い用例パターンの組 (p, q) から順に、両方の用例パターンの直前格要素 n_p, n_q の意味属性を固定する。固定する意味属性は、 n_p, n_q 間の類似度を最大にする意味属性 s_p, s_q とする。
2. p, q に関する用例パターンの類似度を再計算する。
3. 閾値 *threshold* を越える用例パターンの組がなくなるまで、この 2 つの処理を繰り返す。

4.3 アルゴリズム

用例パターンのクラスタリングの手順を以下に示す。

¹ここでは、直前格要素で用例パターンを表している。

1. まず、直前の格要素の出現頻度がある閾値以上あるという条件で足切りを行う。これは、直前の格以外にも格用例がある程度の回数以上出現しているような安定した用例パターンだけを対象にするためである。この閾値は 10 に設定した。
2. あらゆる 2 つ組の用例パターンの類似度を計算し、用例パターンの意味属性を固定する。これらの処理は、4.2 節で述べたように繰り返す。
3. 用例パターン間の類似度が閾値 $threshold$ を越える組について、用例パターンのマージを行う。
4. 頻度の閾値を越えない用例パターン (残りの用例パターン) を作成された用例格フレームに振りわけする。用例パターンと用例格フレーム間の類似度を計算し、類似度が閾値 $threshold_r$ を越え、もっとも類似している用例格フレームにマージする。 $threshold_r$ は副作用を生まないように、ある程度高い値に設定する。

5 必須格の選択

クラスタリングを行った結果得られる用例格フレームについて、格用例の頻度が少ない格は除く。これは、ひとつには構文解析結果の誤りへの対策であり、また頻度の少ない格はその用言と関係が希薄であると考えられるからである。ただし、ガ格についてはすべての用言がとると考え、頻度が少なくても削除せず、逆にガ格の格用例がない場合には、意味属性 <主体> を補うことにした。

頻度の閾値は、現在のところ経験的に $2\sqrt{mf}$ と定めている。ただし、 mf はその用言において最も多く出現した格の延べ格用例数である。例えば、 mf が 100 のとき、閾値は 20 となり、格用例の頻度が 20 未満の格は捨てられることになる。

6 作成した格フレーム辞書

毎日新聞約 7 年分の 360 万文から実際に格フレーム辞書を構築した。クラスタリングの閾値 $threshold$ は 0.65、残りの用例パターンを振り分ける閾値 $threshold_r$ は 0.80 に設定した。これは、格パターンが違ったり、意味が違う格フレームが同じ格フレームにならないという基準で設定したものである。従って、格フレームは基本的にはばらばらで、

表 1: 構築した格フレームの例 (* はその格が用言の直前の格であることを示す。)

用言	格	用例
買う 1	ガ格 ヲ格* デ格	【主体: <数量>人, 乗客】 株, 円, 土地, もの, ドル, 切符 【場所: 店, 駅】, <数量>円
買う 2	ガ格 ヲ格*	対応, 厚生, 絵はがき, 蓄財 怒り, ひんしゅく, 失笑, 反感
:	:	:
読む 1	ガ格 ヲ格*	【主体: 大学生, 首相, 先生】 本, 記事, 新聞, 小説, 投書
読む 2	ガ格 ヲ格 デ格*	【主体: <主体>】 話, <補文>, 意見, 惨状 新聞, 本, 本紙, 教科書
読む 3	ガ格 ヲ格*	【主体: <主体>】 先
:	:	:
ただす 1	ガ格 ヲ格* について	【主体: 氏, 委員, 議員】, 喚問 見解, 真意, 考え, 方針, 問題 問題, <補文>, 展開, 責任
ただす 2	ガ格 ヲ格*	【主体: 委員長, 自ら, 業界】 【主体: 身】, 姿勢, 姿, 威儀
:	:	:
人気 1	ガ格 ニ格*	グッズ, 自転車, ベルト 【主体: 女性, 主婦, 男性】
人気 2	ガ格*	もの, サービス, 商品, 海産物
人気 3	ガ格*	<補文>

意味がほとんど同じ格フレームを最小限まとめたものになっている。格フレームの例を表 1 に示す。この表では、<主体>、<場所> の意味属性をもつ格用例を【主体】、【場所】という意味属性でまとめて表示している。

52,000 個の用言について格フレームが構築され、用言あたりの平均格フレーム数は 1.3 個、格フレームあたりの格の平均数は 1.9 個、格あたりの平均異なり格用例数は 6.0 個であった。また、クラスタリングによって用例格フレーム数は用例パターン数の 38% になった。

構築した格フレーム辞書をみると、「人気」、「賛成」といった名詞+判定詞の格フレームも学習できている。また、「ただす」の「について」のように、複合辞の格についても学習できている。

7 解析実験

得られた格フレーム辞書の静的な評価は難しいので、それを用いた格解析を通して評価する。毎日新聞の記事 100 文をテストセットとし²、これに対して格解析を行った。格解析は [2] の方法を用いた。格解析結果の評価は、提題と被連体修飾詞の格を正しく認識できるかどうかで行う。格解析の評価を表 2、解析結果の例を次に示す³。

- (1) ¹大蔵省は○ガ格 九日、信託銀行の不良債権の処理を促進するため、一九九五年三月期決算で信託銀行各行が ²積み立てている
²特別留保金の○ヲ格 取り崩しを ³認める
³方針を×ニ格⇒ 外の関係 ¹決めた。
- (2) 金権選挙追放策の一つとして、戦後
¹廃止されてしまった 民衆訴訟による
¹当選無効制度の○ガ格 ²復活も ○ヲ格
²試みる……。
- (3) これらの ¹業界は×ヲ格⇒ ガガ、比較的外圧を受けにくく、また政治的発言力が強い、という特徴が ¹ある。
- (4) 代表質問を“影の内閣”として ¹設置した
¹政権準備委員会の×ニ格⇒ ガガ 「施政方針演説」と位置付け、政権担当能力をアピールするのが狙い。

表 2 において、格解析の精度をみるために係り受けの誤りを除いて考えると、提題が 89%、被連体修飾詞が 73% というかなりよい精度で格の認識ができてることがわかる。誤りの大きな原因は、「～を与える役割」のような外の関係、「業界は～という特徴がある」といったガガ構文である。この問題の対処は今後の課題である。

8 おわりに

本論文では、用言とその直前の格要素の組を単位として、生コーパスから用例を収集し、それらのクラスタリングを行うことによって、格フレーム辞書を自動的に構築する手法を提案した。得られた辞書

²このテストセットは、格フレーム辞書の構築には用いていない。

³○の下線部は格解析が正しく、×の下線部は誤っている。下線部の後に、格解析によって認識された格を記述し、格解析が誤っているときは、⇒ の後に正解の格を記述した。

表 2: 提題、被連体修飾詞の格解析の評価

	正解	誤り			
		対応付けの誤り	外の関係による誤り	ガガ構文による誤り	係り受けの誤り
提題	80	6	—	3	16
連体修飾	47	3	14	—	8

を用いて実際に格解析を行った結果、提題、連体修飾の格の解釈をかなり高い精度で行うことができた。従って、「使える」レベルの格フレーム辞書を構築できたと考えられる。今後、この格フレーム辞書を用いて文脈解析を行う予定である。

参考文献

- [1] Michael John Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of ACL*, pp. 184–191, 1996.
- [2] S. Kurohashi and M. Nagao. A method of case structure analysis for japanese sentences based on examples in case frame dictionary. In *IEICE Transactions on Information and Systems*, Vol. E77-D No.2, 1994.
- [3] NTT コミュニケーション科学研究所. 日本語語彙大系. 岩波書店, 1997.
- [4] 宇津呂武仁, 宮田高志, 松本裕治. 最大エントロピー法による下位範疇化の確率モデル学習および統語的曖昧性解消による評価. 情報処理学会 自然言語処理研究会 97-NL-119, pp. 69–76, 1997.
- [5] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 自然言語処理, Vol. 1, No. 1, 1994.
- [6] 情報処理振興事業協会技術センター. 計算機用日本語基本動詞辞書 IPAL (Basic Verbs) 説明書. 1987.