日本語文生成器Haoriにおける複文合成

緒方健人　　佐藤理史　　松崎拓也
名古屋大学大学院工学研究科

1　はじめに

文解析に関する研究に比べ、文生成に関する研究は極端に少ない。我々は、昨年度より、短編小説の自動生成のために必要となる、日本語文生成器を実現することに取り組んでいる[1, 2]。

一般に、文章の生成において、その材料となる部品が小さい方が、少数の部品で多種の文章を生成できる。短編小説の自動生成において、我々が想定している部品のサイズはおおよそ単文であるが、単文を羅列しただけでは、まともな文章になり得ない場合がほとんどである。このため、複数の単文を組み合わせて複文を合成する仕組みが必要となる。

このような背景から、我々は、複数の部品から複文を合成する機構を実装し、文生成器Haoriに組み込んだ。本稿では、その内容について報告する。

2　複文の合成

複文とは、複数の述語を含む文である。複文において、述語を中心としたまとまりを**節**という。複文は、文全体の中心となる**主節**と、それと特定の関係で結びつく**従属節**あるいは**並列節**(これらを合わせて、以下では単に従属節と呼ぶ)から構成される[3]。

ここでは、以下の複文を生成することを考える。

(1) 太郎は、風邪ぎみだったので、大学を休んだ。

この文の主節は「太郎は大学を休んだ」であり、従属節は「風邪ぎみだったので」である。

複文も文節係り受け構造によって表現できるので、Haoriは、以下の文節係り受け構造(JBT)から、この文を生成することができる。(Haoriはプログラミング言語Rubyにより実装されている。以下では、Rubyの表記をそのまま用いる。)

(2)
```
[{mp: :empty},
 [{mp: '太郎', fp: 'ハ副'}],
 [{mp: '風邪ぎみ/ダ', cform: 'タ形',
   fp: 'ノデ接'}],
 [{mp: '休む', cform: 'タ形', punc: '。'},
  [{mp: '学校', fp: 'ヲ格'}]]]
```

単に、この文だけを生成するのであれば、特に不自由はない。しかしながら、小説生成においては、話の流れによっては「学校を休んだ理由」を変えたい場合がある。これを実現するために、まったく別なJBTを用意するのは非効率である。

例文(1)は、以下に示す2文の情報を1文にまとめた文である。

(3) 太郎は学校を休んだ。
(4) [太郎は] 風邪ぎみだった。

つまり、例文(3)はそのまま流用し、例文(4)に相当する部分のみ、他の文で置き換えることができれば、JBTの記述量を削減できる。

一般に、複数の文を一つの文にまとめる処理は、多くの複雑な操作を含む。例文(4)の主語が例文(3)の主語と一致するため、例文(1)の従属節では「太郎が」が省略される。さらに、従属節の従属度の違いにより、従属節がテンス、アスペクト、モダリティ等を持つかどうかが定まると言われている。将来的には、このような複雑な現象を考慮すべきと考えるが、短期的には、これらを無視した単純な複文合成を実現することに集中すべきと考え、以下のような方針を立てた。

1. 従属節への変形
   (a) 従属節の元となる(都合のよい)JBT Cと、生成すべき従属節の仕様が与えらえる。
   (b) 従属節の仕様に合わせて、Cのルート文節を変形する。得られたJBTをC'とする。
2. 従属節の主節への挿入
   (a) 主節となるJBT Sと、(変形された)従属節C'、および、従属節の挿入位置が与えられる。
   (b) 挿入位置の指定に従って、SにC'を挿入し、複文S'を得る。

これらを具体化したのが以下に示すコードである。この後、S2を表層文字列化することにより、文字列として複文を出力する。

(5)
```
S = [{mp: :empty},
     [{mp: '太郎', f: 'ハ副'}],
     [{mp: '休む', cform: 'タ形'},
      [{mp: '学校', f: 'ヲ格'}]]]
C = [{mp: '風邪ぎみ/ダ', cform: 'タ形'}]
spec = {type: '副詞節/原因・理由節/ノデ'}
p = {node: [0], pos: 1}
S2 = insert(S, transform(C, spec), p)
```

続く二つの節では、従属節への変形と、従属節の主節への挿入について述べる。

3 従属節への変形

従属節の生成を実現するためには、まず、生成すべき従属節の範囲と形式を定める必要がある。我々の知る限り、日本語の節の網羅的リストは存在しない。このため、益岡・田窪文法[3]、日本語節境界検出プログラムCBAP[4]、および、Rainbow[5]を参考に、節形式のリストを作成した。

表1と表2にHaoriで生成可能な従属節の一覧を示す。この表の各欄は以下のとおりである。

名称 従属節の名称。階層的な体系を採用している。(中間段階では、それぞれデフォールトが規定されている。)

述語形式 従属節の中核たる述語の形式。典型的には、述語の活用形である。特別な形式として、「引用句」や「文末」などがある。

構造 述語形式に後続する助詞や節末機能語[1]を表す。[ ]は挿入可能であることを示す。{ }はその中の要素の順番が任意であることを示す。助詞は「タリ並列」のように、リテラルの後に助詞の種別(格、副、接続、終、並立、引用)を明記している。「ニ連用」は、判定詞由来の助詞「に」を表す。助詞以外の語は、カタカナ以外で表記している。

従属節への変形は、原則として、JBTのルート文節(BALと呼ばれるデータ構造で表現されている)を、指定された節の形式に変形することで実現する。BALの主要な属性は、

1. :mp　文節の主要部(おおよそ、内容語)。
2. :lbal　文節の中に埋め込まれる「軽い文節」。形式名詞や節末機能語などは、ここに埋め込まれる。
3. :cform　主要部の活用形。
4. :fp　文節の機能部(おおよそ、助詞の列)。

の４つであるので、これらの値をどのように書き換えるかを規定すれば、文節の変形操作を規定できる。

[1]節末機能文節[6]の主要部となる語。

表 1: 従属節の節末形式(1)

名称	述語形式	後続
並列節/順接		
/総記/連用	基本連用形	
/総記/連用	基本連用形	
/総記/連用ニ	連用ニ	ニ連用
/総記/テ	テ形	
/総記/テ２	テ形２	
/例示/タリ	連用形	タリ並列
/例示/ヤラ	基本形	ヤラ並列
/例示/ダノ	基本形	ダノ並列
/例示/トカ	基本形	トカ並列
/例示/ナリ	基本形	ナリ並列
/累加/シ	(基本形|タ形)	シ接
/累加/条件	基本条件形	
/ガ	(基本形|タ形)	ガ接
並列節/逆説		
/ガ	(基本形|タ形)	ガ接
/ケレド	(基本形|タ形)	ケレド接
/ケレドモ	(基本形|タ形)	ケレドモ接
引用節/直接		
/ト	引用句	[ナド副] ト引用 [副]
/ッテ	引用句	[ナド副] ッテ引用 [副]
/トカ	引用句	トカ並列
/ナンテ	引用句	ナンテ副
/ト	文末	[ナド副] ト引用 [副]
引用節/間接		
/ッテ	文末	[ナド副] ッテ引用 [副]
/トカ	文末	トカ並列
/ナンテ	文末	ナンテ副
/ヨウ	基本形	よう
/ヨウニ	基本形	よう ニ連用 [副]
連体節	(基本形|タ形)	
引用連体節		
/トイウ	文末	[ナド副] ト引用 いう
/トイッタ	文末	[ナド副] ト引用 いった
/ッテイウ	文末	[ナド副] ッテ引用 いう
/ッテイッタ	文末	[ナド副] ッテ引用 いった
/トカイウ	文末	トカ並列 いう
/トカイッタ	文末	トカ並列 いった
/ナンテイウ	文末	ナンテ副 いう
/ナンテイッタ	文末	ナンテ副 いった
/トノ	文末	ト引用 ノ連体
/ノ	(疑問文末|副詞節)	ノ連体
/ヨウナ	基本形	ような
/ミタイナ	基本形	みたいな
/ベキ	基本形	べき
補足節	(基本形|タ形)	(こと|の|ところ) {格,[副],[並列]}
疑問補足節	疑問文末	{[格],[副],[並列]}
副詞節/時間		
/トキ	(基本形|タ形)	(とき|時) [ニ連用] [副]
/オリ	(基本形|タ形)	(おり|折) [ニ連用] [副]
/サイ	(基本形|タ形)	(さい|際) [ニ連用] [副]
/サイチュウ	(基本形|タ形)	最中 [ニ連用] [副]
/タビ	(基本形|タ形)	(たび|度) [ニ連用] [副]
/トタン	(基本形|タ形)	途端 [ニ連用] [副]
/ヤイナヤ	基本形	ヤイナヤ接
/ナリ	基本形	ナリ接
/シュンカン	(基本形|タ形)	瞬間 [ニ連用] [副]
/マエ	基本形	前 [ニ連用] [副]
/イゼン	基本形	以前 [ニ連用]
/タアト	タ形	(あと|後) [(ニ連用|デ連用)]
/タノチ	タ形	(のち|後) [ニ連用]
/テカラ	テ形	カラ接
/テイコウ	テ形	以降
/テイライ	テ形	以来
/ウチ	基本形	(うち|内) [ニ連用]
/アイダ	基本形	(あいだ|間) [ニ連用]
/マニ	基本形	(ま|間) ニ
/マデ	基本形	まで [ニ連用]

たとえば、先に示したコード(5)のspecは、「副詞節/原因・理由節/ノデ」という節に変形せよという指定(仕様)である。表1より、この節の述語形式は「(基本形|タ形)」、後続は「ノデ接」とわかる。この情報は、内部的には、以下のような変形規則に変換される。

表 2: 従属節の節末形式(2)

名称	述語形式	後続
副詞節/原因・理由		
/ノデ	(基本形|タ形)	ノデ接
/タメ	(基本形|タ形)	ため [ニ連用]
/ケッカ	(基本形|タ形)	結果
/ダケニ	(基本形|タ形)	ダケ副 [ニ連用]
/アマリ	(基本形|タ形)	(あまり|余り) [ニ連用]
/セイデ	(基本形|タ形)	せい デ連用
/バカリニ	(基本形|タ形)	バカリ副 ニ連用
/オカゲデ	(基本形|タ形)	(おかげ|お陰) デ連用
/テ	テ形	
/カラ	(基本形|タ形)	カラ接理由
/ノダカラ	(基本形|タ形)	のだ カラ接理由
/モノデ	(基本形|タ形)	もの デ連用
/モノダカラ	(基本形|タ形)	ものだ カラ接理由
副詞節/条件		
/バ	基本条件形	
/タラ	タラ形	
/ト	基本形	ト接
/タトコロ	タ形	トコロ接
/ナラ	(基本形|タ形)	ナラ接
/ナラバ	(基本形|タ形)	ナラバ接
/トスレバ	(基本形|タ形)	とすれば
/トシタラ	(基本形|タ形)	としたら
/トスルト	(基本形|タ形)	とする ト接
/バアイ	(基本形|タ形)	場合 [ニ連用] [副]
副詞節/譲歩		
/テモ	テ形	モ副
/タッテ	タ形	ッテ副
/トシテモ	(基本形|タ形)	として モ副
/タトコロデ	タ形	トコロデ接
副詞節/付帯状況		
/タママ	タ形	まま [デ連用]
/タキリ	タ形	キリ接
/テ	テ形	
/ナガラ	基本連用形	ナガラ接
/ツツ	基本連用形	ツツ接
/ツイデニ	(基本形|タ形)	ついで ニ連用
副詞節/様態		
/ヨウニ	(基本形|タ形)	よう ニ連用
/ゴトク	(基本形|タ形)	(ごとく|如く)
/トオリ	(基本形|タ形)	(とおり|通り) [ニ連用] [副]
/ミタイニ	(基本形|タ形)	みたい ニ連用 [副]
副詞節/逆説		
/ケレドモ	(基本形|タ形)	ケレドモ接
/ノニ	(基本形|タ形)	ノニ接
/ニモカカワラズ	(基本形|タ形)	ニモカカワラズ接
/ノニモカカワラズ	(基本形|タ形)	ノニモカカワラズ接
/ナガラ	基本連用形	ナガラ接
/ツツ	基本連用形	ツツ接
/ワリニ	(基本形|タ形)	わり ニ連用 [副]
/クセニ	(基本形|タ形)	(くせ|癖) ニ連用 [副]
/モノノ	(基本形|タ形)	モノノ接
副詞節/目的		
/タメニ	基本形	ため ニ連用 [副]
/ノニ	基本形	の ニ連用 [副]
/ベク	基本形	べく
/ヨウニ	基本形	(よう|様) ニ連用 [副]
/ニ	基本連用形	ニ連用 [副]
副詞節/程度		
/クライ	(基本形|タ形)	くらい [ニ連用] [副]
/ホド	(基本形|タ形)	ほど [ニ連用] [副]
/ダケ	(基本形|タ形)	だけ
副詞節/帰結		
/イジョウ	(基本形|タ形)	以上 [副]
/カラニハ	(基本形|タ形)	から ニ連用 ハ副
/カギリ	(基本形|タ形)	(かぎり|限り) [デ連用 ハ副]
副詞節		
/前提/ウエデ	(基本形|タ形)	(うえ|上) デ連用
/比較/ヨリ	基本形	ヨリ格
/比較/イジョウニ	(基本形|タ形)	以上 ニ連用
/対比/イッポウ	基本形	一方 ニ連用
/対比/ハンメン	基本形	反面
/相関/ニツレテ	基本形	ニツレテ格
/相関/ニシタガッテ	基本形	ニシタガッテ格
/予想外/ドコロカ	基本形	ドコロカ接
/他/ホカ	(基本形|タ形)	(ほか|他)

(6) `{cform: ['基本形', '夕形'], fp: 'ノデ接'}`

この変形規則は、対象となるルート文節のBALをどのように書き換えるかを規定している。値がリストの場合は、属性値がそのリストのメンバーであることを要請すること(そうでない場合は、値を第1要素に変更する)、それ以外の値は、属性値をその値に変更することを意味する。すなわち、上記の規則は、以下のことを意味する。

1. 属性:cformの値が「基本形」または「夕形」ではない場合は、その値を「基本形」に変更せよ
2. 属性:fpの値を「ノデ接」(接続助詞「ので」)に変更せよ、

なお、明示的に指定のない属性は、元のBALの値を引き継ぐ。この規則をコード(5)のCに適用することにより、以下のJBTが得られる。

(7) `[{mp: '風邪ぎみ/ダ', cform: '夕形', fp: 'ノデ接'}]`
(=風邪ぎみだったので)

述語形式が「文末・疑問文節・副詞節末・引用句」の場合は、もう少し複雑である。たとえば、「引用連体節/トイウ」に変形する変換規則は、以下の規則となる。

(8) `{dummy: {fp: 'ト引用/いう'}}`

この規則は、元となるルート文節は変形せず、その親文節としてダミー文節(文節の主要部が空の文節)を作成し、その文節の機能部を「ト引用/いう」に設定する。

このようなダミー文節を導入するのは、文末は終助詞を持つ可能性があるからである。Haoriの文法では、終助詞は機能部に属するため、機能部を直接書き換えてしまうと、終助詞が脱落する。これを避けるための措置である。

表1と表2に示した節には、[ ]で示す挿入可能な要素を持つものがある。たとえば、「書いたときにも」のように、「副詞節/時間/トキ」に「に」と「も」を挿入したい場合は、節の仕様を次のように指定する。

(9) `{type: '副詞節/時間/トキ', attach: 'ニ連用', adv: 'モ副'}`

この指定は、以下の変形規則に変換され、「に」と「も」が挿入されたトキ節が生成される。

(10) `{lbal: 'とき', fp: 'ニ連用/モ副'}`

4 従属節の主節への挿入

従属節の主節への挿入は、原則として、主節のどの位置に挿入するかを規定すれば実現できる。主節のデー

タ構造である文節係り受け構造(JBT)は木構造であるため、挿入位置は、(a)どの節点の、(b) 何番目の子節点として、という2つの情報によって表現できる。(a)は、ルート文節からの経路(パス)として表現する。ルート文節を`[0]`、その一番最初の子節点を`[0,0]`、二番目の子節点を`[0,1]`のように表す。(b)も同様に、一番目の子節点として挿入する場合は`0`(または`:fisrt`)、二番目の子節点として挿入する場合は`1`のように指定する。なお、一番最後の子節点として挿入する場合は、`:last`を指定する。

たとえば、コード(5)の`p`は、「ルート文節(`node: [0]`)の2番目の子節点(`pos: 1`)として、従属節を挿入せよ」ということを意味する。この指定により、「太郎は」の後に、従属節「風邪ぎみなので」が挿入される。

並列節を持つ文の文節係り受け構造をどのように表現すべきかは、自明ではない。たとえば、次の例文を取り上げよう。

(11) 太郎が文面を考えて、花子が清書した。

我々は、ルートにダミー文節を持つ、以下のような構造を採用する。なぜならば、「考える」と「清書する」の間には、係り受け関係は存在しないと考えるからである。

(12)
```
[[{mp: :empty}],
 [{mp: '考える', cform: 'テ形'},
  [{mp: '太郎', fp: 'ガ格'}],
  [{mp: '文面', fp: 'ヲ格'}]],
 [{mp: '清書する', cform: 'タ形'},
  [{mp: '花子', fp: 'ガ格'}]]]
```

このような構造を作り出すために、従属節の挿入時にダミー文節を作成(挿入) することを可能とした。たとえば、以下の指定は、ルート文節(`node: [0]`)の親節点としてダミー文節を作成し(`dummy: true`)、その最初の子節点(`pos: :first`)として従属節を挿入することを意味する。

(13)
```
{node: [0], pos: :first, dummy: true}
```

5 現状と課題

現在までにプログラムの実装はほぼ完了しており、テストのフェーズに移行しつつある。一般に、生成システムのテストおよび評価をどのような形で実施すべきかは、自明ではない。それ自身が研究項目である。我々が考えている一つのプランは、益岡・田窪文法[3]に記述されている例文を用いて、システムが正しく節を生成できるかどうかをチェックし、かつ、生成できる節の網羅性(カバレジ)を調査するというものである。生成システムは、第一に、仕様通り動作するかどうかが問題であり、それを確認するためのテストスイート(test suite)が必要であろう。まずは、文法書の例文から着手し、その後、たとえば、現代日本語書き言葉均衡コーパス(BCCWJ)のようなコーパスから例文を採取するというのが、一つの方向だと考える。

本研究の焦点は、主に、従属節の生成にあるが、一方で、BCCWJに対する節境界付与の研究も進行中である[6, 7]。先に述べたように、日本語の節の全貌は判明しているわけではなく、どの単位を節と認定するか、それを何節と呼ぶかは、いまだ標準化されていない。これまでは、節の生成システムと認定システムの研究を独立に進めてきたが、適当な時期で節の名称の標準化(統一化)を行なう必要があろう。

生成システムでは、節を(形式ではなく)意味的に分類することへの要請度が高い。というのは、最終的には、「文Aに、文Bを『理由』という関係でつなげ」というような合成を行ないたいからである。現在のHaoriの節の名称の体系は、このような方向を志向している。実際、階層的な体系にデフォールトを設定しているので、たとえば、節の仕様として「副詞節/原因・理由」だけを指定することも可能である[2]。

短期的にも、まだ多くの課題が残っているが、より複雑な現象に対しても、少しずつ検討を始める必要がある。小説生成において優先度が高いのは、共通する格要素の省略と、文の連体節への変形(「彼が小説を書いた」→ 「彼が書いた小説」)である。

謝辞 本研究は、JSPS 科学研究費基盤研究 (B) 「文章の読解と産出のための言語処理技術」(課題番号 15H02748) の助成を受けている。

参考文献

[1] 佐藤理史. 「文生成器を作る」とはどういうことか. 言語処理学会第21回年次大会発表論文集, pp. 1080–1083, 2015.
[2] 緒方健人, 佐藤理史, 松崎拓也. 文節木の段階的実体化による日本語文生成器の作成. 2015年度人工知能学会全国大会論文集, 2015.
[3] 益岡隆志, 田窪行則. 基礎日本語文法—改訂版—. くろしお出版, 1992.
[4] 丸山岳彦, 柏岡秀紀, 熊野正, 田中英輝. 日本語節境界検出プログラムcbapの開発と評価. 自然言語処理, Vol. 11, No. 3, pp. 39–68, 2004.
[5] 加納隼人, 佐藤理史. 日本語節境界検出プログラムRainbowの作成と評価. 第13回情報科学技術フォーラム(FIT2014), E-005, 第2分冊, pp. 215–216, 2014.
[6] 佐藤理史, 丸山岳彦. 節境界認定に関する諸問題. 第8回コーパス日本語学ワークショップ予稿集, pp. 225–232, 2015.
[7] 佐藤理史, 丸山岳彦, 夏目和子. 現代日本語書き言葉均衡コーパスに対する節境界付与. 言語処理学会第22回年次大会発表論文集, 2016.

[2] この場合、デフォールトとして定義されている「副詞節/原因・理由/ノデ」が指定されたものと解釈される。