

オブジェクト指向的アプローチを用いた係り受け/語義曖昧性解決 /照応解析/述語項解析システム

伊藤英紀^{†1}

概要: 係り受け解析、語義曖昧性解決、照応解析、および述語項解析の4つのタスクに対する新たなアプローチを提案する。第一に、これら4つの処理を、逐次的（パイプライン的）にでなく同時に行う。これにより、逐次的に行うよりも解析精度が向上することが期待される。第二に、文中の名詞をオブジェクトに対応させ、それらオブジェクトの属性とその変化をトラッキングする。形容詞は一般にオブジェクトの属性を、動詞はそれらオブジェクト属性の変化を表す、と考える。これにより、文脈の情報を解析結果に反映させることが可能となる。このような意味理解のモデルは、人間が脳で行っているであろう理解プロセスに近く、自然言語理解には有望なアプローチである。このアイデアに基づいたシステムを実装し、例文に対して動作を確認した。

キーワード: 係り受け解析、語義曖昧性解決、照応解析、述語項解析、オブジェクト指向

An Object-Oriented System Conducting Dependency Analysis, Word Sense Disambiguation, Anaphora Resolution and Predicate Argument Analysis

EIKI ITO^{†1}

Abstract: A novel approach to dependency analysis, word sense disambiguation, anaphora resolution, and predicate argument analysis is proposed. First, these four tasks are resolved jointly, as opposed to applying those processes in a pipelined (sequential) manner. This is expected to improve the system's accuracy. Secondly, the nouns in the input sentences are associated with corresponding objects. Attributes and their changes of the objects are identified during the analysis. Typically, adjectives are considered as representing the object's attributes, and verbs representing changes in those attributes. Such semantic model naturally reflects the language understanding process by humans. A system using this approach has been implemented, which successfully analyzes sample sentences.

Keywords: Dependency Analysis, Word Sense Disambiguation, Anaphora Resolution, Predicate Argument Analysis, Object-Oriented

1. はじめに

計算機による自然言語処理においては、平文が入力されると通常まず形態素解析が行われる。日本語の場合はこの後に文節分割処理が入る。続いて、単語間（日本語の場合は通常、文節間）の依存構造を決定する依存構造解析（係り受け解析）が行われる[5]。係り受けの後の処理については、まだ技術が確立していない面もあるが、単語の持つ複数の語義のうち適切なものを選ぶ語義曖昧性解決、同じ対象を指す単語どうしを対応づける照応解析、および述語の格補語や修飾語を同定する述語項解析が必要になると一般に考えられていて、多くの研究がなされ、またタスクコンペティションも行われている[7,10,11]。

各ステージの技術の達成度を見ると、形態素解析と文節

分割は、完璧ではないまでもほぼ実用的と言える高い精度を達成している。係り受けも研究が進んでおり、精度もかなり向上しているが、形態素解析と比較するとまだ不十分な面があり、一層の精度向上が望まれている。語義曖昧性解決、照応解析、および述語項解析はいずれもまだ発展途上の段階であり、高い精度は得られておらず、また手法も確立されているとは言えない[4,7,10,11,12]。

語義曖昧性解決、照応解析、および述語項解析の従来研究でなぜ精度が上がらないのかを考察してみる。まず考えられるのは、判断に使用している情報（素性）が適切・十分なのだろうか？という疑問である。たとえば「彼を逮捕するため…」という文において「彼」の照応先を特定したいとする。候補として「太郎」と「次郎」が挙げたとして、このどちらかを選ぶとしよう。人間が文を理解するときは、太郎や次郎が逮捕されるような行為（犯罪等）を犯したか、を必ず考えているはずである。しかし従来の研究においては「太郎」や「次郎」という単語のみを対象としており、

^{†1} (株)富士通研究所
Fujitsu Laboratories Ltd.

「太郎」という実体がどういう人か、何をしたかをまったく考慮しない。これで正しい判断ができるのか、という疑問は当然出てくる。オブジェクトの実体を考慮することは、人間が頭で行っているプロセスに近いはずであり、その意味で自然かつ有望なアプローチであると考えられる。

もうひとつは、逐次処理の限界である。従来の述語項解析研究においては、まず係り受け処理を行い、その出力を述語項解析の入力とするものが多かった。しかし、実際には述語項解析の結果が係り受けの判定に影響を及ぼす場合もある。たとえば、「割り箸を食べた弁当の箱にしまう」という文を考える。述語項を考えずに係り受けだけを行うシステムだと、「割り箸を」が「食べた」に係る、と判断しがちだ（実際、cabocha^a v0.64 ではこうなる）。こう判断してしまうと、「食べた」の目的(ヲ)格は「割り箸」と決まってしまう。しかし、後段の述語項解析において、「弁当」が「食べた」の何格かを考えると、選好性から考えて当然目的格だろうと思われるのだが、これは係り受けの結果と矛盾する。「矛盾するから、そもそも『割り箸を』が『食べた』に係るのが間違い、『しまう』に係るべきだ」というのが正しいのだが、逐次処理であるかぎりこれは不可能である。つまり、逐次処理自体を見直すことが必要である。

同じことが、係り受けと述語項だけでなく、語義曖昧性解決や照応解析に関しても成り立つ。すなわち、語義や照応詞の選択によって最適な係り受け方が変わるケースはよく見られる。たとえば「プログラムが走る速度を測定した」という文では、「走る」がどちらの意味（人が走る/プログラムが走る）かによって、「プログラムが」の係り先が「走る」なのか「測定した」なのかが変わるだろう。どちらが正しいのかはおそらく、前後の文脈から「走る」がどちらの意味か（陸上競技の話なのかソフト開発の話なのか、等）から決まるだろう。つまり、語義が係り受けに影響するため、やはり逐次処理ではうまくいかない。

本稿では、前述した課題に向けて、新たなアプローチを提案する。まず「実体に関する情報を考慮していない」という欠点を克服するため、オブジェクト指向のアイデアを用いる。更に、逐次処理の限界を克服するため、係り受け・語義曖昧性解決・照応解析・述語項解析の4つの処理を同時に行う。この2つの特徴を持ったシステムを実装し、いくつかの例文に対して動作を確認済である。

2. 関連研究

依存構造解析と述語項解析を逐次ではなく同時に解決することの必要性は比較的理解が浸透しつつあり、近年は研究が増えてきている[1,2]。これは、CoNLLのShared Task[10]で採用されたことが大きく影響していると思われる。しかしながら、これらを語義曖昧性解決や照応解析と同時に解

決しようとする試みは見当たらない[4,7,11,12]。

単語の指すオブジェクトの実体を考慮する研究は、人文系の言語学サイドではアイデアとしては存在したが[13]、計算機による自然言語理解の研究で実際のシステム構築にまで踏み込んだものは他に見受けられない。しかし多少なりとも近いものを挙げるとすれば、照応解析におけるEntity-Mentionモデルがある[3]。一般的なモデルでは単語と単語のペアについて照応性を判定していくが、Entity-Mentionモデルにおいては、それまでに照応性が判明した単語どうしをまとめたクラスタとみなし、クラスタと単語の間の照応性を判定する。ここでのクラスタは本稿でいうオブジェクトに対応していると考えられるので、単語ペアのみを考慮するモデルよりは一歩進んでいると言える。

しかしながら従来のEntity-Mentionモデル研究では、クラスタに含まれる単語しか考慮せず、その単語の使われた文脈を見ていない。上述の、「彼」が指すのは「太郎」か「次郎」か?の例で言うと、「太郎」や「次郎」という単語（に基づく素性）だけを見ており、たとえば「太郎が犯罪を犯した」という文が前に出てきていたとしても、その「犯罪を犯した」という記述を利用できていない。この意味で、本稿で提唱しているオブジェクト指向的アプローチのメリットは享受できていなかった。

なお自然言語*理解*ではないが、物語を生成する研究においては、オブジェクトの属性をトラッキングするシステムの提案がなされている[8]。自然言語以外でも、ゲームのNPCのAIなどでもやはりオブジェクトの属性を管理して実現されるものがある。つまり、言語を理解する受け手の側でなく、発話する側ないし物語を進める側の観点からすると、このような「オブジェクトを想定し、その属性を管理していく」ことはきわめて自然な発想である。また、我々人間自身が物事を考えるプロセスを内省してみても、何かしらそのようなモデルが存在していることは間違いない。そして、発話の側にそのようなオブジェクトモデルが存在するのであれば、言語理解のタスクのゴールとして、受け手の側で発話側のオブジェクトモデルの再構築を試みることに、と定義するのは十分な妥当性があると考えられる。

3. 提案システム概要

本稿では、1)オブジェクト指向のアイデアを用い、2)係り受け・語義曖昧性解決・照応解析・述語項解析の4つの処理を同時に行う、という2つの特徴を持ったシステムを提案する。それぞれの特徴について、以下に説明する。

(1) オブジェクトと属性

基本アイデアとしては、名詞には対応するオブジェクトがあると考え、オブジェクトのテーブルを管理する。オブ

a <http://code.google.com/p/cabocha/>

ジェクトは属性を持つ。おおむね形容詞は属性に対応する。動詞によって表される動作やイベントによって、属性が変化する。オブジェクト指向のプログラミング言語との類推で言うと、名詞がクラス/インスタンス、形容詞がメンバ変数、動詞がメソッドにおおむね対応する。このような世界モデルを管理していくことで、世界を解釈・理解する。

連続した文を解析していくなかで、名詞が出てきたら、照応解析によって既存のオブジェクトを指しているか否かを判定する。Noならオブジェクトを新規生成する。Yesなら指すオブジェクトのリンクを貼る。述語(事象)が先行詞になる場合もあるので、述語にも(「事象」フラグをつけたうえで)オブジェクトを割り当てる。

各文により、「このオブジェクトのこの属性値をセット」のような情報が付加されていき、オブジェクトごとに属性がついていく。照応が成立した場合は先行詞オブジェクトの属性が照応詞に伝播される。

簡単な例で動作を説明する。次のような2つの連続した文があるとすると。

太郎は本を買った。
彼はその本を読むと、すぐに花子に渡した。

システムは最初の文を読むと、「太郎」・「本」・「買った(事象)」の3つのオブジェクトを登録する。「買った」ことの結果として、本の所有者属性には太郎をセットする。続いて次の文を読むと、照応解析により「彼」は太郎オブジェクトを、「その本」は本オブジェクトを指すことを理解する。またゼロ照応により、「渡した」の主語は太郎、直接目的語は本であることを理解する。「花子」、「読む(事象)」、「渡した(事象)」は新たなオブジェクトとして登録される。「渡した」の結果として、本の所有者属性が太郎だったのを花子にセットしなおす。このようにして文が順に処理されていく。

このように、オブジェクトとその属性、という考え方をとることにより、他にも副次的な利点がある。たとえば「長さ」という属性に対して、次のような互いに関連する一連の単語群が存在する。

長い : 属性値が高い状態 ('HIGH')
短い : 属性値が低い状態 ('LOW')
伸びる : 属性値が高くなる変化 ('UP')
縮む : 属性値が低くなる変化 ('DOWN')
長さ : 属性値が高いこと ('HIGHNESS') または 属性そのもの ('ATTR')
短さ : 属性値が低いこと ('LOWNESS')

長さの他にも、速さ(速い/遅い/速まる/...)、高さ(高い/低い/高まる/...)等でも同様である。

たとえば「太郎の背が伸びる」とあったら、太郎.身長.UP()のように表す(太郎オブジェクトの「身長」属性値が上がる)。このように表すことの利点の一つは、上の表現から太郎.身長.HIGH()に結果として成る、ということが、UP や HIGH の意味関係から自動的に導かれる。

つまり、関連する単語間の関係が属性を介して自然に反映されるため、わざわざ同義語・反意語や派生語の情報を辞書に入れる必要がなくなる。また、単語の数よりも属性の数の方が少ないため、辞書に記述すべき情報の量も減る。結果として、辞書作成の手間が削減される。言語理解システムの構築においては辞書作成の工数はかなりの割合を占めるため、このことのメリットは非常に大きい。

(2) 係り受け・語義曖昧性解決・照応解析・述語項解析の同時処理

各文節において係り先・語義・照応先・述語項のすべてについて、候補を列挙する。基本的には、候補のすべての組合せについて全幅探索を行う。即ち、各組合せに対して評価関数を定義し、全ての選択肢の組み合わせをトライして、評価が最大となる組合せを選ぶ。

全幅だと計算量が問題になる可能性があるが、係り受けでは全幅探索を行うシステムを既に試作しており、通常の長さの文なら問題なく瞬時に処理終了することが確認できている。そのため、本システムにおいても、計算量は致命的な問題にはならないであろうと予想している。仮に問題になった場合は、全幅探索の計算量を削減する手法として他分野で利用されている既知の手法[9]がいくつかあるので、それらをトライしてみる予定である。

語義の候補は辞書から直接得られる。述語項の候補は(ゼロ照応を除いて)係り受けから得られる。照応の候補生成については、現状では単語の類似度や照応詞との距離を元に選択している。

評価関数については、全幅探索のため、評価関数に対する制限はない。パーザの構成によっては、係り関係の2項間の情報しか使えない(3項以上は不可)といった制限がつく場合がある[14]が、本提案ではそのような制限はない。したがって、評価に寄与しそうなあらゆる項目が評価項目候補となりうる。

現在のところ、評価項目として考えているものを表1に示す。これらはすべてが揃わないとだめというものでもないので、当初は実装が容易でかつ重要そうなものから実装し、徐々に項目を増やしていく方針で開発を進めている。現状のプログラムでは、表中で'*'をつけたものはまだコーディングされていない。

評価項目	備考、例
述語と格補語名詞の相性 (従来技術)	連体節の述語とそれが係る 名詞、も含む
述語と格補語名詞オブジェ クトの属性の相性	連体節の述語とそれが係る 名詞、も含む
連体修飾～形容詞連体形と 名詞の相性	
連体修飾～形容詞連体形と 名詞オブジェクトの属性の 相性	
連体修飾～「(名詞 X)の(名 詞 Y)」のケースの、X と Y の関係	所属、部分、場所等
連用修飾～形容詞連用形/副 詞と述語の相性	
*副詞と機能表現の呼応	あまり～ない<否定>、た ぶん～だろう<推量>等
*時の副詞/副詞可能名詞と テンス・アスペクト	× 明日～した
*意志/依頼の現実化	太郎は留学したがって いた。去年留学した。
*名詞節が格補語となりや すい述語	言う、思う、わからない、 等
*主語とモダリティ	×私は食べたいそうだ
*語義の頻度	頻度の少ない語義は減点
*語義の一貫性	近くと同じ単語が異なる意 味だと減点
先行詞と照応詞の距離	
先行詞と照応詞の表層マッ チ度	一般名詞のケース
*照応における先行詞の Salience	主題、主節の主語・目的語 は加点
格の複数割り当ての回避	述語の一つの格に複数のオ ブジェクトがあると減点
*ゼロ照応での同一述語の 回避	同じ述語の複数の格に同一 オブジェクトが入ると減点
係り受けの距離	
コンマの直前から直後への 係りを回避 (並列除く)	
提題のハから動詞・形容詞 連体節への係りの回避	

表 1 評価項目一覧

Table 1 List of Evaluation Factors.

(3) システム構成

以上のアイデアに基づいたシステムを実装した。このシ
ステムの具体的なプログラム構成の概要を図 1 に示す。

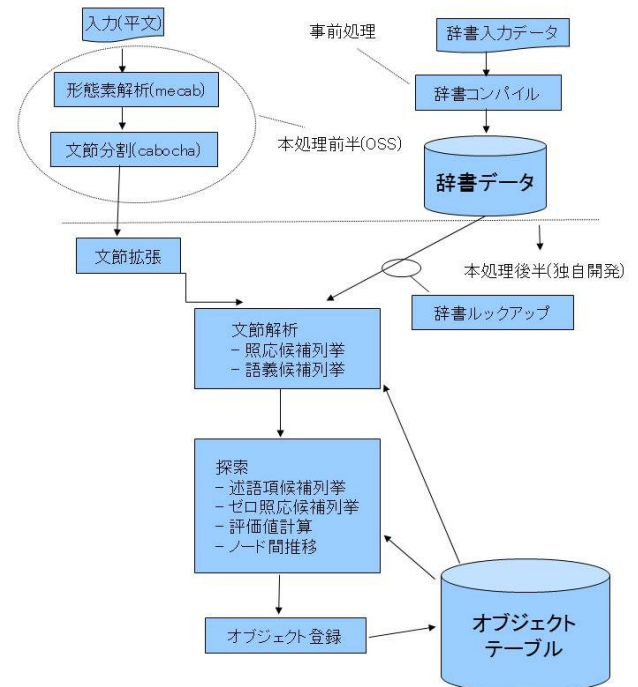


図 1 提案システムのブロック図

Figure 1 Block Diagram of the Proposed System.

大きく、事前処理と本処理に分かれる。事前処理では、
辞書用の (人が書いた) 入力データを読み込み、本処理で
使いやすいバイナリデータ形式に変換する。本処理は、入
力の平文を一文ずつ読み込んでいく。各文ごとに、形態素
解析、文節分割、文節拡張、文節解析、探索を順に行う。
それによって得られた文解析結果に従い、オブジェクトテ
ーブルを更新していく。

本処理中の形態素解析および文節分割は、オープンソー
スのツールと辞書を使用している(mecab, cabocha, IPAdic)。
それ以外のモジュールは独自に開発している。以下、主要
なモジュールについて説明する。

・辞書コンパイル

辞書を入力データをバイナリデータに変換する。辞書の
入力人は人が書くため、人が書き易いようなフォーマットに
する必要がある。現在は、入力フォーマットは YAML とし
て、辞書データベースには MongoDB を使っている。Ruby
スクリプトで YAML を読み込んでから、MongoDB の Ruby
ドライバで DB に書き出している。

・辞書ルックアップ

文章中の単語から辞書のエントリを探す。DB が MongoDB なので、DB からの読み出しには MongoDB の C++ API を使う。読み出した辞書データは、C++側で定義している構造体にセットされる。

・文節拡張

オープンソースのツールによる文節分割結果を、本システムで扱う形式に変換する。文節分割は `cabocha` のモジュールを用いているが、`cabocha+IPAdic` ではたとえば「XX するのが」と一文節になるが、「XX することが」と二文節に分かれる、という挙動になる。こうした差異を吸収し、解析エンジンの想定する文節分割に変換するのがこのモジュールの役目である。

・文節解析

探索中の計算量削減のため、文節ごとにできるだけ前もって情報を抽出しておく。全幅探索の場合、個々の文節には何度もアクセスする。アクセスのたびに文節を解析するのでなく、最初に一度だけ解析してできるだけ情報を拾っておき、探索中はできるだけ計算せず解析済みの情報を使うことによって計算時間を削減する。

・語義候補列挙

語義曖昧性解決の候補を列挙する。基本的には辞書に複数の語義が書いてあるので、それをそのまま取ってくるだけである。語義候補列挙は文節解析モジュールの中で呼ばれる。

・照応詞候補列挙

照応解析の候補を列挙する。照応解析の候補をどう決めるかは難しい問題で、まだどうするのが正しいかわかっていない面があるが、現在は比較的割り切った手法をとっている。

まず名詞が代名詞か、固有名詞か、一般名詞かに（辞書エントリを見て）分類する。

代名詞では、ヒト/モノ・人称・性別・数の情報だけを抽出しておき（「これ」「彼女ら」）、その情報に合う名詞で最近（一定文数以内に）出てきたものは全て候補とする。

固有名詞は、当面は単純に単語の一致を見るが、将来はそれだけでなく、敬称の区別（村山さん<->村山首相）等を扱う必要がある。

一般名詞では、照応詞に意味が「非常に近い」単語だけを候補とする。辞書エントリに類義語とその類似度をあらかじめ書いておく。

将来的には、機械学習的な手法で候補かそうでないかを分類するようなアプローチも有望だろう。

なお照応詞候補列挙には、係り受けや述語項の情報は用

いない。係り受けや述語項に関係なく、候補となる可能性があるならばとりあえず列挙しておく。それらが決まった後、その照応詞候補が述語に対して与えられた格として適切かどうかを判断するのは、評価関数の仕事である。照応候補列挙も文節解析モジュールの中で呼ばれる。

また、ここでいう照応にはゼロ照応は含まない。ゼロ照応は別モジュールで扱う。これは主に、ゼロ照応は述語項解析の結果を必要とし、したがって文節解析の段階ではまだ実行できないためである。

・探索

探索木をトラバースして、最大評価を与えるノードを決定する。当面は単純に全幅探索で実装する。計算量的に問題が出てくるようなら、モンテカルロや UCB といった、他分野で使われている計算量削減手法をトライしてみるようになるだろう。

探索の関数は3種類ある。1つめの関数は、文節番号 `c` を引数にとり、その文節における係り受け候補、語義候補、照応候補についてループし、ループの最奥部で、最後以外の文節ならば次の文節 `c+1` で自分を再帰呼び出しし、最後の文節では2つめの探索関数を呼ぶ。つまり、2つめの関数が呼ばれる時には、すべての係り受け、語義、照応（ゼロ照応を除く）は決定されている。2つめの関数は、各文節についてループして、各文節において述語項候補列挙ルーチンを呼ぶ。そしてそれぞれの述語項候補についての情報を「ループスタック」にプッシュしていく。その後、ループスタックを参照しながらすべての述語項候補の組み合わせをトライし、ループ最奥部で3つめの探索関数を呼ぶ。3つめの関数では、やはり文節についてループして、述語項も決定しているの、その情報を用いてゼロ照応候補列挙ルーチンを呼び、候補をループスタックにプッシュして、それからすべてのゼロ照応候補の組み合わせについてループし、評価関数を呼ぶ。このようにして、すべての可能性を調べる全幅探索を実現している。ループスタックを使用するのは、述語項やゼロ照応ではいくつの格を対象にするかが事前にわからず、ループの段数を可変にする必要があるためである。

・述語項候補列挙

述語項解析の候補を列挙する。述語項解析は係り受けの結果に大きく依存するため、係り受け候補選択の後に行う。係る側の文節に格助詞があれば、その助詞のヴォイスに応じた格（ヴォイスは最初に解析してある）。名詞+（係副）助詞、または助詞がない名詞では主格か目的格。連体節の場合も、連体節の係る名詞が連体節の述語の主格か目的格の候補と（当面は）する。述語項候補列挙もまだ正解がわからない部分があり、今後改善は必要になっていくだろう。格のダブリは候補列挙の段階では考慮せず、ダブリを減点

するのは評価関数の仕事である。

・ゼロ照応候補列挙

ゼロ照応解析の候補を列挙する。これは述語項候補選択の後に行う。

まず、どの格に対してゼロ照応を行うかを定める。各述語に対して、その単語によって属性が影響を受けるオブジェクトの格を「必須格」とする。例えば動詞「(太郎が花子に本を) 渡す」であれば、直接目的語(OBJECT)の「所有者」属性と、主格(AGENT)および間接目的語(GOAL)の「所有物」属性が変わるので、OBJECT, AGENT, GOAL が必須格となる。必須格のうち、述語項解析によって割り当てられていない格があればそれがゼロ照応の対象となる。

対象となる格に対して、照応候補となるオブジェクトを探す。述語と格がわかっているので、その述語と格に適合する(&最近出現した)オブジェクトが候補となる。この判定では、評価関数中に「オブジェクトの、与えられた述語と格に対する適合度」を計算する関数があるので、同じ関数が使える。

ゼロ照応候補列挙も、今後改善が必要な側面のひとつである。

・評価

各リーフノードに対して評価値を計算する。評価項目は表1に挙げた通りである。

・ノード遷移

探索木のトラバースでノードを遷移する際に各データ構造を変更・復元する。係り受け・語義・照応・述語項・ゼロ照応の5つに対して、候補を選択するごとに、選択したという事実をプログラム上のデータ構造に反映する。次の候補を選択する前に、前回の選択を外すのも、必要ならばやはりデータ構造に反映する。たとえば、各文節の構造に係り先文節を表す変数があるが、これは係り受け候補を選択するときその選択した文節にセットする。選択を外すときは無効な値(-1 等)にセットすることでその事実を示す。

・オブジェクト登録

得られた文章の解析結果に応じて、オブジェクトテーブルの属性等を変更する。たとえば「彼は花子に本を渡した」という文に対して、彼:AGENT、花子:GOAL、本:OBJECTと解析され、「彼」が「太郎」に照応している(太郎オブジェクトを指している)とする。このとき、「本 a」が本のオブジェクトを指しているとして、

- 太郎オブジェクトから「持っている(本 a)」属性を消す
- 花子オブジェクトに「持っている(本 a)」属性を加える
- 本 a オブジェクトに「所有者(太郎)」属性を消し、「所有者(花子)」属性を加える

という操作を行う。行うべき操作は、述語の単語(この例では「渡す」)の辞書エントリに、たとえば「GOAL オブジェクトに、『持っている(OBJECT)』属性を加える」のように書いてある。こうして、文ごとにオブジェクトの属性の変化をトラッキングしていく。

4. 現状と今後の展開

現在、システムの基本部分の実装を完了している。3(1)節に示したような簡単な例文を入力として与えて、正しく文を解析してオブジェクト登録するところまで動作の確認がとれている。今後は、より複雑な文構造や照応パターン、評価項目に対する対応コードを追加していく。また辞書についても語彙を増やしていく。

本システムの次のステップとして RTE (Recognizing Textual Entailment: 含意関係認識) システムが考えられる。動詞の直接の属性だけでなく、動詞の提示するフレームに関する世界知識を持たせ、常識的な推論をするようなシステムは本システムの自然な拡張として考えられるため、そのようなシステムを構築していく予定である。

本稿のアプローチでは、辞書作成が工数的にかなりの部分を占める。2千語がおおむね日本語能力試験[6]でいう N4 (基礎) レベル、1万語が N1 (上級) レベルと考えられるので、短期的 (~1年程度) にはまず2千語の辞書を作成して、基本的な会話が理解できるレベルを実現することを目指している。

照応解析・述語項解析については、まだ「どうするのが正しいのか」がわかっていない部分が多いため、まずは基本的な動作ができるシステムを作り、それでは対応できない複雑な文に対しては、どのようなケースが重要なのかを見極め、実験を重ねながら徐々に対応ケースを増やして精度を上げていく、といったアプローチが現実的だと考えており、そのように進めていく予定である。

5. おわりに

新たなアプローチに基づく自然言語理解システムを提案した。本システムは、オブジェクト指向のアイデアを用いて意味を表現し、また係り受け・語義曖昧性解決・照応解析・述語項解析の4つの処理を同時に行う。この2点により、従来技術によるものよりも深い言語理解が実現されることが期待される。システムは基本実装を完了し、例文に対して動作を確認している。今後システムをより発展させ、また実験結果を報告していきたい。

参考文献

- 1) Xavier Lluís, Xavier Carreras, and Lluís Marquez. Joint arc-factored parsing of syntactic and semantic dependencies. Transactions of the

- Association for Computational Linguistics (TACL), pp.219-230 (2013).
- 2) Kristina Toutanova, Aria Haghighi, and Christopher Manning. Joint learning improves semantic role labeling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005), pp.589-596.
 - 3) Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp.968-977.
 - 4) Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1-10:69 (2009)
 - 5) 工藤 拓, 松本 裕治 : チャンキングの段階適用による係り受け解析, 情報処理学会研究報告, SIG-NL-142, pp.97-104 (2001).
 - 6) 大隅 et al. 新しい日本語能力試験が目指すもの. 日本語教育学会 2009 年秋季大会シンポジウム配付資料.
 - 7) Pradhan, Sameer, et al. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. Joint Conference on EMNLP and CoNLL-Shared Task. Association for Computational Linguistics (2012).
 - 8) 秋元泰介, 栗澤康成, 福田至, 小方孝. 物語内容における状態を管理する機構の構築—状態—事象間関係の知識ベースの内容的検討—. 言語処理学会第 19 回年次大会, pp.378-381 (2013).
 - 9) Cameron Browne, Edward Powley, Daniel Whitehouse, Simon Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp.1-43 (2012).
 - 10) Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antonia Marti, Lluís Marquez, Adam Meyers, Joakim Nivre, Sebastian Pado, Jan Stepanek, Pavel Stranak, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, pp.1-18.
 - 11) Navigli, Roberto, Kenneth C. Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. Proceedings of the 4th International Workshop on Semantic Evaluations, pp.30-35 (2007).
 - 12) Poesio, Massimo, Simone Ponzetto, and Yannick Versley. Computational models of anaphora resolution: A survey. *Linguistic Issues in Language Technology* (2011).
 - 13) Heim, Irene. The semantics of definite and indefinite noun phrases. Diss. University of Massachusetts (1982).
 - 14) Koo, Terry, and Michael Collins. Efficient third-order dependency parsers. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (2010).