構文・述語項構造解析システム KNP の解析の流れと特徴

笹野 遼平 †　河原 大輔 ‡　黒橋 禎夫 ‡　奥村 学 †

{sasano,oku}@pi.titech.ac.jp　{dk,kuro}@i.kyoto-u.ac.jp

† 東京工業大学 精密工学研究所　‡ 京都大学 大学院情報学研究科

1　はじめに

当初，ルールベースの構文解析システムとして公開された KNP であるが，その最新版である KNP 4.1[1]では大規模な格フレームに基づく格解析やゼロ照応解析などを行うことが可能である．本稿では，最新版の KNP がサポートしている解析の概要，および，その使い方と特徴をまとめる．

2　解析の流れと利用する知識

構文・格解析　KNP はデフォルトでは自動構築した大規模格フレームに基づく統合的確率モデル [8] により構文・格解析を行う．解析の概要は以下のようにまとめられる．

1. 形態素の品詞・活用，機能語などの情報に基づき可能な構文・格構造を絞り込む
2. 統計的情報に基づき各構文・格構造の生成確率を算出し，確率最大の構文・格構造を出力する

KNP はこれらの処理を大規模ウェブテキストから自動的に構築した格フレーム [7] を用いて行っており，構文・格構造解析を行う際，係り受け構造や，述語と係り受け関係にある各述語項がどのような格として出現しているかだけでなく，格フレームの選択も行っている．KNP が使用している格フレームは用言の意味，用法ごとに構築されているため，用言の意味曖昧性の解消も行っているとみなせる．

構文・格解析の単位は文であり，1 文ごとにもっとも可能性の高い構文・格構造，および，述語ごとに選択した格フレームを出力する．ヲ格の項は多くの場合もっとも近い述語に係るなど，どのような構文となりやすいかの選好を反映する構文の生成確率は，係り受け情報付与済みコーパス[2]を用いて計算される．一方，語彙選好を反映する語の生成確率は 70 億文から自動構築した格フレームなど大量の生テキストから獲得した知識に基づき計算される．

談話解析　KNP 4.1 では “-anaphora” というオプションを付与することにより照応関係と述語項構造を統合的に解析することが可能である．照応解析において，“-ne” というオプションを指定することにより得られる固有表現情報を利用しているため，“-anaphora” オプションを指定する場合は，“-ne” オプションも付与することが望ましい．これらのオプションを指定した場合の解析の流れは以下のようにまとめられる．

1. 構文・格解析を行う
2. CRF に基づく固有表現解析を行う
3. ルールに基づく共参照解析を行う
4. 省略された項の同定（ゼロ照応解析）も含む述語項構造解析を行う

CRF に基づく固有表現解析は，形態素を解析の単位とし，一般的に使用される前後それぞれ 2 形態素の出現形，品詞などに加え，大域的な情報として，先行文脈における固有表現解析結果や，構文解析結果から得られる係り先の情報[3]などを素性として使用している．また，共参照解析は同義表現に関する知識，および，文字列のマッチングをベースとしたルール[4]に基づき行っている．

省略された項の同定も含む述語項構造解析は大規模格フレームを用いた識別モデルに基づく手法 [11] で行っている．解析の単位は述語であり，共参照関係にある表現を 1 つにまとめた談話要素（ENTITY）の中から述語の項を選択している．この際，構文・格解析によって得られた係り受け関係は入力として与えるが，格解析結果や格フレーム選択結果は入力として与えず，省略された項も考慮した上で新たに決定する．

[1] KNP 4.1: http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?京都大学テキストコーパス

[3] 文献 [10] におけるキャッシュ素性，および，係り先素性に相当．

[4] 基本的に文献 [9] における共参照関係認定基準 1 を使用．

```
1: # S-ID:1
2: * 1D <文頭><ヲ><助詞><体言><係:ヲ格><区切:0-0><格要素><連用要素><正規化代表表記:女の子/おんなのこ><主辞代表表記:女の子/おんなのこ> ···
3: + 1D <文頭><ヲ><助詞><体言><係:ヲ格><区切:0-0><格要素><連用要素><先行詞候補><正規化代表表記:女の子/おんなのこ><解析格:ヲ><EID:0>
4: 女の子 おんなのこ 女の子 名詞 6 普通名詞 1 * 0 * 0 "代表表記:女の子/おんなのこ カテゴリ:人 ドメイン:家庭・暮らし" ···
5: を を を 助詞 9 格助詞 1 * 0 * 0 NIL <かな漢字><ひらがな><付属>
6: * -1D <文末><時制-過去><句点><用言:動><係:文末><主節><格要素>< 連用要素><正規化代表表記:見掛ける/みかける> ···
7: + -1D···<格解析結果:見掛ける/みかける:動 1:ガ/U/-/-/-/-;ヲ/C/女の子/0/0/1;···><EID:1><述語項構造:見掛ける/みかける:動 1:ヲ/C/女の子/0>
8: 見かけた みかけた 見かける 動詞 2 * 0 母音動詞 1 タ形 10 "代表表記:見掛ける/みかける" ···
9: 。 。 。 特殊 1 句点 1 * 0 * 0 NIL <文末><英記号><記号><付属>
10: EOS
```

図 1: -tab オプションによる詳細出力の例（入力:「女の子を見かけた。」）

これは，明示的に格助詞を伴って解析対象の述語に係る項がない場合など，省略された項を考慮しないと適切な格フレーム選択や格解析が行われない場合があるためである．

(1) 太郎が公園に戻ってくるとさきほど 見かけた 女の子はいなくなっていた。

たとえば上記のような文があった場合，省略された項を考慮しない解析，すなわち，デフォルトの構文・格解析では,「見かけた」が「女の子」を連体修飾しているということは正しく解析できるものの,「女の子」は「見かけた」のガ格であると解析されてしまう．しかし，文頭に出現した「太郎」も考慮に入れ，新たに「女の子」が「見かけた」のヲ格である可能性も含めて述語項構造解析を行うことにより,「太郎」が「見かけた」のガ格であり,「女の子」は「見かけた」のヲ格であると正しく解析することができるようになる．

デフォルトの構文・格解析の場合と同様に，テキスト中のどの位置に出現した談話要素がガ格先行詞となりやすいかなど，語彙に依存しない談話的な選好は人手で作成された述語項構造情報付与済みコーパス [1] から獲得し，語彙的選好は大規模格フレームなど大量の生テキストから自動構築した知識から獲得している．

3 KNP の出力フォーマット

KNP は基本的に以下のオプションで指定できる 4 つの出力形式で解析結果の出力を行う．

- -tree: 木構造の出力 (デフォルト)
- -tab: 詳細出力
- -simple: -tab 出力のシンプル版
- -td: 解析結果汎用表示ツール[5]用の出力

デフォルトでは “-tree” オプションを指定した場合と同様に係り受け木の出力を行う．係り受け解析結果を確認するのに適した出力であるが，格解析結果などの詳細解析結果は表示されず，また，KNP の解析結果を計算機で処理したい場合には向いていない．

[5]http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/tabledisplay/

“-tab” オプションを指定すると，図 1 に示すような計算機で扱うのに適した形式で詳細解析結果が出力される．“#” から始まる行は新しい文の始まり，“EOS” は文の終わりであることをそれぞれ意味する．“*” から始まる行は文節，“+” から始まる行は基本句,「女の子」,「を」など形態素から始まる行は形態素の情報をそれぞれ記述している．ここで基本句とは，KNP で格解析などの処理を行う基本単位であり，1 つの自立語と後続する付属語から構成される．

格解析結果や共参照解析結果，述語項構造解析結果等は基本句行に山括弧 “<>” で囲まれた feature として出力される．まず，デフォルトで行われる格解析結果は以下のフォーマットで出力される．

<格解析結果:格フレーム ID:格要素群>

“格フレーム ID” は格解析で選択された格フレームの ID を表し，格要素群は格ごとに “;” で区切られ，格フレームに存在するすべての格の格要素について以下のフォーマットで解析結果が記述される．

格/フラグ/表記/基本句番号/0[6]/文 ID

“フラグ” は格助詞により格が明示されている場合は “C”，被連体修飾詞であるなど格が明示されていない場合は “N”，格要素の割り当てがない場合は “U” となる．たとえば図 1 の 7 行目の基本句行に含まれる格解析結果 feature は，“見掛ける/みかける:動 1” という ID の格フレームが選択され，ガ格は割り当てられず，ヲ格の格要素は明示的に格助詞を伴い（C），0 番目の基本句として出現した「女の子」という表現であるということを意味する．

談話構造解析結果は “EID” と “述語項構造” という 2 つの feature により記述される．EID は基本句ごとに付与される談話要素番号であり，共参照関係にある基本句には同一の EID が，それ以外の場合は異なる EID が付与される．また，述語項構造は格解析結果と同様のフォーマットで出力される．

[6]格要素が述語の何文前に出現したかを表す欄であるが，現在は係り受け関係にある格要素のみを考慮しているため常に 0 となる．

<述語項構造:格フレーム ID:格要素群>

ただし，格要素のフォーマットは格解析結果と異なっており，以下のフォーマットで出力される．

格/フラグ/表記/EID

ゼロ照応解析の結果対応付けられた格要素である場合はフラグは “O” となる．また，出力されるのは格要素が割り当てられた格のみであるためフラグが “U” となることはない．たとえば図 1 の 7 行目の基本句行に含まれる述語項構造 feature は，“見掛ける/みかける:動 1” という ID の格フレームが選択され，ヲ格の格要素は直接係り受け関係にあり格も明示された（C），EID が 0 である「女の子」という表現であるということを意味する．

“-tab” オプションを指定した場合の出力は，非常に多くの feature が出力されることから，人間にとって読みやすいとは言いがたい．このため，人間が解析結果を分析しやすいように “-simple” オプションと，“-td”（table display）オプションが用意されている．前者は “-tab” オプションで出力される feature のうち特に重要なもののみを出力するオプションである．デフォルト解析では “格解析結果” などが，談話解析を行う場合は，“EID”，“述語項構造” など数種の feature のみが出力される．

後者は解析結果を解析結果汎用表示ツール[5]を用いてブラウザで閲覧することを想定したオプションである．出力を解析結果汎用表示ツールで読み込むと，“-tree” オプションで表示される係り受け木が表示され，さらに詳細を知りたい基本句にマウスを当てることで，“-tab” オプションで出力される対象の基本句に関連する詳細解析結果がツールチップとして表示される．“-anaphora” オプションが指定されている場合はさらに EID と述語項構造解析結果が表示され，述語項構造解析結果にマウスを当てることで，10-best の述語項構造を閲覧することが可能である．“-anaphora”，“-ne” オプションを指定した場合の解析結果汎用表示ツールによる解析結果の表示例を図 2 に示す．

4 KNP の特徴

新聞記事，Web から収集したテキスト各約 40 万文を，SVM に基づく日本語係り受け解析器である CaboCha 0.66[7]，ゼロ照応も考慮した日本語の述語項

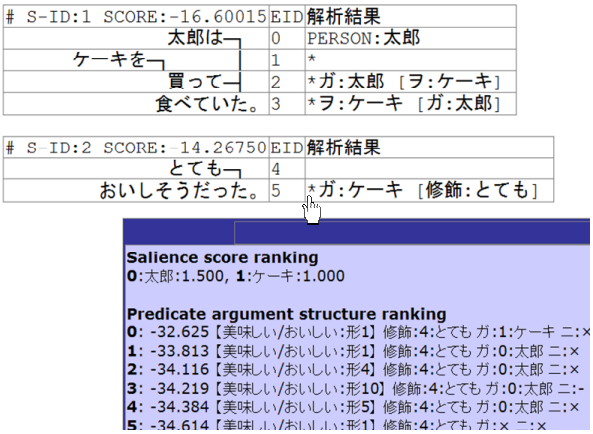

図 2: 述語項構造解析結果の解析結果汎用表示ツールによる表示例（入力:「太郎はケーキを買って食べていた。とてもおいしそうだった。」）

表 1: 解析時間（秒）の比較

対象	新聞記事	Web
文書数	696	500
文数	9,284	15,415
文字数	404,007	397,047
CaboCha 0.66	5.16	4.57
SynCha 0.3 [n=2]	2,984	1,602
SynCha 0.3 [n=∞]	9,031	17,161
KNP 4.1 [談話解析なし]	5,905	5,405
KNP 4.1 [談話解析あり]	7,595	6,889

構造解析器である SynCha 0.3[8]，および，KNP 4.1 で解析したときの解析時間の一覧を表 1 に示す．解析はいずれも同一の Linux 環境（CPU: Intel Xeon 3.33GHz）で行い，SynCha 0.3 では先行詞の探索範囲をデフォルトである 2 文（n=2）とした場合と，1 記事の最大文数より大きくした場合（n=∞）の 2 条件で，KNP 4.1 は談話解析を行わない場合と，行う場合（解析オプションとして “-anaphora -ne” を指定）の 2 条件で実験を行った．

KNP の解析速度は，デフォルトの設定で 1 秒あたり新聞記事であれば 1.6 文，Web から収集したテキストであれば 2.9 文，談話構造解析も行う場合はそれぞれ 1.2 文，2.2 文であり，SynCha と同程度の解析速度であった．

係り受け解析器 CaboCha と比べると，KNP の解析速度は圧倒的に遅く，大規模に係り受け解析を行いたい場合などは KNP より CaboCha が向いていると言える．しかし，KNP は述語項構造解析など CaboCha よりも深い解析を行っており，単純な係り受け構造では表現されないテキストの意味を扱いたい場合は KNP

[7] http://code.google.com/p/cabocha/

[8] http://www.cl.cs.titech.ac.jp/~ryu-i/syncha/

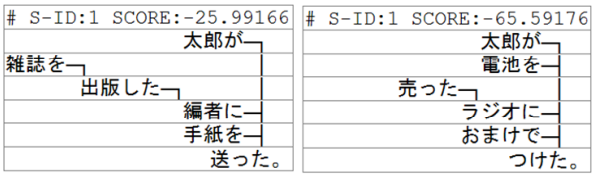

図 3: ガーデンパス文の解析例（入力:「太郎が雑誌を出版した編者に手紙を送った。」,「太郎が電池を売ったラジオにおまけでつけた。」）

が向いていると考えられる．また，浅原ら [6] が報告しているように，CaboCha などの係り受け解析ツールと比べガーデンパス文のような語彙的選好や広い文脈を考慮する必要がある文であっても頑健に解析できるという特徴がある．これは，KNP は大規模格フレームから算出される語彙的選好を利用しており，また，CaboCha のように直後の文節に係るか係らないかという観点のみで決定的に解析を行わず，文全体で最適な係り受けを決定しているためであると考えられる．図 3 に以下の 2 つのガーデンパス文を KNP を用いて解析した場合の解析例を示す．

(2) 太郎が雑誌を出版した編者に手紙を送った。

(3) 太郎が電池を売ったラジオにおまけでつけた。

これらの文を CaboCha で解析すると「太郎」の係り先はそれぞれ「出版した」,「売った」であると解析されるのに対し，KNP ではそれぞれ「送った」,「つけた」であると正しく解析することができる．

次に，ゼロ照応解析の精度を，関連研究の精度と比較した結果を表 2 に示す．表 2 中の Iida ら [2] のモデルは n=∞とした場合の SynCha 0.3 に相当する．いずれの実験も NAIST テキストコーパス 1.4β のうち 1 月 1 日から 11 日の通常記事，1 月から 8 月の社説記事を訓練データに，1 月 12 日，13 日の通常記事，9 月の社説記事を開発データ[9]に，1 月 14 日から 17 日の通常記事，10 月から 12 月の社説記事をテストデータとして実験した結果である．ただし，KNP 4.1 の精度は KNP において動詞・形容詞であると判断されたもののみを対象として算出した値である．また，NAIST テキストコーパスは原形に対する格構造が付与されているのに対し，KNP では出現形の格構造解析を行うことから，受身形・使役形で出現した述語も解析対象から除いている．このため，これらの精度は単純には比較できないが，KNP のゼロ照応解析の精度は関連研究の精度と同程度であり，文間のガ格，ヲ格の解析精度が他の手法に比べ高い傾向があると言える．

[9]KNP4.1 で採用されているパラメータも同様の訓練データ，開発データを用い決定したものである．

表 2: ゼロ照応解析の精度比較（F 値）

	ガ			ヲ			ニ		
	文内	文間	合計	文内	文間	合計	文内	文間	合計
Taira ら [4]	30.2	23.5	-	11.4	9.3	-	3.7	11.8	-
Imamura ら [3]	50.0	13.1	38.6	30.8	0.7	24.5	0.0	0.0	0.0
Iida ら [2]	-	-	34.6	-	-	-	-	-	-
Yoshikawaら[5]	54.1	-	-	10.3	-	-	0.0	-	-
KNP 4.1	41.6	26.3	35.7	22.6	13.8	20.1	8.8	1.0	5.9

5 おわりに

本稿では，最新版の KNP がサポートしている解析の概要，および，その使い方と特徴をまとめた．KNP は単純な係り受け解析器と比べると解析速度は遅いものの，語彙的選好を考慮する必要があるガーデンパス文の構文構造や，文間のゼロ照応関係を比較的高精度に解析することができるシステムであると言える．今後の課題としては，ゼロ照応解析の精度向上，原形格構造の出力機能の追加などが挙げられる．

参考文献

[1] Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proc. of ACL'07 Workshop: Linguistic Annotation Workshop*, pp. 132–139, 2007.

[2] Ryu Iida and Massimo Poesio. A cross-lingual ilp solution to zero anaphora resolution. In *Proc. of ACL-HLT'11*, pp. 804–813, 2011.

[3] Kenji Imamura, Kuniko Saito, and Tomoko Izumi. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. In *Proc. of ACL-IJCNLP'09*, pp. 85–88, 2009.

[4] Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. A Japanese predicate argument structure analysis using decision lists. In *Proc. of EMNLP'08*, pp. 523–532, 2008.

[5] Katsumasa Yoshikawa, Masayuki Asahara, and Yuji Matsumoto. Jointly extracting japanese predicate-argument relation with markov logic. In *Proc. of IJCNLP'11*, pp. 1125–1133, 2011.

[6] 浅原正幸. shWiiFit reduce dependency parsing. 自然言語処理, Vol. 18, No. 4, pp. 351–366, 2011.

[7] 河原大輔, 黒橋禎夫. 格フレーム辞書の漸次的自動構築. 自然言語処理, Vol. 12, No. 2, pp. 109–131, 2005.

[8] 河原大輔, 黒橋禎夫. 自動構築した大規模格フレームに基づく構文・格解析の統合的確率モデル. 自然言語処理, Vol. 14, No. 4, pp. 67–81, 2007.

[9] 笹野遼平, 黒橋禎夫. 自動獲得した名詞関係辞書に基づく共参照解析の高度化. 自然言語処理, Vol. 15, No. 5, pp. 99–118, 2008.

[10] 笹野遼平, 黒橋禎夫. 大域的情報を用いた日本語固有表現認識. 情報処理学会論文誌, Vol. 49, No. 11, pp. 3765–3776, 2008.

[11] 笹野遼平, 黒橋禎夫. 大規模格フレームを用いた識別モデルに基づく日本語ゼロ照応解析. 情報処理学会論文誌, Vol. 52, No. 12, pp. 3328–3337, 2011.