語彙的・構造意味情報を用いた語義タギング

田中貴秋 † **Francis Bond**♢ **藤田早苗** † **橋本力** ♣
† NTT コミュニケーション科学基礎研究所
♢ 情報通信研究機構
♣ 山形大学工学部

1 はじめに

WordNet[1] や PropBank[2] のような語彙的な意味情報や構造的な意味情報を持つ言語資源が利用できるようになり，意味情報を積極的に活用した高度な自然言語処理が期待されている．一方で，これらと同等の情報を出力できる意味解析システムは実現されていない．本来意味解析は，語彙的，構造的意味情報が補完しあって行われるべきだと考えられるが，本稿では，様々な情報を利用して，語彙的意味情報である語義を確定する (語義曖昧性解消) 方法について焦点を当てる．辞書，オントロジ，ドメイン辞書から得られる語彙的意味情報，HPSG に基づく解析結果から得られる構造的意味情報を組み合わせて入力文の単語に語義タグを付与する方法について述べる．最後に，評価実験の結果について考察する．

2 語義タギング

本稿のタスクは，入力された文中の単語に対して，辞書で定義された語義タグを割り当てることである (ここでは語義タギングと呼ぶ)．SENSEVAL の all words タスクと同様であるが，対象語を後述する辞書 Lexeed に収録されている基本語 28,000 語のみに限定している．

語義タギングでは，形態素解析，構文解析，オントロジなど異なる情報を組み合わせて利用する．語義選択のモデル作成時は，これらの情報を含むコーパスを統合して訓練データとする．タギング時には，形態素解析や構文解析などの他の解析器 (一次解析器) から得られた情報を統合して利用する．

語義タギングには，次のような処理が必要となる．

入力: 入力文の複数の一次解析器による解析結果間で単語の対応付けを行い，統合した解析結果を得る

特徴量抽出: 統合された一次解析結果から，語義選択モデルに関わる特徴量を抽出する

解探索: 語義選択モデルに基づいて各単語に対する語義の尤度を計算し，探索により最適な語義タグの組合せを求める

本稿では，このうち特徴量抽出と解探索について述べる．以下，3 節で，使用した言語リソースについて説明し，4 節では，語義選択モデルに使用した素性について述べる．5 節で，語義タギング時に行う語義の組合せの探索方法について述べ，6 節で，評価実験結果について述べる．

3 檜コーパス

檜コーパス [3] は，日本語語彙データベース Lexeed [4] と，単語の意味情報 (語義タグ) や構文情報を付加したコーパスから構成されている．

3.1 Lexeed

Lexeed は，国語辞典のように語義を定義するデータベースであり，日本人に馴染みの深い 28,000 語 (基本語と呼ぶ) を見出語として収録している．この基本語は，評定実験により単語の親密度を 7 段階で計測した結果，評定値が 5 以上の語である．Lexeed では，基本語に対して合計 46,000 の語義が定義文により定義されており，各語義は例文を持つ．定義文，例文は，基本語と機能語のみで記述されている．

本稿の語義タギングでは，Lexeed で定義された語義を使用して，各単語に語義タグを割り当てていく．表 1 は，Lexeed の見出し語一単語あたりの語義数の分布である．単語親密度の低い語ほど一単語あたりの語義数は減少し，多義性の無い語が増える傾向があるため，語義選択が困難な語の多くは基本語に含まれていると考えられる．

単語親密度	単語数	多義語	平均語義数	単義語 (%)
6.5 -	368	182	4.0	186 (50.5)
6.0 -	4,445	1,902	3.4	2,543 (57.2)
5.5 -	9,814	3,502	2.7	6,312 (64.3)
5.0 -	11,430	3,457	2.5	7,973 (69.8)

表 1: Lexeed の見出し語一語あたりの語義数の分布

3.2 オントロジとシソーラス

語彙的意味の情報として，Lexeed の語義間の関係を持つ檜オントロジ [5] を使用した．このオントロジは，定義文を解析した結果から自動構築したものを元にして一部人手で修正が施されている．同義関係，上位-下位関係を中心として約 8 万のリンクを持っている．語義単位の対応関係であるので，同じ単語の表層形であっても複数の語義がある場合はそれぞれ区別されている．

また，Lexeed の各語義には，日本語語彙大系 [6] の約 2,700 分類の一般名詞意味属性と 130 分類の固有名詞意味属性が付与されている．これらの意味属性は元々名詞の体系で，日本語彙大系では，固有名詞を含む 400,000 語の名詞に付けられているが，Lexeed では，他の品詞を含むほぼ全ての語義にて付与している．

3.3 語義タグ付きコーパス

Lexeed の定義文 (LXD-DEF)，Lexeed 例文 (LXD-EX)，京大コーパス (KYOTO)[7] に出現した基本語 (Lexeed の見出語) に対して，人手により Lexeed の語義タグを付加した [8]. 各単語あたり平均 5 人のアノテータによりタグ付けを行い，全員が一致しない場合は最も多くのアノテータが選択したタグを採用した．各トークンのアノテータ間の平均一致度は，最も低い京大コーパスで 78.7%，最も高い Lexeed の定義文で 83.3%であった．これらの値が，語義タギングの性能の上限値と考えることができる．表 2 に，各コーパス別のタグ付けされた単語の数を示す.

コーパス	単語総数	基本語	単義語
LXD-DEF	691,072	318,181	31.7
LXD-EX	498,977	221,224	30.5
KYOTO	969,558	472,419	36.3

表 2: 使用コーパスの統計

3.4 ツリーバンク

語義タグ付きコーパスと同じコーパスに対して HPSG (Head-driven Phrase Structure Grammar)[9] に基づいて統語情報，構造的意味情報のアノテーションを行いツリーバンクを作成した．これは，日本語 HPSG である JACY[10] を用いて複数の解析候補を機械的に出力させ，その中から正解の解析結果を選択する方法で構築している．この方法の利点は，作成したアノテーションは解析器で使用する文法に適合しており，同一の文法で作成したツリーバンクは一貫性が保たれる点である．弱点は，解析器が解析を失敗する文にはアノテーションが行えないことである.

ツリーバンクは，統語情報だけでなく，Minimal Recursion Semantics (MRS) [11] という構造的意味表現 (semantic dependency) を付与している．これには述語-項構造の情報も含まれる．このような構造的な意味情報は単独ではデータスパースネスに弱いが語彙的な意味情報を組み合わせてより抽象化することによって語彙選択に有効な情報を獲得できると考えている.

4 語義選択モデル

Lexeed の基本語は品詞によって，名詞，動詞，サ変名詞，形容詞，副詞，その他に大きく分類することができる．本稿では，その他以外の 5 分類に属する単語を語義選択の対象とした．各品詞ごとに，語義選択モデルを最大エントロピモデルとして構築した．以下では，モデル構築に使用した素性について述べる.

4.1 単語共起

語義選択の基本素性として n-gram と文内共起を用いた．n-gram は，対象語の前方および後方の unigram, bigram, trigram を使用した．また，文内共起は，ウィンドウを設けず，同一文内の内容語全てを共起語として使用した.

#	sample features
C1	⟨COLWS:人 $_4$⟩
C2	⟨COLWS_{SC}:C33:**他人**⟩
C3	⟨$\text{COLWS}_{\text{HYP}}$:人間 $_1$⟩
C4	⟨$\text{COLWS}_{\text{HYPSC}}$:C5:**人間**⟩
C1	⟨COLWS:電車 $_1$⟩
C2	⟨COLWS_{SC}:C988:**乗り物（陸圏）**⟩
C3	⟨$\text{COLWS}_{\text{HYP}}$:車両 $_1$⟩
C4	⟨$\text{COLWS}_{\text{HYPSC}}$:C988:**乗り物（陸圏）**⟩
C1	⟨COLWS:自動車 $_1$⟩
C2	⟨COLWS_{SC}:C988:**乗り物（陸圏）**⟩
C3	⟨$\text{COLWS}_{\text{HYP}}$:車 $_2$⟩
C4	⟨$\text{COLWS}_{\text{HYPSC}}$:C988:**乗り物（陸圏）**⟩

図 1: 語義共起素性 (SEM-Col) の例

4.2 語義共起

単語の表層形による素性はデータスパースネスの影響を受けやすいので，語義タグ情報とオントロジ・シソーラスから得られる語彙的意味情報を利用して同一文内の単語共起の情報を拡張した (これを語義拡張と呼ぶ)．3.2 節で述べたように，オントロジ・シソーラスは Lexeed の語義単位で定義されているので，オントロジによる同義語，上位語，日本語語彙大系の意味属性は文脈によって区別される．例えば，「運転」の場合，「乗物や機械を運転する」という意味では，意味属性 ⟨C2003:**操縦**⟩ が，「資金を活用する」という意味では，意味属性 ⟨C2005:**使用**⟩，上位語「活用 $_2$」に拡張される．ただし，この拡張は 5 節で述べる探索の過程で，どちらかの語義に仮定されている場合のみ行われ，多義性が残っている場合には行われない.

図 1 は，「運転手 $_1$」[1]の定義文「電車や車を運転する人」から得られる語義共起素性の例である．最初の列は，素性のテンプレートを示している．素性は，例に示している，単語文内共起を語義 (COLWS)，同意語義 ($\text{COLWS}_{\text{syn}}$)，上位語義 ($\text{COLWS}_{\text{hyp}}$)，意味属性 ($\text{COLWS}_{\text{SC}}$) で語義拡張したものの他に，複合語の構成語の語義の判別のために対象語の直前，直後の単語を語義拡張した情報も区別して使用している.

4.3 意味構造

意味構造素性は，HPSG パーザの解析結果に含まれる述語-項構造を展開して抽出する．例えば，「電車や車を運転する人」の意味表現からは図 2 のような素性が得られる．述語「運転する」は，二つの項 ARG1 「人」と ARG2 「や」を持ち，並列項である ARG2 「や」は，ARG2 「電車」と ARG2 「自動車」に展開される．この構造から各述語-項の

[1]単語の右下の添字は，語義番号を表す

#	sample features for 運転する $_1$
D1	⟨PRED:運転する, ARG1:人⟩
D1	⟨PRED:運転する, ARG2:電車⟩
D1	⟨PRED:運転する, ARG2:自動車⟩
D2	⟨PRED:運転する, ARG1:人 $_4$⟩
D2	⟨PRED:運転する, ARG2:電車 $_1$⟩
D2	⟨PRED:運転する, ARG2:自動車 $_1$⟩
D3	⟨PRED:運転する, ARG1$_{\mathsf{SC}}$:C33⟩
D3	⟨PRED:運転する, ARG2$_{\mathsf{SC}}$:C988⟩
D4	⟨PRED:運転する, ARG2$_{\mathsf{SYN}}$:モーターカー $_1$⟩
D5	⟨PRED:運転する, ARG1$_{\mathsf{HYP}}$:人間 $_1$⟩
D5	⟨PRED:運転する, ARG2$_{\mathsf{HYP}}$:車両 $_1$⟩
D5	⟨PRED:運転する, ARG2$_{\mathsf{HYP}}$:車 $_2$⟩
D6	⟨PRED:運転する, ARG1$_{\mathsf{HYPSC}}$:C5⟩
D6	⟨PRED:運転する, ARG2$_{\mathsf{HYPSC}}$:C988⟩
D11	⟨PRED:運転する, ARG1:人, ARG2:電車⟩
D22	⟨PRED:運転する, ARG1:人 $_4$, ARG2:電車 $_1$⟩
D23	⟨PRED:運転する, ARG1:人 $_4$, ARG2:C1460 ⟩
D24	⟨PRED:運転する, ARG1:人 $_4$, ARG2$_{\mathsf{SYN}}$:モーターカー $_1$⟩
D32	⟨PRED:運転する, ARG1:C5, ARG2:電車 $_1$⟩
D33	⟨PRED:運転する, ARG1:C5, ARG2:C988⟩
D55	⟨PRED:運転する, ARG1$_{\mathsf{HYP}}$:人間 $_4$, ARG2$_{\mathsf{HYP}}$:車両 $_1$⟩
D56	⟨PRED:運転する, ARG1$_{\mathsf{HYP}}$:人間 $_4$, ARG2$_{\mathsf{HYPSC}}$:C988⟩
D65	⟨PRED:運転する, ARG1$_{\mathsf{HYPSC}}$:C5 , ARG2$_{\mathsf{HYP}}$:車両 $_1$⟩
D322	⟨PRED:C2003, ARG1:人 $_4$, ARG2:電車 $_1$⟩

図 2: 意味構造素性 (SEM-Dep) の例

関係を展開して D11 のような全ての項を持つ素性，D1-D3 のような単項のみを持つ素性を得る．

述語および各項のうち多義性のないもの，探索時に語義が仮定されたものは，オントロジ・シソーラスを使用して語義拡張される．この例では,「電車 $_1$」と「自動車 $_1$」は，両者ともに意味属性 ⟨C988:**乗物** (**陸圏**)⟩ に拡張される.「運転」,「人」は，語義がそれぞれ「運転 $_2$」,「人 $_4$」に仮定されていた場合，⟨C2003:**操縦**⟩，⟨C5:**人間**⟩ に拡張される．

4.4 ドメイン

対象文の書かれているドメインの情報は，時に語義を決定する有力な手がかりとなる.「文化・芸術」,「スポーツ」のような，Open Directory Project など WWW のディレクトリ型検索サイトで大分類に使用されているレベルの 12 カテゴリを定義して用いた.「包丁」や「ホームラン」など出現する典型的なドメインがある語に対してこのドメインカテゴリを付与した． WWW 上の分布を使用して自動的に付与を行った [12] 結果，JUMAN[2] の辞書に使用されている語のうち約 6,000 語にドメイン情報が付与した．

5 最適語義タグ組合せの探索

語義選択モデルにより個々の単語の出現に対する各語義の条件付き確率を求めることができる．初期状態では，多義性のない単語を除いて語義が決定していないので，これらの語について，語義共起，および意味構造素性における語義拡張を使用することはできない．各単語に対する語義の全ての組合せについて考えるのは数が多く現実的でないので，ビーム探索によって有望な組合せを求める．

以下にそのアルゴリズムについて説明する．

入力文 S の内容語 (Lexeed の基本語) のうち多義性のある単語集合 $\{w_1,...,w_n\}$ について，語義が決定されていない単語集合を W，単語 w_i の k 番目の語義を t_{w_ik}，決定された語義のリストを T とする．探索は，$N=[W,T]$ をノードとする木に対して行う，またノードのスコア $s(N)$ を文脈 C のもとで語義タグの集合 T が現れる確率と定義する．また，ビーム幅を b とすると以下の手続きで最終的な語義タグの組合せを求める．

1. 初期ノード $N_0=[T_0,W_0]$, $T_0=\{\}$, $W_0=\{\}$ を初期キュー Q_0 に挿入する．
2. キュー Q 中の全ての N に対して以下を行う
   - 全ての w_i $(\in W)$ それぞれに対して，W から w_i を取り出して W_i' を作る．
   - w_i の取り得る語義 t_{w_i1},...,t_{w_il} をそれぞれ T に加えて $T_1',...,T_l'$ を作る．
   - 新しいノード $[W_i',T_0']$, ...,$[W_i',T_l']$ を作り，キュー Q' に加える．
3. キュー Q' 内のノードをスコア $s(N)$ の高い順にソートする
4. キュー Q' の先頭ノードの W が空であれば，終了する (T が求める語義タグ)．そうでなければ，Q' の中からスコア順で上位 b のノードのみを残してこれを新しいキュー Q とし，2 に戻る．

6 評価実験と考察

檜コーパスを訓練データとして語義選択モデルも構築し，語義タギングの精度を評価した．手法そのものを評価するため，テストデータは，実際の解析器の出力ではなく，檜コーパス自身を使用した．実際に語義タギングの対象となるのは Lexeed で多義語と定義されている約 9,000 語である．Lexeed の定義文 (LXD-DEF)，例文 (LXD-EX)，定義文+例文 (LXD-ALL) のそれぞれを訓練データに用い,，最大エントロピモデルとして語義選択モデルを作り実験を行った．語義タグは定義文 75,000 文，例文 46,000 文全文に付与されているが，ツリーバンクが存在する，すなわち意味構造素性が使えるのは定義文の約 54,000 文，例文の約 36,000 文である．モデルの構築には，オープンソースのツールキット Maximun Entropy Modeling Toolkit [3]を使用した．テストデータとして，Lexeed の定義文，例文に京大コーパスを加えたコーパスの中からそれぞれ 1,000 文ずつを抜き出

[2] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

[3] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Corpus (test / train)	Baseline	Surface	+SEM-Col	+SEM-Dep	+Domain	FULL
$\text{LXD-DEF}_{\text{test}}$ / LXD-EX	66.2	70.1	71.6	70.2	70.3	71.9
$\text{LXD-EX}_{\text{test}}$ / LXD-DEF	60.5	64.2	64.8	64.5	64.0	65.0
$\text{KYOTO}_{\text{test}}$ / LXD-EX	55.4	59.7	59.8	59.2	60.0	60.2
$\text{KYOTO}_{\text{test}}$ / LXD-DEF	55.0	57.5	57.3	57.6	57.8	58.0
$\text{KYOTO}_{\text{test}}$ / LXD-ALL	57.1	60.2	60.9	60.8	61.2	61.3

表 3: 語義選択の精度

したものを使用した ($\text{LXD-DEF}_{\text{test}}$, $\text{LXD-EX}_{\text{test}}$, $\text{KYOTO}_{\text{test}}$). 一文に含まれる平均の多義語は 3.4 語 ($\text{LXD-EX}_{\text{test}}$) から 5.2 語 ($\text{KYOTO}_{\text{test}}$) である．それぞれの訓練データに対して，単語共起のみ (Surface) を素性として使用したもの，これを基本素性として，語義共起 (+SEM-Col)，意味構造素性 (+SEM-Dep)，ドメイン素性 (+Domain) をそれぞれを加えた素性を使用したもの，全素性を使用したもの (FULL) の計 5 種類のモデルを構築した．また，訓練コーパス中で最も出現する頻度の高い語義を選択する方法をベースラインとした．

結果を表 3 に示す．いずれの場合もベースラインと全素性使用のモデルとの精度の差は有意である結果が得られている．一方で，単語表層のみのモデルと素性使用のモデルとの差は限定的であった．品詞別で見ると，語義共起は名詞，動詞に効果があったが，形容詞，副詞にはあまり効果がなかった．意味構造素性は，今回の結果では全体に効かなかったが，内容を見ると，効果のあった事例に対してと同程度悪影響のあった事例があるために相殺している傾向があり，意味構造に対する語義拡張は工夫する余地がある．ドメイン素性も現状では効果が弱いが，これは一文内のみの情報を用いたことと，モデリングを辞書文のみで行ったことの影響が出ていると考えられる．

精度を，人間がアノテートした場合の作業者内一致度と比較すると 10%強ぐらい差があるが，コーパス間の傾向は，人間の作業の場合の難易度の傾向と一致している．また，[8] でも述べられているように，Lexeed の語義の分け方は国語辞典を元にしており必ずしもこのままの分解能が機械処理に向いているとは限らない．人間のアノテーションの一致度が低い語義，例えば，一つの単語に属する語義で日本語語彙大系の意味属性が同一になるような語義[4]は，まとめて語義の粒度を粗くしても十分なタスクが多いと考えている．訓練データ LXD-EX －テストデータ $\text{LXD-DEF}_{\text{test}}$ の場合，同一意味属性を持つ語義をまとめると 71.9%から 81.2%，第 4 階層の意味属性が同一になる語義でまとめると，88.8%の精度になる．

7 おわりに

語彙的意味情報，構造的意味情報を用いて，単語の語義を選択する語義タギングの方法について述べた．檜コーパスを使用して評価実験を行った結果，全ての素性を用いることによりベースラインに比較して一定の効果があった．素性選択とモデリングを改善するとともに，実際の形態素解析器，パーザを使用しての実験を行う予定である．

[4]「競技に出るために選ばれた**選手** $_1$」と「職業としてスポーツを行う**選手** $_2$」の語義の区別など．

参考文献

[1] Christine Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[2] Palmer, Martha, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, Vol. 31, No. 1, pp. 71–106, March 2005.

[3] Francis Bond, Sanae Fujita, and Takaaki Tanaka. The Hinoki syntactic and semantic treebank of Japanese. *Language Resources and Evaluation*, 2007. (Special issue on Asian language technology).

[4] Kaname Kasahara, Hiroshi Sato, Francis Bond, Takaaki Tanaka, Sanae Fujita, Tomoko Kanasugi, and Shigeaki Amano. Construction of a Japanese semantic lexicon: Lexeed. SIG NLC-159, IPSJ, Tokyo, 2004. (in Japanese).

[5] Eric Nichols and Francis Bond. Acquiring ontologies using deep and shallow processing. In *11th Annual Meeting of the Association for Natural Language Processing*, pp. 494–498, Takamatsu, 2005.

[6] Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. *Goi-Taikei — A Japanese Lexicon*. Iwanami Shoten, Tokyo, 1997. 5 volumes/CDROM.

[7] Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus — while improving the parsing system. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 14, pp. 249–260. Kluwer Academic Publishers, 2003.

[8] Takaaki Tanaka, Francis Bond, and Sanae Fujita. The Hinoki sensebank — a large-scale word sense tagged corpus of Japanese —. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pp. 62–69, Sydney, 2006. (ACL Workshop).

[9] Carl Pollard and Ivan A. Sag. *Head Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.

[10] Melanie Siegel and Emily M. Bender. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*, Taipei, 2002.

[11] Ann Copestake, Daniel Flickinger, Carl Pollard, and Ivan A. Sag. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 2005. in press.

[12] 橋本力, 黒橋禎夫. 基本語ドメイン情報の構築. 第 13 回言語処理学会年次大会, 2007. (to appear).