

# Abstract Meaning Representation を用いた 名詞句の意味構造解析

澤井 裕一郎<sup>1,a)</sup> 進藤 裕之<sup>1,b)</sup> 松本 裕治<sup>1,c)</sup>

概要：本研究では、Abstract Meaning Representation を用いた名詞句の意味構造解析手法を提案する。Abstract Meaning Representation は、有向非巡回グラフによる文の意味表現であり、名詞句の内部構造や述語項関係などの意味関係がアノテートされている。我々は、Abstract Meaning Representation コーパスから名詞句に対応する部分を抽出し、名詞句の意味構造データを構築した。また、意味構造木のノードと意味関係の同定を同時に行う遷移型解析アルゴリズムを提案し、実験により意味構造解析の性能が向上することを示す。

## 1. はじめに

名詞句の意味・構造解析は、自然言語処理における重要な研究対象であり、機械翻訳や質問応答システムの高精度化に寄与することが知られている [11], [12]。名詞句の意味・構造解析に関する既存研究としては、名詞句の内部構造解析 [16], [17]、複合名詞の意味関係解析 [5], [8], [15] や述語項関係解析 [10] などが挙げられる。

名詞句の内部構造解析では、例えば“lung cancer deaths”という英語の名詞句に対して、“(lung cancer) deaths”というブラケット情報を自動付与することが目的である。上記の例では、“lung cancer”が一つの内部的な句を構成し、名詞句全体の主辞が“deaths”であることを表している。複合名詞の意味関係解析では、主に2単語で構成される複合名詞に対して、あらかじめ定義された意味関係を表すラベルを自動付与することが目的である。例えば、“cooking pot”という複合名詞は、「料理用の鍋」という意味で用いられるので、“purpose”という意味関係ラベルを付与すれば正解となる。一方、複合名詞の述語項関係解析では、例えば“car lover”という複合名詞に対して、“car”が“love”の主語なのか目的語なのかを推定する。

以上のように、これまでの既存研究の多くは、あらゆる名詞句を対象とするのではなく、細分化された各タスクごとに個別のデータセットを構築し、解析手法の評価を行ってきた。しかしながら、実際にテキスト中に現れる名

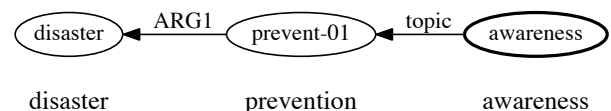


図1 “disaster prevention awareness” の AMR による意味構造

詞句は、ブラケット構造や述語項関係などが複合的に組み合わせられて構成されている。例えば、“disaster prevention awareness”は、“prevention”と“disaster”の間に述語項の関係があり、さらに“(disaster prevention) awareness”という内部構造を有すると考えられる。このような複合的な名詞句の意味構造解析を行うために、本研究では、Abstract Meaning Representation (AMR) コーパスを利用する。

AMR は、有向非巡回グラフ構造の文の意味構造形式であり、語の形態的な情報や文の統語的な情報を省略し、主に文の述語項構造を表す。1 万文以上の英語テキストに対して、人手によって AMR の情報が付与されたコーパスが AMR Bank<sup>\*1</sup> として公開されている。本研究では、この AMR コーパスから、名詞句に対応する部分構造を抽出することによって、名詞句の意味構造データを構築する。抽出された名詞句と、対応する AMR の例を図1に示す。ただし、名詞句の内部では、複数の述語による項の共有が起きないので、名詞句に対応する AMR の部分構造は、グラフ構造ではなく木構造であると仮定できる。したがって、本研究で扱う名詞句の意味構造解析は、名詞句を構成する単語列から、AMR の木構造を推定するというタスクになる。

従来の AMR 解析手法では、まず、意味構造グラフのノード (コンセプト) の同定を行い、次に、コンセプト間の依存

<sup>\*1</sup> <http://amr.isi.edu/>

<sup>1</sup> 奈良先端科学技術大学院大学  
Nara Institute of Science and Technology  
a) sawai.yuichiro.sn0@is.naist.jp  
b) shindo@is.naist.jp  
c) matsu@is.naist.jp

表 1 作成した名詞句 AMR データの統計量

訓練データ	開発データ	テストデータ
3504	463	398

構造を推定するという二段階のアプローチをとる [4]。コンセプトの同定は、従来の品詞タグ付けとは異なり、一つの単語に対して必ず一つのコンセプトが割り当てられるとは限らないことや、コンセプトの種類が品詞タグの種類よりも遥かに多い等の理由から、同定精度は 67.5%程度にとどまっている。したがって、従来研究のようなパイプライン処理に基づく手法では、コンセプト同定の誤りが依存構造解析へ伝搬してしまい、全体の解析精度が極めて低いという問題がある。

上記の問題点を解決するために、本研究では、単語の依存構造解析で用いられてきた遷移型の解析手法を拡張し、AMR のコンセプト同定と、コンセプト間の依存構造解析とを同時に行う新たな手法を提案する。実験により、提案手法は、名詞句の AMR 解析において、既存手法を 2.7 ポイント上回ることを確認した。また、外部資源として単語の動詞化・名詞化の辞書を使うことで、既存手法を 9.8 ポイント上回ることを確認した。

## 2. Abstract Meaning Representation

### 2.1 名詞句の抽出

まず、我々は AMR Bank から名詞句に対応する部分構造を抽出した。AMR では、文の各単語と、意味構造のノードであるコンセプトとの対応関係（アライメント）の注釈がない。そこで、我々は Flanigan ら [4] によるルールベースのアライメントツールを利用し、単語とコンセプトとのアライメント情報を得た。次に、Stanford Parser [9] を用いて各文の構造解析を行い、名詞を 2 つ以上含む名詞句 (NP) のうち、他の名詞句には含まれないものを抽出した。ただし、固有表現を含む名詞句は、本研究の対象ではないため抽出対象から除いた。また、所有代名詞および接続詞を含む名詞句は、ルールベースのアライメントが不正確である傾向があったため、抽出対象から除いた。表 1 に、最終的に作成された名詞句 AMR データの統計量を示す。

### 2.2 既存の解析手法

名詞句の AMR を用いた意味解析のベースラインとして、Flanigan ら [4] の手法を用いる。この手法では、まず、入力文の単語分割と、各単語に割り当てるコンセプトの同定を系列ラベリング問題 [7] として定式化し、動的計画法 (Viterbi アルゴリズム) で推論を行う。次に、最大全域グラフを求めるアルゴリズム [2] を用いて、コンセプト間の意味構造（依存構造）、および意味関係ラベルを推定する。前述のように、この手法では、コンセプトの同定と意味構造の推定を別々に行うため、コンセプト同定の誤りが意味

表 2 操作と状態遷移の定義

操作	現在の状態	次の状態
Shift( $c(w_i)$ )	$(\sigma, [w_i \beta], R)$	$([\sigma c(w_i)], \beta, R)$
Left-Reduce( $r, n$ )	$([\sigma c_i c_j], \beta, R)$	$([\sigma c_j], \beta, R \cup \{n_{\text{root}}(c_i) \xleftarrow{r} n(c_j)\})$
Right-Reduce( $r, n$ )	$([\sigma c_i c_j], \beta, R)$	$([\sigma c_i], \beta, R \cup \{n(c_i) \xrightarrow{r} n_{\text{root}}(c_j)\})$
Empty-Reduce	$([\sigma \phi], \beta, R)$	$(\sigma, \beta, R)$

構造の推定へ伝播するという問題がある。

## 3. 提案手法

本研究では、名詞句を構成する単語列から、AMR の木構造を推定するタスクに取り組む。名詞句の AMR 解析では、各単語に対応するコンセプトを同定し、さらにコンセプト間の依存関係を推定する必要がある。ただし、コンセプトの同定は従来の品詞タグ付けとは異なり、一つの単語に対して必ず一つのコンセプトを割り当てれば良いとは限らない。そこで、Nivre の arc-standard 法 [13] を拡張し、コンセプトの同定と意味関係の同定を同時に行う遷移型アルゴリズムを提案する。このアルゴリズムでは、入力された単語列をバッファに、同定済みのコンセプト部分木列をスタックに保持する。そして、Shift 操作で、バッファの先頭の単語に対応するコンセプト部分木の同定とバッファへの追加を行い、Reduce 操作で、バッファ上のコンセプト部分木間の意味関係を同定する。例として、図 2 に、名詞句 “the retired plant worker” に対する意味構造を導出する際の状態遷移を、図 3 に、最終的に導出される意味構造を示す。

表 2 に、遷移システムの各操作と状態遷移の定義を示す。遷移システムの状態は、コンセプト部分木を要素に持つスタック  $\sigma$ 、単語を要素に持つバッファ  $\beta$ 、コンセプト間の意味関係の集合  $R$  の 3 つ組  $(\sigma, \beta, R)$  である。Shift( $c(w_i)$ ) 操作は、バッファの先頭にある単語  $w_i$  を取り出し、 $w_i$  から生成されるコンセプト部分木  $c(w_i)$  をスタックにプッシュする。コンセプト部分木  $c(w_i)$  は、表 3 のルールのいずれかにより単語  $w_i$  から生成される。Left-Reduce( $r, n$ ) 操作は、スタックの先頭 2 つの部分木  $c_i, c_j$  をポップし、部分木  $c_i$  の根ノード  $n_{\text{root}}(c_i)$  から部分木  $c_j$  内のノード  $n(c_j)$  への意味関係  $r$  を意味関係の集合  $R$  に追加し、部分木  $c_j$  をスタックの先頭に再びプッシュする。ここで、 $n$  は部分木から部分木内の特定のノードへの写像である。今回作成したデータ中に出現するコンセプト部分木の大きさは高々 2 であったため、 $n \in \{n_{\text{root}}, n_{\text{child}}\}$  である。ただし、 $n_{\text{root}}$  は部分木から根ノードへの写像、 $n_{\text{child}}$  は部分木から根ノードの直接の子への写像である。Right-Reduce( $r, n$ ) 操作も同様に定義される。Empty-Reduce 操作は空の部分木  $\emptyset$  をスタックの先頭から取り除く。Shift( $\emptyset$ ) 操作で空の部分

表 3 コンセプト部分木の生成ルール (生成例中の縦棒は複数の生成候補があることを示す。)

ルール名	作用	単語 “fighters” への適用例
EMPTY	空のコンセプト部分木 $\emptyset$ を生成。	fighters $\rightarrow \emptyset$
KNOWN	訓練データ中で対応があるコンセプト部分木を生成。	fighters $\rightarrow$ (person) $\xrightarrow{\text{ARG0-of}}$ (fight-01)   ...
LEMMA	単語の見出し語形を, 唯一のコンセプトに持つ部分木を生成。	fighters $\rightarrow$ (fighter)
DICT_PRED	派生元の動詞と関連する述語を, 唯一のコンセプトに持つ部分木を生成。	fighters $\rightarrow$ (fight-01)   (fight-02)   ...
DICT_NOUN	派生元の名詞を, 唯一のコンセプトに持つ部分木を生成。	fighters $\rightarrow$ (fight)

	直前の操作	$\sigma_1$	$\sigma_0$	$\beta$	R
0	(初期状態)			[ a retired plant worker ]	$\emptyset$
1	Shift(EMPTY(a))		$\emptyset$	[ retired plant worker ]	$\emptyset$
2	Empty-Reduce			[ retired plant worker ]	$\emptyset$
3	Shift(DICT_PRED(retired))			[ plant worker ]	$\emptyset$
4	Shift(LEMMA(plant))			[ worker ]	$\emptyset$
5	Shift(KNOWN(worker))		 ↓ ARG0-of 	[ ]	$\emptyset$
6	Left-Reduce(ARG2, $n_{\text{child}}$ )		 ↓ ARG0-of 	[ ]	$\{(work-01) \xrightarrow{\text{ARG2}} (plant)\}$
7	Left-Reduce(ARG0-of, $n_{\text{root}}$ )		 ↓ ARG0-of 	[ ]	$\{(work-01) \xrightarrow{\text{ARG2}} (plant), (person) \xrightarrow{\text{ARG0-of}} (retire-01)\}$

図 2 “a retired plant worker” の意味構造の導出例 ( $\sigma_0, \sigma_1$  はそれぞれスタックの先頭, 先頭から 2 番目の部分木を表す)

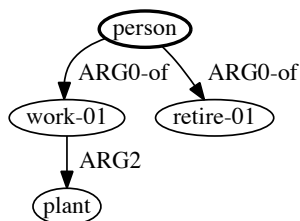


図 3 名詞句 “the retired plant worker” の導出された意味構造

木  $\emptyset$  を生成した直後には Empty-Reduce 操作を必ず行う。初期状態では, スタック  $\sigma$  は空で, バッファ  $\beta$  に名詞句の全ての単語があり, 意味関係の集合  $R$  は空である。終了状態は, バッファ  $\beta$  が空で, スタック  $\sigma$  に唯一の部分木がある状態である。状態遷移の過程で Shift 操作によって生成された全ての部分木を, 終了状態での  $R$  に含まれる意味関係で結んだものが, 最終的に導出される名詞句の意味構造である。

従来の AMR 解析手法では, 訓練データに出現しなかった単語に対して, 対応するコンセプト部分木を生成することができない。その結果, コンセプトの同定精度が低いという問題があった。一方, 我々の提案手法では, 表 3 に示

すように, 単語からコンセプトを生成する 5 つの汎用的なコンセプト生成ルールを定義し, Shift 操作時にいずれかの生成ルールを適用することによって, 未知の単語に対しても何らかのコンセプト部分木を割り当てることができる。

### 3.1 素性

現在の状態  $s$  と次の操作  $a$  に対して定義される素性集合  $\phi(s, a)$  を, 状態の素性集合  $\phi_{\text{state}}(s)$  と操作の素性集合  $\phi_{\text{action}}(s, a)$  の直積 (2 つの集合の要素の総当たりの組み合わせ) とする。

$$\phi(s, a) = \phi_{\text{state}}(s) \times \phi_{\text{action}}(s, a)$$

状態の素性集合  $\phi_{\text{state}}(s)$  は表 4 で定義される素性集合の和集合である。ただし,  $w(c)$  は部分木  $c$  の生成元の単語,  $\text{lem}$  は見出し語,  $\text{suf}$  は長さ 3 の接尾辞,  $\text{pos}$  は品詞,  $\text{rel}$  は依存構造木での親単語への依存関係ラベル,  $\text{off}$  は依存構造木での親単語への相対位置,  $\circ$  は素性の連結を表す。

表 5 に各操作  $a$  に対する素性集合  $\phi_{\text{action}}((\sigma, [w_i|\beta], R), a)$  を示す。ただし,  $\text{rule}(w_i, c)$  は, 状態  $(\sigma, [w_i|\beta], R)$  において, バッファの先頭の単語  $w_i$  から部分木  $c$  を生成する際

表 4 状態の素性集合

名前	定義
LEMMA	$\{w(\sigma_1).lem, w(\sigma_0).lem, \beta_0.lem, w(\sigma_1).lem \circ w(\sigma_0).lem, w(\sigma_0).lem \circ \beta_0.lem\}$
SUFFIX	$\{w(\sigma_1).suf, w(\sigma_0).suf, \beta_0.suf, w(\sigma_1).suf \circ w(\sigma_0).suf, w(\sigma_0).suf \circ \beta_0.suf\}$
POS	$\{w(\sigma_1).pos, w(\sigma_0).pos, \beta_0.pos, w(\sigma_1).pos \circ w(\sigma_0).pos, w(\sigma_0).pos \circ \beta_0.pos\}$
DEP_REL	$\{w(\sigma_1).rel, w(\sigma_0).rel, \beta_0.rel, w(\sigma_1).rel \circ w(\sigma_0).rel, w(\sigma_0).rel \circ \beta_0.rel\}$
DEP_HEAD	$\{w(\sigma_1).off, w(\sigma_0).off, \beta_0.off, w(\sigma_1).off \circ w(\sigma_0).off, w(\sigma_0).off \circ \beta_0.off\}$
ROOT	$\{n_{root}(\sigma_1), n_{root}(\sigma_0), n_{root}(\sigma_1) \circ n_{root}(\sigma_0)\}$
BETWEEN	$w(\sigma_1)$ と $w(\sigma_0)$ の間の全ての単語 $\cup$ $w(\sigma_0)$ と $\beta_0$ の間の全ての単語

表 5 操作の素性集合

操作 $a$	$\phi_{action}((\sigma, [w_i \beta], R), a)$
Shift( $c$ )	$\{“S”, “S” \circ rule(w_i, c), “S” \circ rule(w_i, c) \circ c\}$
Left-Reduce( $r, n$ )	$\{“L-R”, “L-R” \circ r, “L-R” \circ r \circ n\}$
Right-Reduce( $r, n$ )	$\{“R-R”, “R-R” \circ r, “R-R” \circ r \circ n\}$
Empty-Reduce	$\{“E-R”\}$

に適用したルール名を返す関数である。生成する部分木が異なる Shift 操作，意味関係が異なる Left/Right-Reduce 操作間で素性を共有するために，各操作ごとに段階的な粒度の素性を定義している。例えば，Shift( (run-01) ) と Shift( (sleep-01) ) は異なる操作であるが，適用した生成ルール DICT\_PRED が共通しているため，“S”，“S” $\circ$ “DICT\_PRED”という素性を共有する。

#### 4. 評価実験

今回作成した名詞句の意味構造データ (表 1) を用いて，提案手法の評価実験を行う。Flanigan らによる解析手法の実装<sup>\*2</sup>をベースラインとする。ベースラインでのコンセプト同定時の素性は，訓練データ中の単語とコンセプトの対応頻度，意味関係同定時の素性は，品詞タグ付けや依存構造解析の結果等を使用する。これらの素性はデフォルト設定のものである。

Flanigan らの手法では，訓練データ中に出現するコンセプトのみを生成する。提案手法では，表 3 に示すコンセプト生成ルールのうち，LEMMA, DICT\_PRED, DICT\_NOUN を使用することで，訓練データ中に出現していないコンセプトを生成可能である。したがって，公正な比較のために，コンセプト生成ルールとして EMPTY, KNOWN のみを使用する (表 6, 7 の提案手法 (EMPTY/KNOWN))。次に，コンセプト生成ルールとして単語の見出し語形のコンセプトを生成する LEMMA を追加する (表 6, 7 の提案手法+LEMMA)。提案手法+LEMMA では訓練デー

<sup>\*2</sup> <https://github.com/jflanigan/jamr>

表 6 意味構造解析の性能比較

手法	適合率	再現率	F 値
(Flanigan et al., 2014)	75.5	61.1	67.5
提案手法 (EMPTY/KNOWN)	78.0	63.8	70.2
提案手法+LEMMA	75.7	75.2	75.4
提案手法+LEMMA/DICT	77.3	77.3	77.3

表 7 コンセプト同定の性能比較

手法	適合率	再現率	F 値
(Flanigan et al., 2014)	88.4	71.4	79.0
提案手法 (EMPTY/KNOWN)	88.9	72.2	79.7
提案手法+LEMMA	84.8	84.2	84.5
提案手法+LEMMA/DICT	85.8	85.6	85.7

タ以外の外部資源が必要ないことに注意されたい。そして，提案手法+LEMMA に加えて，コンセプト生成ルールとして DICT\_PRED と DICT\_NOUN を追加する (表 6, 7 の提案手法+LEMMA/DICT)。コンセプト生成ルール DICT\_PRED, DICT\_NOUN では名詞，形容詞から派生元の動詞や名詞への変換が必要である。この変換のための辞書として CatVar を使用した [3]。また，AMR で利用される PropBank [14] の述語の定義データを外部資源として利用し，PropBank で定義が存在する述語に関しては，定義されている意味関係のみを持つことができるという制約を課す。

訓練時には，ビーム探索 (ビーム幅 8) を行う max-violation パーセプトロン [6] を使用し，さらにパラメータの平均化を行った。テスト時にもビーム探索 (ビーム幅 8) を行った。

表 6 に，意味構造解析の性能の比較を示す。評価指標として，意味構造グラフのノード，エッジ，ルートの一致度を適合率，再現率，F 値で評価する Smatch スコア [1] を用いる。提案手法では，Flanigan らによる手法に比べて，適合率と再現率がともに向上し，F 値では 2.7 ポイント向上した。提案手法+LEMMA では，コンセプト生成ルール LEMMA を加えることで，訓練データに出現しないコンセプトを生成できるようになるため，再現率が大幅に向上し，結果として F 値が 5.2 ポイント向上した。提案手法+LEMMA/DICT では，外部資源である辞書と PropBank の述語の定義情報を加える事により，F 値が更に 1.9 ポイント向上した。

また，表 7 に，コンセプト同定のみでの性能の比較を示す。正解のコンセプト集合に対する適合率，再現率，F 値を報告する。各条件で，表 6 と同様の傾向が見られた。したがって，提案手法により，コンセプトの同定と意味関係の同定の両方における解析性能が向上することが示された。

#### 5. おわりに

本研究では，Abstract Meaning Representation (AMR) を名詞句の解析に用いた。AMR Bank から名詞句に対応する部分構造を抽出し，名詞句の意味構造データを構築し

た。そして、意味構造木のノード(コンセプト)の同定とコンセプト間の依存構造の推定を同時に行う手法を提案し、解析性能が向上することを実験により確認した。また、外部辞書の情報を加える事で更に解析性能が向上することを確認した。

#### 参考文献

- [1] Cai, S. and Knight, K.: Smatch: An evaluation metric for semantic feature structures, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 748–752 (2013).
- [2] Chu, Y. and Liu, T.: On the shortest arborescence of a directed graph, *Science Sinica*, Vol. 14, pp. 1396–1400 (1965).
- [3] Dorr, B., Park, C. and Park, C.: A categorial variation database for English, *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 17–23 (2003).
- [4] Flanigan, J., Thomson, S., Carbonell, J., Dyer, C. and Smith, N. A.: A discriminative graph-based parser for the Abstract Meaning Representation, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1426–1436 (2014).
- [5] Girju, R., Moldovan, D., Tatu, M. and Antohe, D.: On the semantics of noun compounds, *Computer Speech and Language*, Vol. 19, pp. 479–496 (2005).
- [6] Huang, L., Fayong, S. and Guo, Y.: Structured perceptron with inexact search, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–151 (2012).
- [7] Janssen, J. and Linnios, N.: *Semi-Markov models and applications*, Springer (1999).
- [8] Kim, S. N. and Baldwin, T.: A lexical semantic approach to interpreting and bracketing English noun compounds, *Natural Language Engineering*, Vol. 19, No. 1, pp. 385–407 (2013).
- [9] Klein, D. and Manning, C.: Accurate unlexicalized parsing, *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pp. 423–430 (2003).
- [10] Lapata, M.: The disambiguation of nominalizations, *Computational Linguistics*, Vol. 28, No. 3, pp. 357–388 (2002).
- [11] Nakov, P.: On the interpretation of noun compounds: Syntax, semantics, and entailment, *Natural Language Engineering*, Vol. 19, No. 1, pp. 291–330 (2013).
- [12] Nakov, P. I. and Hearst, M. A.: Semantic interpretation of noun compounds using verbal and other paraphrases, *ACM Transactions on Speech and Language Processing*, Vol. 10, No. 3, pp. 1–51 (2013).
- [13] Nivre, J.: Incrementality in deterministic dependency parsing, *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pp. 50–57 (2004).
- [14] Palmer, M., Gildea, D. and Kingsbury, P.: The Proposition Bank: An annotated corpus of semantic roles, *Computational Linguistics*, Vol. 31, No. 1, pp. 71–106 (2005).
- [15] Tratz, S. and Hovy, E.: A taxonomy, dataset, and classifier for automatic noun compound interpretation, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 678–687 (2010).
- [16] Vadas, D. and Curran, J. R.: Adding noun phrase structure to the Penn Treebank, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 240–247 (2007).
- [17] Vadas, D. and Curran, J. R.: Parsing noun phrase structure with CCG, *Proceedings of ACL-08: HLT*, pp. 335–343 (2008).