自然言語処理 141－4
（2001．1．25）

単語間二項関係の制約を使用したGDA-UNL変換システム

白松　俊[†]　伊藤誠悟[†]
橋田浩一[††]　奥乃　博[†]

GDA (Global Document Annotation) タグ付き日本語文から UNL (Universal Network Language) 文への自動変換システムでは，GDA タグ付き文から単語間二項関係を抽出し，UW (Universal Word) 間二項関係への変換を行う．その際，変換候補が複数生じるため，曖昧性解消が必要となる．そのために，品詞情報・共起情報・上位下位関係を制約として候補の絞り込みを行い，シソーラス上での単語間距離を用いて各単語間二項関係のスコアリングを行う．本稿では，候補の絞り込みとスコアリングを行うアルゴリズムを設計したので，報告する．

GDA-UNL Conversion System Using Binary Relation between Words as Constraint

Shun Shiramatsu ,[†] Seigo Ito ,[†] Koiti Hasida [††]
and Hiroshi G. Okuno[†]

The automatic conversion system from a Japanese sentence with the GDA (Global Document Annotation) tag to the UNL (Universal Network Language) sentence extracts the binary relation between words from the sentence with the GDA tag, and converts to binary relation between UW (Universal Word). On binary relation conversion, resolving ambiguity is needed. Then, we research the way to reslove ambiguity by part of speech, co-occurence, and broader or narrower terms. In this paper, we design algorithm for resolving ambiguity and scoring on GDA-UNL binary relation conversion.

1. はじめに

1.1 研究の背景

21 世紀に入り，コンテンツの充実がますます重要となっていくと期待されているが，これは新規コンテンツの開発だけを意味するのではない．従来から電子化されていたデータの様々なアプリケーション間で共用するという要求がますます増大すると予想される．例えば，特定のアプリケーション用に作成されたデータを他のデータに変換する．生データから情報を抽出し，アプリケーションで使いやすいような 2 次情報を作成する．複数のアプリケーションで共用できるデータ表現法を規定し，それに基づいてアプリケーションを開発する．等々．

そのような共通データ表現法の一つとして，電子技術総合研究所の橋田は，XML の拡張である GDA (Global Document Annotation)[1] を提案し，GDA では，統語・意味情報を付加することによって，要約，翻訳，検索，データ変換などの作業を軽減することを狙っている．また，国際連合大学高等研究所の内田らは，意味ネットワーク構造をした人工言語 U NL (Universal Networking Language)[2] を提案し，多国語間での翻訳や検索システム開発の促進を狙っている．実際，UNL から世界 18 カ国語との翻訳システムが現在開発中である．

このような 2 つの知識・データ表現法の間での変換を自動化することは，(1) 既存資産の有効利用，(2) 各システムでゼロからデータを作るコストの軽減，などに役立つと期待される．とくに，翻訳用の中間言語としても使われる UNL では，UNL から目標言語への変換 (DeConverter と呼ばれる) はそれほど難しくもないが，その逆の言語から UNL への変換 (EnConverter と呼ばれる) は，言語理解の課程が必要なために，自動化が難しいと考えられる．

我々は，このような問題点を 2 つの知識・データ表

† 東京理科大学 理工学研究科 情報科学専攻
Department o f Information Sciences, Science University of Tokyo
†† 電子技術総合研究所 情報科学研究部
Information Science Division, Electrotechnical Laboratory

現法間の変換で解決する方法を研究している，つまり，人手で付けられた GDA タグつきの文書を自動的に UNL 2 項関係に変換する．このような自動変換により，UNL コミュニティは GDA から変換された数多くの UNL 文書を手に入れることができ，逆に GDA コミュニティは，作成した GDA 文書が少なくとも世界 18 カ国語への自動翻訳のサービスが享受できるようになる．

1.2 GDA

GDA[1] は，文書に対する統語・意味的アノテーションを目的として設計された XML タグセットである．GDA タグは，品詞，係り受け，単語間二項関係，照応関係，多義語の意味などの情報を含む．

- 品詞
  品詞タグ `n`, `np`, `v`, `vp`, `aj`, `ajp`, `ad`, `adp` で表される．
- 係り受け（依存構造）
  エレメントは高々1 つの兄弟エレメントに係る．係り先の方向は属性 `syn` で表される．
- 単語間二項関係
  二項関係ラベル `agt`, `pat`, ･･･ で表される．次節に詳しく述べる．

図 1 に，GDA タグ付き文の例を示す．

本稿では GDA タグ付き日本語文から UNL への変換について述べるが，その以前に日本語の平文に GDA タグを付加する必要があり，以下のような処理になる．

(1) 日本語パーザを用い，統語構造・依存関係を表す GDA タグを付加
(2) 動詞の格解析を行い，単語間二項関係を表す GDA タグを付加
(3) 照応関係を表す GDA タグを付加

上記の各フェーズを自動的に行うシステムの出力には誤りがあると予想されるため，用途によっては人手によるタグ修正が必要となる．

また，自動タグ付けの誤りを軽減するため，あらかじめ人手によって簡単なタグ付けがされた文を入力としてタグの詳細化を行うシステムの研究も我々の研究室で行っている．

本システムでは，次のような条件を満たす GDA タグ付き文を入力として想定している．

- 最大投射☆にはタグが付いている
- 二項関係ラベルで単語間二項関係が明示されている

☆ 主辞でない構成素（隣接する他の構成素を受けていない構成素）を最大投射 (maximal projection) または 句 (phrase) と言う．

```
<su syn="fc"> ･･･ 依存関係（前向き連鎖）
  <adp opr="agt"> ･･･ 二項関係（主体）
    <np><ajp>若い</ajp><n> 男</n></np>
    <ad>が</ad>          品詞（名詞タグ）
  </adp>
  <adp opr="pat">
    <np>タコ</np>
    <ad>を</ad>
  </adp>
  <v><vp>食べて</vp><v>いた</v></v>。
</su>
```

図 1 GDA タグ付き文
「若い男がタコを食べていた。」

```
[S]
{org:jp}
若い男がタコを食べていた。
{/org}
{unl}
[W]
young(aoj>thing):00
man(icl>person):01
```

```
octopas(icl>food):02
eat(icl>do).@past:03
[/W]      UW
[R]
00aoj01 ･･･ 二項関係
03agt01
03obj02
[/R]
{/unl}
[/S]
```

図 2 UNL 式
「若い男がタコを食べていた。」

図 1 に GDA タグ付き文の例を示す．

1.3 UNL

UNL[2] は，意味ネットワークの構造を持つ人工言語である．UNL を構成する Universal Word，二項関係，属性，UNL 知識ベースについて説明する．

- UW (Universal Word)
  概念を表し，文字列 (head word) と制約リストから成る．意味ネットワークのノードにあたる．
- 属性
  時制やアスペクトなど
- 二項関係
  UW 間二項関係は二項関係ラベルによって表される．意味ネットワークのリンクにあたる．
- UNL 知識ベース
  可能性のある UW 間二項関係とその共起頻度

図 2 に，UNL 式の例を示す．

2. GDA から UNL への変換処理

2.1 変換処理の概要

GDA から UNL への変換処理の流れは以下のようになる．

(1) タグで分割されていない形態素に品詞タグを付ける．
(2) GDA タグ付き文から二項関係を抽出する．

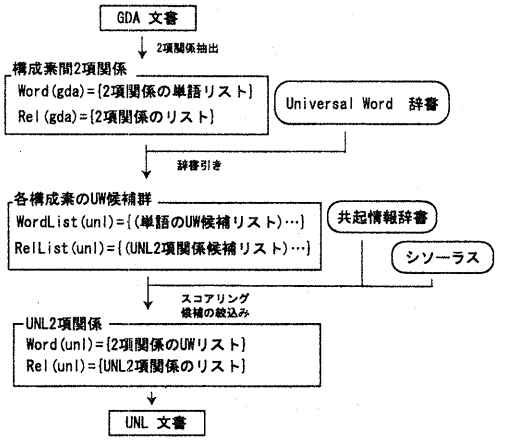

図 3 サブシステム間の中間データ

(3) 二項関係を変換する．つまり，GDA 関係ラベルを UNL 関係ラベルに，日本語単語を UW に変換する．

(i) 辞書引き

(ii) 共起情報の制約を用いた候補の絞り込み

(iii) シソーラスを用いたスコアリング

(iv) 候補からの選択

(4) UNL 形式にフォーマットする

2.2 変換処理における問題点

形態素レベルまでタグがついていない GDA の場合，二項関係抽出のために形態素解析が必要．

また，日本語から UW, GDA 関係ラベルから UNL 関係ラベルへの変換の際に，候補が一意に定まらず複数の概念候補が出現するため，曖昧性解消が必要である．

2.3 形態素解析

形態素レベルまでタグが付いていない場合にも対応できるように，ChaSen[5] を用いて形態素解析を行う．ChaSen でタグを記号として扱うように設定できるため，タグを取り除くデタグ処理を行う必要はない．

また，ChaSen で動詞の語幹と基本形も取得できる．この語幹処理は，後の処理で辞書引きを行うために必要である．

2.4 GDA 文からの二項関係の抽出

GDA 文から単語間二項関係を抽出するサブシステムは，次のような GDA タグ付き文を入力とする．

- 形態素レベルまでタグが付いている
- 二項関係ラベルで単語間二項関係が明示されている

また，入力データから単語間二項関係を抽出し，次のような中間データ表現として出力する．

文中の日本語単語リスト
$W^{gda} = \{\ jw_i\ ;\ i = 1, \cdots, n\ \}$

文中の単語間二項関係リスト
$R^{gda} = \{\ (jw^1, gdaLabel_j, jw^2)\ ;$
$jw^1, jw^2 \in W^{gda},\quad j = 1, \cdots, m,$
$gdaLabel_j$は GDA 二項関係ラベル $\}$

2.5 単語間二項関係の変換

単語間二項関係の変換を行うサブシステムは，上記の W^{gda}, R^{gda} を入力とする．

単語間二項関係変換サブシステムの出力となるのは次のようなデータである．

W^{gda} 中の各日本語単語に対応する UW のリスト
$W^{unl} = \{\ uw_i\ ;\ i = 1, \cdots, n\ \}$

R^{gda} に対応する UW 間二項関係リスト
$R^{unl} = \{\ (uw^1, unlLabel_j, uw^2)\ ;$
$uw^1, uw^2 \in W^{unl},\quad j = 1, \cdots, m,$
$unlLabel_j$は UNL 二項関係ラベル $\}$

2.6 UNL 式にフォーマットする

上記の W^{unl}, R^{unl} を UNL 式にフォーマットする．その文字列が，システム全体の出力となる．

3. GDA タグ付き文からの単語間二項関係の抽出

単語間二項関係とは，二つの単語間の関係を表すものである．GDA では，タグの属性に二項関係ラベルを指定することによって明示できる．

GDA における単語間二項関係の表し方には，次の 4 つの方法がある．

- opr 属性値に二項関係ラベルを指定
- sem 属性値に二項関係ラベルを指定
- grel 属性値に二項関係ラベルを指定
- 二項関係ラベルを属性として指定（関係ラベル）

これら 4 つの表現形式で表される単語間二項関係の抽出について述べる．

3.1 opr 属性

opr 属性によって二項関係を示した場合，

第一項: opr 属性を持つエレメントの最大投射の係り先

第二項: opr 属性を持つエレメントの中の隣接項☆，またはそのエレメントの最大投射

となる．

☆ ここでは，助詞，助動詞，補助用言，接尾語，接頭語に隣から係る語句を隣接項 (adjacent argument) と呼ぶ。

```
<adp opr="pat">タコを</adp>
<v>食べる</v>
```

↓

食べる pat タコ

3.2 sem 属性

sem 属性によって二項関係を示した場合,

第一項: sem 属性を持つエレメントの最大投射の係り先

第二項: sem 属性を持つエレメントの隣接項

となる.

```
<adp>タコ<ad sem="pat">を</ad></adp>
<v>食べる</v>
```

↓

食べる pat タコ

3.3 grel 属性

grel 属性によって二項関係を示した場合,

第一項: grel 属性を持つエレメント

第二項: grel 属性を持つエレメントを H 統御☆する名詞エレメントのうち最も近いもの

となる.

```
<vp grel="agt">走る</vp><n>人</n>
```

↓

走る agt 人

3.4 関係属性

関係属性によって二項関係を示した場合,

第一項: 関係属性を持つエレメント

第二項: 関係属性を持つエレメントの先行詞

となる.

```
<vp>
  <adp>
    <n id="stone1">石</n>
    <ad>を</ad>
  </adp>
  <v>拾って</v>
</vp>
<v pat="stone1">投げた</v>
```

↓

投げた pat 石

3.5 文中の単語リストと二項関係リストの抽出

二項関係抽出段階では,上記の規則に従って文中の関係ラベルで明示された二項関係リストと,その項となる単語のリストを抽出する.ただし,形容詞が名詞を修飾している場合は,関係ラベルによって明示されていなくとも mod 関係にあると見なして抽出を行う.

次の例は,図1の入力 GDA 文から抽出した単語リスト W^{gda} と二項関係リスト R^{gda} である.

$$W^{gda} = \{ \text{若い}, \text{男}, \text{タコ}, \text{食べる} \}$$
$$R^{gda} = \{ (\text{男 mod 若い}), (\text{食べる agt 男}), (\text{食べる pat タコ}) \}$$

4. 単語間二項関係の変換

4.1 辞書引き

最初に,文中の全ての日本語単語について辞書引きを行い,各語に対応する UW 候補リストを得る.また,GDA 二項関係ラベルについても,各ラベルに対応する UNL 二項関係ラベルの候補リストを得る.

この際,UW 辞書に含まれる品詞情報を用い,元の単語と品詞が一致しない UW は UW 候補リストから削除する.下記の function **convertBinaryRelation** は,辞書引きと品詞チェックを行い,その結果を *juMap*, *labelMap* に保持した後,function **choose** を呼び出す.function **choose** では,スコアリング,候補の絞り込み・選択を行い,変換した UW リスト W^{unl} と UW 二項関係リスト R^{unl} を返すが,これについては後述する.

```
# 二項関係リストを変換
function convertBinaryRelation(W^gda, R^gda)
  juMap = nil
  labelMap = nil
  for each jw ∈ W^gda
    uwList = 辞書中の jw に対応する UW リスト
    uwList から jw と品詞が異なる UW を削除
    juMap.put(jw, uwList)
  end for
  for each gdaBiRel ∈ R^gda
    gdaLabel = gdaBiRel の関係ラベル
    unlLabelList =gdaLabel に対応する
                  UNL 関係ラベルリスト
    labelMap.put(gdaLabel, unlLabelList)
  end for
  (W^unl, R^unl) =choose(W^gda, R^gda, juMap, labelMap)
  return (W^unl, R^unl)
end convertBinaryRelation
```

4.2 変換候補の絞り込み

UW 共起情報を制約として用い,変換候補の絞り込みを行う.

☆ ここでは,y または y を含む構成素が x に係るとき,x が y を H 統御するという.

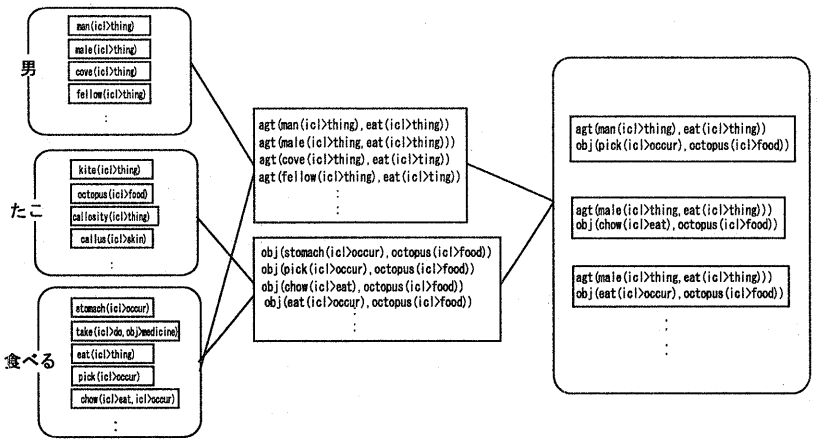

図 4 候補の絞り込みの例「男がタコを食べる」

UNL 知識ベースは可能な全ての UW 間二項関係を持つので，知識ベースから UW 間共起情報と UW 間の上位下位関係を得ることができる．しかし本稿では，それに類し実験に必要なエントリのみを持つ UW 共起辞書を，EDR[6] 共起辞書を参考に手動で作成し，知識ベースの代わりにこれを用いた．

まず，次の例を変換する場合について考える．

食べる **pat** タコ

の場合，まず辞書引きによって単語「食べる」,「タコ」に対応する UW リスト，GDA 関係ラベル「pat」に対応する UNL 関係ラベルリストを得る．

```
食べる → { eat(icl>do, obj>food),
           take(icl>do),
           have(icl>do,obj>drink) }

タコ → { octopas(icl>food),
         kite(icl>toy),
         callus(icl>skin) }

pat → { obj }
```

このように，第一項「食べる」，第二項「タコ」のそれぞれに対して複数の UW 候補があり，曖昧性の解消が必要となる．また，上の例では GDA 二項関係ラベル **pat** に対応する UNL 二項関係ラベルの候補は **obj** 1 つであったが，複数の候補が存在する場合もある．

この曖昧性を解消するために，前述した UW の共起辞書を用いる．

まず，各 UW 候補の組み合わせによる UW 間二項関係にスコアを付ける．そのために，スコアリングの対象となる UW 間二項関係の第一項・第二項の上位語の共起情報を，UW の共起辞書から検索する．

4.2.1 共起情報の検索

下に，UW 間二項関係 $uwBiRel$ の上位語の共起情報を検索する function **searchCoOccurence** を示す．

```
#共起情報の検索を行う
function searchCoOccurence(uwBiRel)
  (uw1, label, uw2) = uwBiRel
  birelList = nil
  # uw1 かその上位語を第一項とする二項関係を検索
  uw1^0 = uw1
  while uw1^0 is not null
    uw2^0List =getCoOccTerm2List(uw1^0, label)
    if uw2^0List isn't nil then
      birelList.append(
        {(uw1^0, label, uw2^0) ; uw2^0 ∈ uw2^0List})
      break while
    end if
    uw1^0 =getBroaderWord(uw1^0)
  end while
  # uw2 かその上位語を第二項とする二項関係を検索
  uw2^0 = uw2
  while uw2^0 is not null
    uw1^0List =getCoOccTerm1List(uw2^0, label)
    if w1^0List isn't nil then
      birelList.append(
        {(uw1^0, label, uw2^0) ; uw1^0 ∈ uw1^0List})
      break while
    end if
    uw2^0 =getBroaderWord(uw2^0)
  end while
  return birel
end searchBaseRelation

# uw の 1 段階上位の語を返す
function getBroaderWord(uw)
  return getCoOccTerm2List(uw, UNL 関係ラベル"icl")
end getBroaderWord

function getCoOccTerm2List(uw, label)
```

uw あるいは uw の上位語を第一項にとり，二項関係ラベル $label$ によって表される二項関係の，第二項となり得る UW のリストを返す

function **getCoOccTerm1List**($uw, label$)
uw あるいは uw の上位語を第二項にとり，二項関係ラベル $label$ によって表される二項関係の，第一項となり得る UW のリストを返す

このようにして検索した UW 共起情報を用いて，UW 間二項関係にスコアを付ける．ここで，先に述べた function **choose** を以下に示す．**choose** では，共起情報を **searchCoOccurence** で検索し，**scoreBiRel** でスコア付けを行い，それをもとに候補の絞り込みを行っている．

```
juMap: 日本語単語をキーとし，UW リストを値とするマップ
labelMap: GDA 関係ラベルをキーとし，UNL 関係ラベルリストを値とするマップ
# 候補の絞り込み
function choose(W^gda, R^gda, juMap, labelMap)
  biRelMap = nil
  scoreMap = nil
  for each jwBiRel ∈ R^gda
    (jw1, gdaLabel, jw2) = jwBiRel
    labelList = labelMap.get(gdaLabel)
    term1List = juMap.get(jw1)
    term2List = juMap.get(jw2)
    uwBiRelList = { (uw1, label, uw2) ;
                    uw1 ∈ term1List,
                    uw2 ∈ term2List,
                    label ∈ labelList }
    biRelMap.put(jwBiRel, uwBiRelList)
    scoreMap = nil
    for each uwBiRel ∈ uwBiRelList
      # 共起情報の検索
      coOccList =searchCoOccurence(uwBiRel)
      # スコアリング
      score = scoreBiRel(uwBiRel, coOccList)
      scoreMap.put(uwBiRel, score)
    end for
  end for
  # 絞り込みとソート
  candList =limitCandidate(W^gda, R^gda,
            biRelMap, scoreMap)
  (W^unl, R^unl) = candList の先頭の要素 (スコア最大)
  return (W^unl, R^unl)
end choose

# 絞り込みとソート
function limitCandidate(W^gda, R^gda,
                        biRelMap, scoreMap)
  candList = {(W^unl, R^unl) ;
              biRelMap 中の全組み合わせ }
  candList を scoreMap 中のスコア順にソート
  return candList
end limitCandidate
```

4.3 シソーラスを用いたスコアリング

変換候補の UW 間二項関係 $(uw_1, label, uw_2)$ のスコアリングを行う．検索した共起情報 $(uw_1^0, label, uw_2^0)$ の第一項，第二項とのシソーラス上の距離を用いる．

2 つの UW$uw, uw'$ 間距離 $d(uw, uw')$ のシソーラス上のパス数を距離 $\mathbf{distance}(uw, uw')$ とし，$(uw_1, label, uw_2)$ のスコアを次のように定める．

$$\mathbf{score}(uw_1, label, uw_2) = 2^{-\mathbf{distance}(uw_1, uw_1^0)} + 2^{-\mathbf{distance}(uw_2, uw_2^0)}$$

文全体のスコアは文中の二項関係のスコアの総和とする．つまり，

R^{unl}, W^{unl} に対し，

$$\mathbf{score}(R^{unl}) = \sum_{rel \in R^{unl}} score(rel)$$

とする．

5. 実　　装

本システムは，Java によって実装中である．Java 上での DOM (Document Object Model) の実装が用意されており，DOM を用いることによって GDA タグの操作を容易に実装できるため，Java を用いた．

また，システムの効率及び変換精度の面からの評価を行う必要がある．

6. おわりに

本稿では GDA から UNL への変換の際に生じる曖昧性の解消を，単語間二項関係の制約などを用いて絞込みを行い候補を選定する方法を示した．

謝辞　本研究に御協力頂いた国連大学高等研究所 内田裕士氏に深謝致します．

参 考 文 献

1) 大域文書修飾 Global Document Annotation, http://www.etl.go.jp/etl/nl/gda/
2) The Universal Networking Language Specifications Version 3.0 (2000), http://www.unl.ias.unu.edu/unlsys/spec/unls20.htm
3) 伊藤他: 意味構造タグ付き文書から中間言語への変換ツール群の開発, 情報処理全国大会 (2000)
4) Katashi Nagao and Koiti Hasida: Automatic Text Summarization Based on the Global Document Annotation, COLING-ACL (1998)
5) 日本語形態素解析システム『茶筌』 version 2.2.1 使用説明書 (2000), http://chasen.aist-nara.ac.jp/index.html.ja
6) EDR 電子化辞書 1.5 版 仕様説明書 (1996),http://www.iijnet.or.jp/edr/J_index.html