

自然言語処理2014
-平成26年11月10日(No6)-

東京工科大学
コンピュータサイエンス学部
亀田弘之

復習

- (英文の場合の復習 & 確認)
- (授業ではさっと眺めるだけにしますが、不安な人はじっくり読んでください。)

今日は日本語処理の話
も始めたいので...

NLPのプログラムを書いてみよう！

I. Prologのインストール

II. プログラム作成手順

1. 解析対象テキストの収集
2. IC分析(直接構成素分析, Immediate Constituent Analysis)
3. 形式文法の設定
4. Prolog形式への書き換え
5. NLPプログラムの実行(その1)
6. 構文木を出力するプログラムへの拡張
7. NLPプログラムの実行(その2)

NLPのプログラムを自力で書く

1. Prologのインストール(www.swi-prolog.org)
2. プログラム作成手順
 1. **言語要素分析** <= 対象言語に関する知識*が必要
 2. **形式文法の設定** <= 形式言語に関する知識が必要
 3. **Prolog形式への翻訳** <= Prologに関する知識が必要
 4. NLPプログラムの実行(その1)
 5. 構文木を出力するプログラムへの拡張
 6. NLPプログラムの実行(その2)

*: 語彙や文法など

[手順0]: 処理対象の収集

Tom broke the cup.

例

研究課題: 処理対象の文をどのようにして集めればよいのか?

- Webから集める?
- 文法書の例文を集める?
- 語学の教科書(例: 中学校3年間の英語の教科書)の例文?
- 適切な例文を自分で作成?
- それとも...

参考情報

1. Project Gutenberg (<http://www.gutenberg.org/>)
2. 青空文庫 (<http://www.aozora.gr.jp/>)
3. Oxford Text Archive (<http://ota.oucs.ox.ac.uk/>)
4. Linguistic Data Consortium
(<https://www.ldc.upenn.edu/>)
5. その他(いろいろあります)
 - PythonやRubyで、テキスト収集プログラムを作成？
(To collect a various genre of text from the Web,
Twitter and etc. by self-made tools in Python and etc.
Let's try!)

[手順2]:文法の設定

例

文法 $G = \langle V_n, V_t, \sigma, P \rangle$

ここで、

- ・非終端記号の集合:

$V_t = \{s, vp, np, n, v, d\}$

- ・終端記号の集合:

$V_n = \{tom, cup, broke, the\}$

- ・開始記号 $\sigma = s$

- ・書き換え規則群 P : 右記の通り。

{ $s \rightarrow n, vp.$
 $vp \rightarrow v, np.$
 $np \rightarrow d, n.$
 $n \rightarrow tom.$
 $n \rightarrow cup.$
 $v \rightarrow broke.$
 $d \rightarrow the. \}$

統語規則

単語辞書

研究課題

- 文法を設計するのは、簡単ではない。
- 人によって、設計内容が変わる？
- 科学的客観性はどうなるのか？
- 人間の主観によらず、科学的かつ客観的な方法で、文法設計はできないのか？

==> 人工知能(言語獲得)、
機械学習(帰納論理プログラミング)、
心理学(学習・発達・コミュニケーション等)、 etc.

[手順3]:Prolog形式へ変換

s(A,C):-n(A,B),vp(B,C).
vp(A,C):-v(A,B),np(B,C).
np(A,C):-d(A,B),n(B,C).
n([tom | T],T).
n([cup | T],T).
v([broke | T],T).
d([the | T],T).

例

Version1 (Prog1.pl)

Version2 (Prog2.pl)

例

```
s(A, C, s(_n, _vp)) :- n(A, B, _n), vp(B, C, _vp).  
vp(A, C, vp(_v, _np)) :- v(A, B, _v), np(B, C, _np).  
np(A, C, np(_d, _n)) :- d(A, B, _d), n(B, C, _n).
```

```
n([tom|T], T, n(tom)).
```

```
n([cup|T], T, n(cup)).
```

```
v([broke|T], T, v(broke)).
```

```
d([the|T], T, d(the)).
```

問題： 前頁のversion1との
違いは何？

動作確認

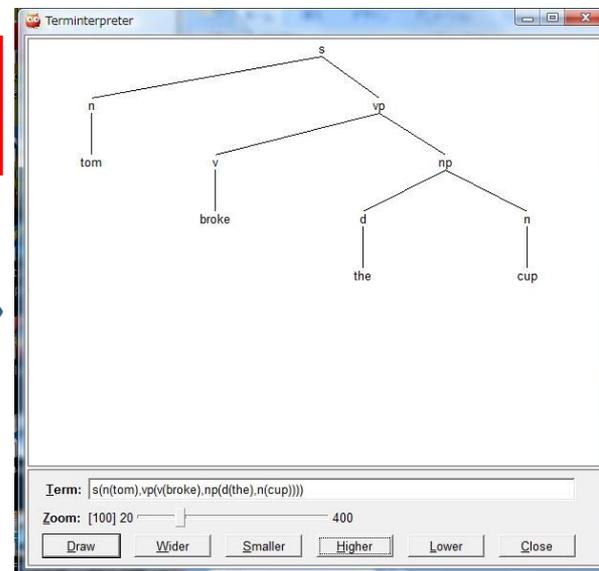
- 実際にswi-Prologで実行してみよう！
- drawterm.pl を使うと、木構造（構文構造）を図として表示することができるよ！

```
?- s([tom,broke,the,cup],[],P),draw_term(P).  
P = s(n(tom), vp(v(broke), np(d(the), n(cup))))
```



SWI Prolog

<http://www.swi-prolog.org/>



今日の内容

1. 構文解析プログラムの作成手順を会得する
2. 構文解析プログラムが自力で書ける
3. 日本語のプログラムも分かる。

1. 構文解析プログラムの 作成手順を会得する

- 説明は終わりました。
- 実作業を通して会得しましょう。

2. 構文解析プログラムを 自力で書ける

それでは、構文解析プログラムを 自分で書いて見よう！

- 例文

- Time flies like an arrow.

- (教科書p.42-47 参照のこと)

Let's challenge!

作成手順の概要

1. 処理対象を決める
2. 処理対象の各文に対して、
統語構造の分析を行う
3. 上記の分析結果をもとに、
文法を書き下す
4. 上記で得られた文法を、
Prologの形式に書きかえる
5. 動作を確認する

処理対象文

- **Tom went to a park.** または
- Time flies like an arrow.

ソースコードの例(未記入)

自分で作ってみよう！(5分)

統語規則部分

```
sentence(A,C,  
    stentence(_noun,_verbal_phrase)) :-  
noun(A,B,_noun),  
verbal_phrase(B,C,_verbal_phrase).
```

```
verbal_phrase(A,C,  
  verbal_phrase(_intransitive_verb,  
    _prepositional_phrase)):-  
  intransitive_verb(A,B,_intransitive_verb),  
  
  prepositional_phrase(B,C,_prepositional_phrase).
```

```
prepositional_phrase(A,B,  
  prepositional_phrase(_preposition,  
    _noun_phrase)):-  
  preposition(A,B,_preposition),  
  noun_phrase(A,B,_noun_phrase).
```

```
noun_phrase(A,C,  
  noun_phrase(_determiner,_noun)):-  
  determiner(A,B,_determiner),  
  noun(B,C,_noun).
```

単語辞書部分

%time

noun([time | T],T,noun(time)).

transitive_verb([time | T],T,transitive_verb(time)).

%flies

intransitive_verb([flies | T],T,

intransitive_verb(flies)).

%like

transitive_verb([like | T],T,
transitive_verb(like)).

preposition([like | T],T,preposition(like)).

adjective([like | T],T,adjective(like)).

%an

determiner([an | T],T,determiner(an)).

%arrow

noun([arrow | T],T,noun(arrow)).

noun([flies | T],T,noun(flies)).

英文はまずはここまで

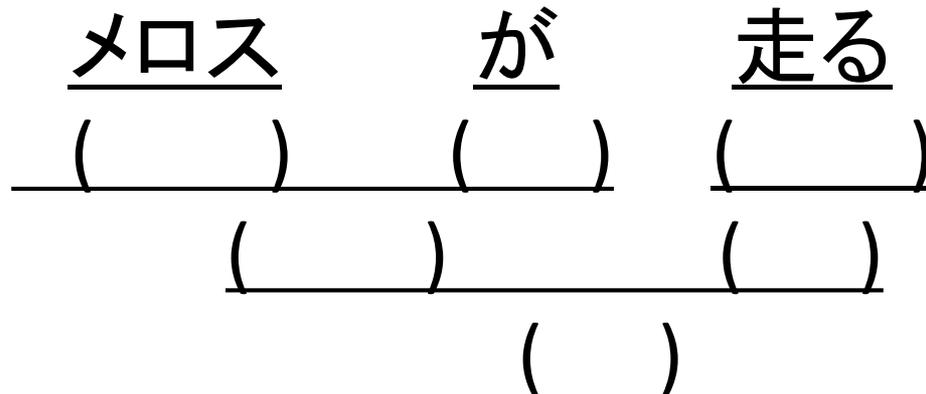
- 以下では、日本語処理について考えましょう。

日本語処理

- 処理対象

例： メロスが走る

- 言語要素分析



日本語処理

- 処理対象

例： メロスが走る

- 言語要素分析

<u>メロス</u>	<u>が</u>	<u>走る</u>
(名詞)	(助詞)	(動詞)
<u>(主語)</u>		<u>(述語)</u>
(文)		

- 形式文法の設定

文法 $G = \langle V_n, V_t, \sigma, P \rangle$

ただし、

$V_n = \{ \quad \quad \quad \}$

$V_t = \{ \quad \quad \quad \}$

$\sigma = \text{文}$

$P = \{ \quad \quad \quad \}$

- 形式文法の設定

文法 $G = \langle V_n, V_t, \sigma, P \rangle$

ただし、

$V_n = \{ \text{文, 主語, 述語, 名詞, 助詞, 動詞} \}$

$V_t = \{ \text{メロス, が, 走る} \}$

$\sigma = \text{文}$

$P = \{ \text{文} \rightarrow \text{主語} + \text{述語}, \text{主語} \rightarrow \text{名詞} + \text{助詞}, \text{述語} \rightarrow \text{動詞} \}$

- Prolog形式への書き換え
文 :- 主語, 述語.
主語 :- 名詞, 助詞.
述語 :- 動詞.
名詞(メロス).
助詞(が).
動詞(走る).

- Prolog形式への書き換え(2)
文(A,C) :- 主語(A,B), 述語(B,C).
主語(A,C) :- 名詞(A,B), 助詞(B,C).
述語(A,B) :- 動詞(A,B).
名詞([メロス|T],T).
助詞([が|T],T).
動詞([走る|T],T).

- Prolog形式への書き換え(3)

文(A,C,文(_主語,_述語)) :-

主語(A,B,_主語), 述語(B,C,_述語).

主語(A,C,主語(_名詞,_助詞)) :-

名詞(A,B,_名詞), 助詞(B,C,_助詞).

述語(A,B,述語(_動詞)) :- 動詞(A,B,_動詞).

名詞([メロス|T],T,名詞(メロス)).

助詞([が|T],T,助詞(が)).

動詞([走る|T],T,動詞(走る)).

動作の確認

-? 文([メロス,が,走る],[,],X).

文(A ,B, 文(_主部, _述部)

黒板にて動作トレースをします。Prologプログラムの実行過程、ユニフィケーション(同一視できるかな処理)を理解してください。

統語解析＋形態素解析の版

- 今までの例は統語解析だけを行うものであった。英文ではまだしも、日本語の場合は形態素解析を避けて通ることはできない。

どうすればいいのだろうか？

形態素解析も合わせてできる版

文(A,C,文(_主語,_述語)) :-

主語(A,B,_主語), 述語(B,C,_述語).

主語(A,C,主語(_名詞,_助詞)) :-

名詞(A,B,_名詞), 助詞(B,C,_助詞).

述語(A,B,述語(_動詞)) :- 動詞(A,B,_動詞).

名詞([メ,ロ,ス|T],T,名詞(メロス)).

助詞([が|T],T,助詞(が)).

動詞([走,る|T],T,動詞(走る)).

次回、この続きをやります。

- なぜ、形態素解析も合わせてできるのか、考えてみてください。
- 次週は未知語処理の話しをします。

レポート課題(第6日目)

- 今日やった内容を、自分なりにまとめ、「自然言語処理プログラムの書き方(A manual of how to write a NLP program.)」というタイトルのマニュアルを作成しなさい。
- ページ数やスタイルは自由ですが、表紙や目次は付けるようにしてください。
- 提出日は、11月17日(月)とします。

意味の処理

- 来週以降に向けて、少し意味処理について話しておきます。詳細は後日あらためてお話しします。

意味とは

格文法

- Tom broke the cup with a hammer.
- 動作主格、対象格、経験者格、道具格、結果格、源泉格、目標格、對抗動作主格、時間格、場所格 など