実習:レシピの言語処理の現状

京都大学
笹田鉄郎、前田浩邦、森信介
2013年8月18日

公開に際しての注意

- 必要環境
  - Perl
  - KyTea
  - Eda
  - Firefox (ver. 14.0.1以前のバージョン)
- 著作権の関係上、係り受け解析の実習で利用した学習コーパスを公開することはできません。
ご了承ください。

目次

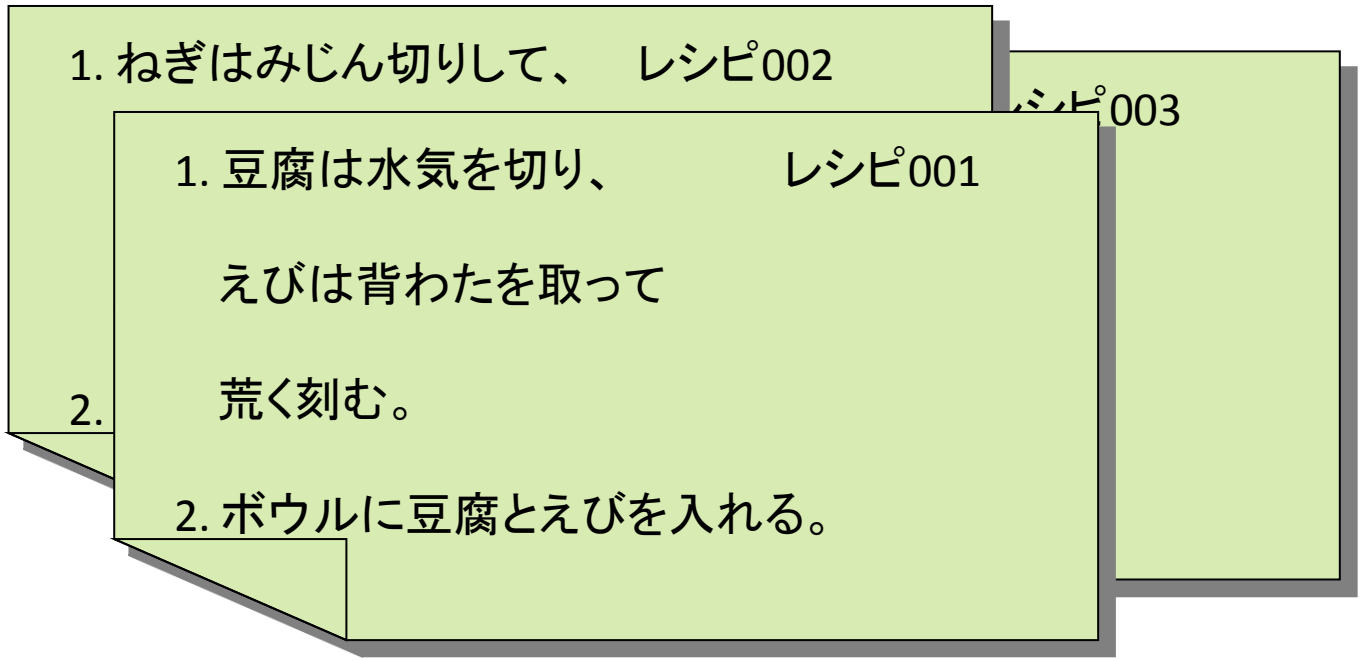
1. はじめに
2. 注意事項
3. アノテーション支援ツールPNAT
4. 自動単語分割
5. 固有表現認識
6. 係り受け解析
7. まとめ

1. はじめに

- よろしくおねがいします
- レシピの言語処理
  →情報の抽出と構造化

1. はじめに

- レシピの言語処理
  →情報の抽出と構造化



1. はじめに

- レシピの言語処理
  →情報の抽出と構造化

1. 豆腐は水気を切り、　　レシピ001

   えびは背わたを取って

   荒く刻む。

2. ボウルに豆腐とえびを入れる。

1. はじめに

- レシピの言語処理
  →情報の抽出と構造化

食べ物、道具、
動作、…

1. 豆腐 は 水気 を 切り 、　　レシピ001

   えび は 背わた を 取って

   荒く 刻む 。

2. ボウル に 豆腐 と えび を 入れる 。

1. はじめに

- レシピの言語処理
  →情報の抽出と構造化

自動単語分割
&
固有表現認識

1. 豆腐 は 水気 を 切り 、　　レシピ001

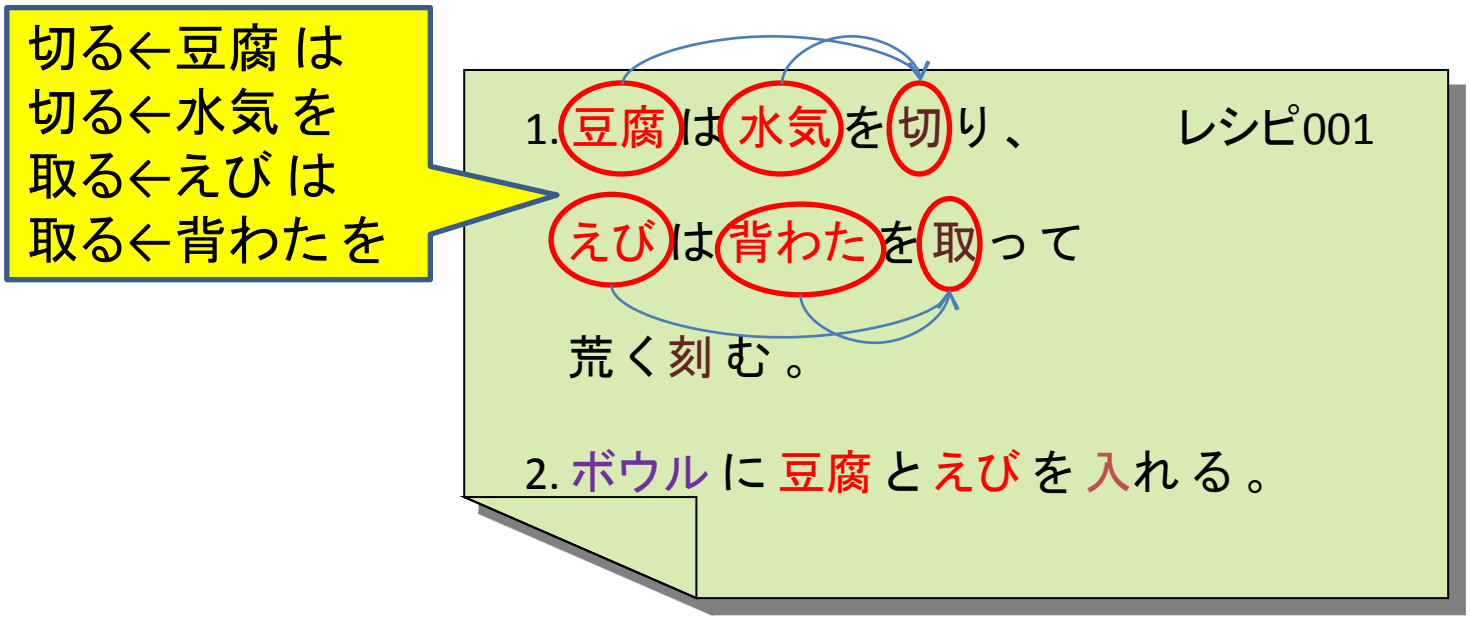
   えび は 背わた を 取って

   荒く 刻む 。

2. ボウル に 豆腐 と えび を 入れる 。

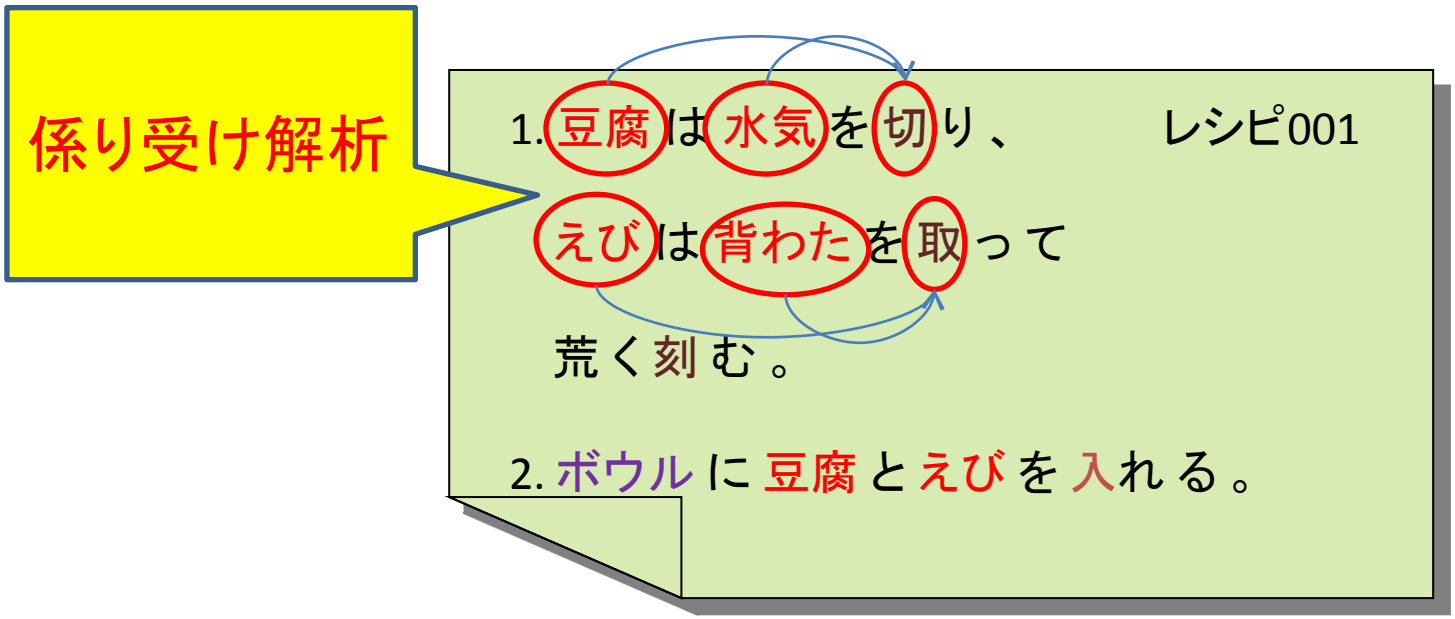
1. はじめに

- レシピの言語処理
  →情報の抽出と構造化



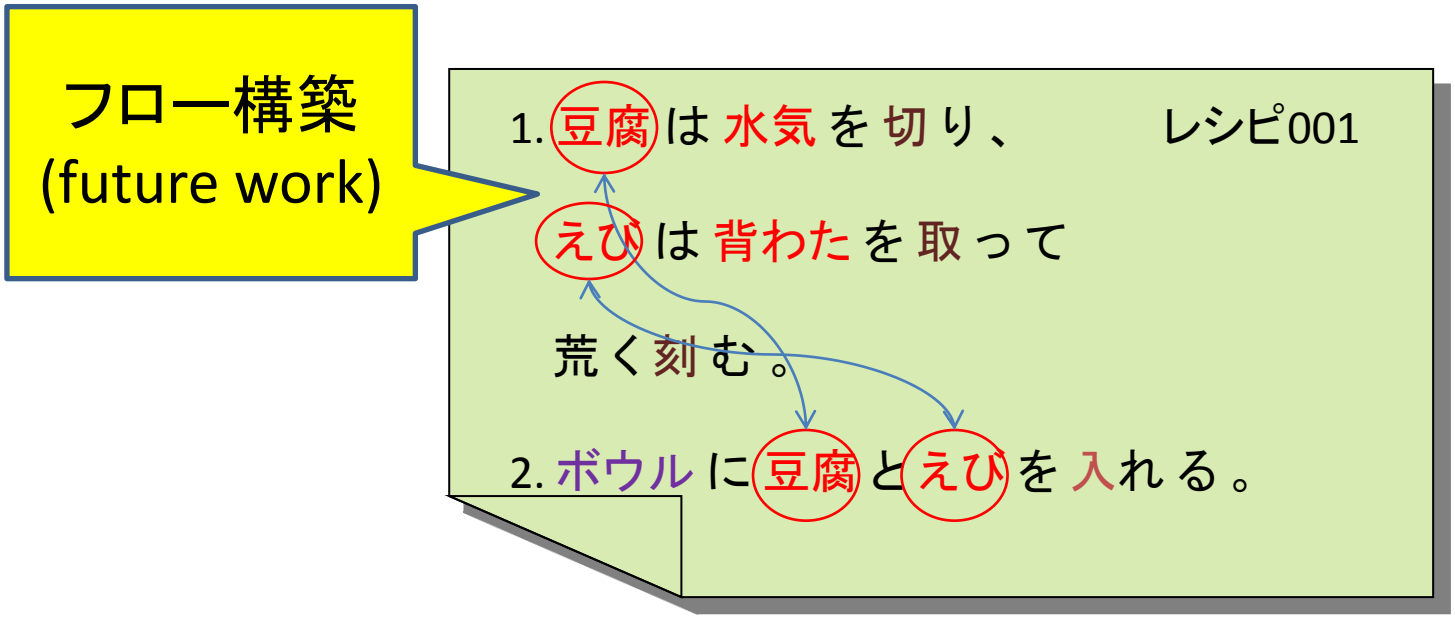
1. はじめに

- レシピの言語処理
  →情報の抽出と構造化



1. はじめに

- レシピの言語処理
  →情報の抽出と構造化



1. はじめに

- レシピの言語処理
  →情報の抽出と構造化
  - レシピ検索、調理補助、…

1. はじめに

- レシピの言語処理
  →情報の抽出と構造化
  - レシピ検索、調理補助、…
- 実際にやってみましょう
  - 自動単語分割
  - 固有表現抽出
  - 係り受け解析

2. 注意事項

- ご用意いただいたPCで実習します
  - 用意した実習環境（USBブートのLinux）を使用
  - Linux環境があれば基本的に同じことができます
- 解説 -> 実習を1セット、3回
  - 自動単語分割 (practice/WS)
  - 固有表現抽出 (practice/NE)
  - 係り受け解析 (practice/DA)
- 随時質問を受け付けます

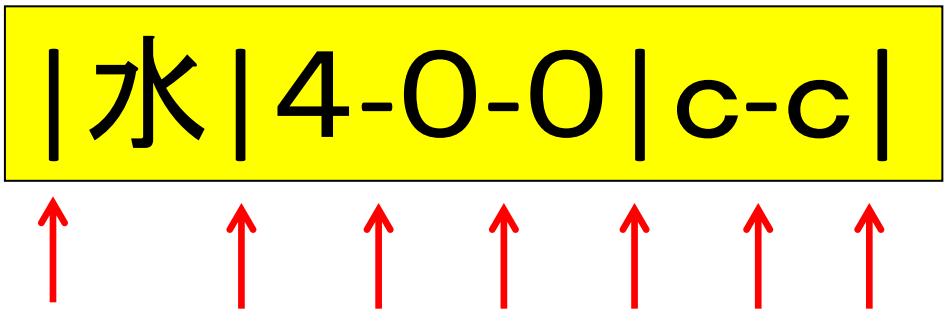
3. アノテーション支援ツールPNAT

http://plata.ar.media.kyoto-u.ac.jp/mori/research/topics/PNAT/

- 便利な機能色々
  - 単語分割位置修正
  - GUIによるタグ、係り受け付与
  - 不正NEタグ判定
- 必要動作環境
  - Firefox(ver .14以前, 全OS)
  - 最新版（未公開）ではFirefox, Google Chorome,Internet Explorer (Windows XP, Vista, 7)

4. 自動単語分割

- 日本語は分かち書きされない言語
  - 様々な言語処理のためにまず単語分割
- KyTeaを使用してモデル学習・解析
  - 点予測による単語分割
  - 2値分類: 単語境界の有無を推定する



4. 自動単語分割

- 日本語は分かち書きされない言語
  - 様々な言語処理のためにまず単語分割
- KyTeaを使用してモデル学習・解析
  - 素性ファイルを使うことで、BCCWJと同様の言語資源から学習可能（約5万文）
  - 本実習ではノートPC用の小規模セット（約1万文）
  - 部分的アノテーションコーパスに対応

4. 自動単語分割

以下[% **入力するコマンド**]
コピー&ペーストで入力すると確実です

デスクトップ→端末
% **cd ~/practice/WS**

gedit, emacs, lv, less, moreなどで
　ファイルの内容を参照できます

4. 自動単語分割

train-kyteaによるモデル学習

基本はフルアノテーションコーパス

(スペース区切り)から学習

% **cat train/sample.word**

% **train-kytea -full train/sample.word -model sample.kwm**

作成したモデル(sample.kwm)を指定し、テスト文(test.sent)をkyteaで解析

% **kytea -model sample.kwm < test.sent**

4. 自動単語分割

良いモデルを作る→学習コーパスを増やす

あらかじめ用意した素性頻度ファイルから学習（BCCWJ約1万文相当）

% **train-kytea -notags -feat train/base.feat -model base.kwm**

作成したモデル(base.kwm)を指定して解析

% **kytea -model base.kwm < test.sent**

4. 自動単語分割

解析結果をtest01.wordに保存

% **kytea -model base.kwm < test.sent**
> work/test01.word

単語分割精度を計算し、保存・確認

% **perl WordAccu.pl work/test01.word**
test-gold.word >> accu.text

% **cat accu.text**

4. 自動単語分割

精度(Accuracy)について

- Precision(適合率、精度)
  - 正解単語数 / システム出力の単語数
  - ごみの少なさ（正確性）
- Recall (再現率)
  - 正解単語数 / テストセットの単語数
  - もれの少なさ（網羅性）
- F-measure (F値)
  - 適合率と再現率の調和平均

4. 自動単語分割

PNATを使って結果を確認

kyteaの解析結果をtree形式に変換

% **perl word2tree.pl < work/test01.word > work/test01.tree**

4. 自動単語分割

デスクトップ->PNAT (firefoxが起動します)
file://のスクリプトが～～という警告文が出ます
“今後も同様に処理する”をチェックし、“許可”をクリック
cea/practice/WS/work/test01.treeを自動的に開きます
“不許可”をクリックしてしまったり、開かなかった場合は必ずお知らせください

FILE OPENから任意のtreeファイルを開けます
自由に動かしてみましょう
cea/practice/WS/work/test01.treeは以後使わないので、編集しても問題ありません

4. 自動単語分割(分野適応)

部分的アノテーションコーパスを追加して
分野適応モデルを学習

% **train-kytea -feat train/base.feat -part train/adapt.part -model adapt.kwm**

分野適応モデルでtest.sentを解析

% **kytea -model adapt.kwm < test.sent**

% **kytea -model adapt.kwm < test.sent > work/test02.word**

4. 自動単語分割(分野適応)

単語分割精度を測り、確認

% **perl WordAccu.pl work/test02.word**
test-gold.word >> accu.text

% **cat accu.text**

約8時間の作業(adapt.partの作成)
による改善

5. 固有表現認識

- 構造化の前処理
  - 各単語に対してBIO2記法のNEタグを推定
  - 本実習では、正しく単語分割された テキストを入力とする
- KyTeaを使用
  - 点予測によるタグ推定

デスクトップ→端末

% **cd ~/practice/NE**

5. 固有表現認識

train-kyteaによるタグ推定モデル学習

% **cat train/base.iob2**

% **train-kytea -nows -full train/base.iob2 -global 1 -solver 6 -model base.knm**

test.word (test.sentの正しい単語分割結果)にNEタグを確率的に付与

% **kytea -out conf -nows -tagmax 0 -model base.knm < test.word > work/test01.conf**

% **cat work/test01.conf**

5. 固有表現認識

解釈可能な最適タグ列を探索

% **perl NEsearch.pl work/test01.conf work/test01.iob2**

% cat work/test01.iob2

NE推定精度を計算し、保存・確認

% **perl NEAccu.pl work/test01.iob2 test-gold.iob2 >> accu.text**

% **cat accu.text**

5. 固有表現認識(分野適応)

PNATを用いたアノテーションの実習
FILE OPEN -> cea/practice/NE/train/adapt.tree

最後の1文を修正してください
F:食材、T:道具、D:時間、Ac:調理動作
ex. オーブン: T-B

Iタグ(inside)が先頭に来るなど、不正なフォーマットになると赤く表示されます
SAVEボタンを押すと保存されます

5. 固有表現認識(分野適応)

編集したtree形式のファイルを
train-kytea対応形式に変換

% **perl tree2iob2.pl < train/adapt.tree >**
train/adapt.iob2

5. 固有表現認識(分野適応)

アノテーションコーパスを追加して学習

% **train-kytea -nows -full train/base.iob2 -full train/adapt.iob2 -global 1 -solver 6 -model adapt.knm**

test.wordにNEタグを確率的に付与

% **kytea -model adapt.knm -out conf -nows -tagmax 0 < test.word > work/test02.conf**

5. 固有表現認識(分野適応)

解釈可能な最適タグ列を探索

% **perl NEsearch.pl work/test02.conf work/test02.iob2**

NE推定精度を計算し、保存・確認

% **perl NEAccu.pl work/test02.iob2 test-gold.iob2 >> accu.text**

% **cat accu.text**

6. 係り受け解析

- レシピテキストを構造化
  - 修飾・被修飾の関係を各文に付与
  - 被修飾語から修飾語へ矢印が行くように設定

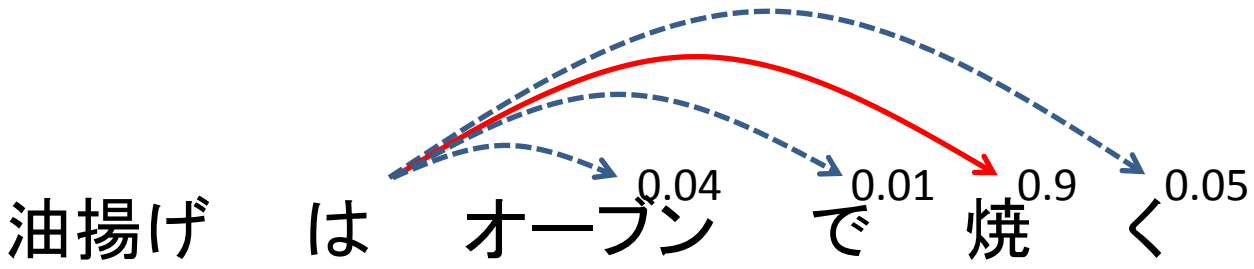
- Edaを使用
  - 点予測による係り受け解析

6. 係り受け解析

- 点予測による係り受け解析
  1. 係り先候補に対して独立にスコアを計算

     $\sigma(w \rightarrow w') = \exp\{ feats(w \rightarrow w') \cdot \theta \} / (\text{正規化定数})$

  2. スコアの一番高い係り受けを選択



6. 係り受け解析

% **cd ~/practice/DA**

学習コーパスの確認

% **cat train/base.tree**

train-edaによるモデル学習

% **train-eda -c train/base.tree -m base.edm**

6. 係り受け解析

作成したモデルでtest.treeを解析

% **eda -m base.edm < test.tree > work/test01.tree**

解析精度を計算、結果を確認

% **perl eval.pl test.gold work/test01.tree > accu01.text**

% **cat accu01.text**

分野適応

- 分野適応のコーパスが必要
  - 一般分野のコーパスのみでは精度が低い
  - レシピ特有の表現に対してアノテーションが必要
- 分野適応のコーパスは早く作りたい

部分的アノテーション

フルアノテーション

001	002	ジャガイモ	名詞	0
002	005	は	助詞	0
003	004	皮	名詞	0
004	005	を	名詞	0
005	006	む	語尾	0
006	007	い	語尾	0
007	010	て	助詞	0
008	009	水	名詞	0
009	010	に	助詞	0
010	011	さら	動詞	0
011	012	し	語尾	0
…				

部分的アノテーション

001	002	ジャガイモ	名詞	0
002	005	は	助詞	0
003	-1	皮	名詞	0
004	-1	を	名詞	0
005	-1	む	語尾	0
006	-1	い	語尾	0
007	-1	て	助詞	0
008	-1	水	名詞	0
009	-1	に	助詞	0
010	-1	さら	動詞	0
011	-1	し	語尾	0
…				

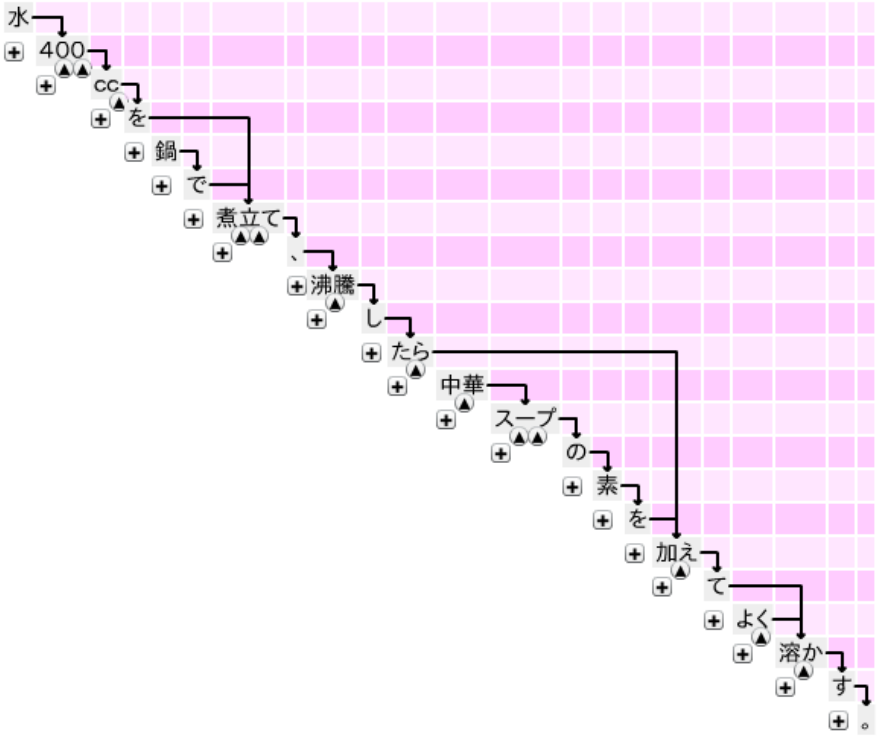
- レシピ特有の係り受けのみをアノテーション

  → **迅速な分野適応が可能**

部分的アノテーション

PNATを使って部分的アノテーションが可能

単語分割等と同様の作業なので省略します



6. 係り受け解析(分野適応)

分野適応のコーパスを確認

% **cat train/adapt.tree**

部分的アノテーションコーパスを
追加して学習

% **train-eda -c train/base.tree -c
train/adapt.tree -m adapt.edm**

6. 係り受け解析(分野適応)

適応済みモデルでtest.treeを解析

% **eda -m adapt.edm < test.tree > work/test02.tree**

解析精度を計算、結果を確認

% **perl eval.pl test.gold work/test02.tree > accu02.text**

% **cat accu02.text**

7. まとめ

- レシピの言語処理
  →情報の抽出と構造化

自動単語分割
&
固有表現認識

1. 豆腐 は 水気 を 切り 、　　レシピ001

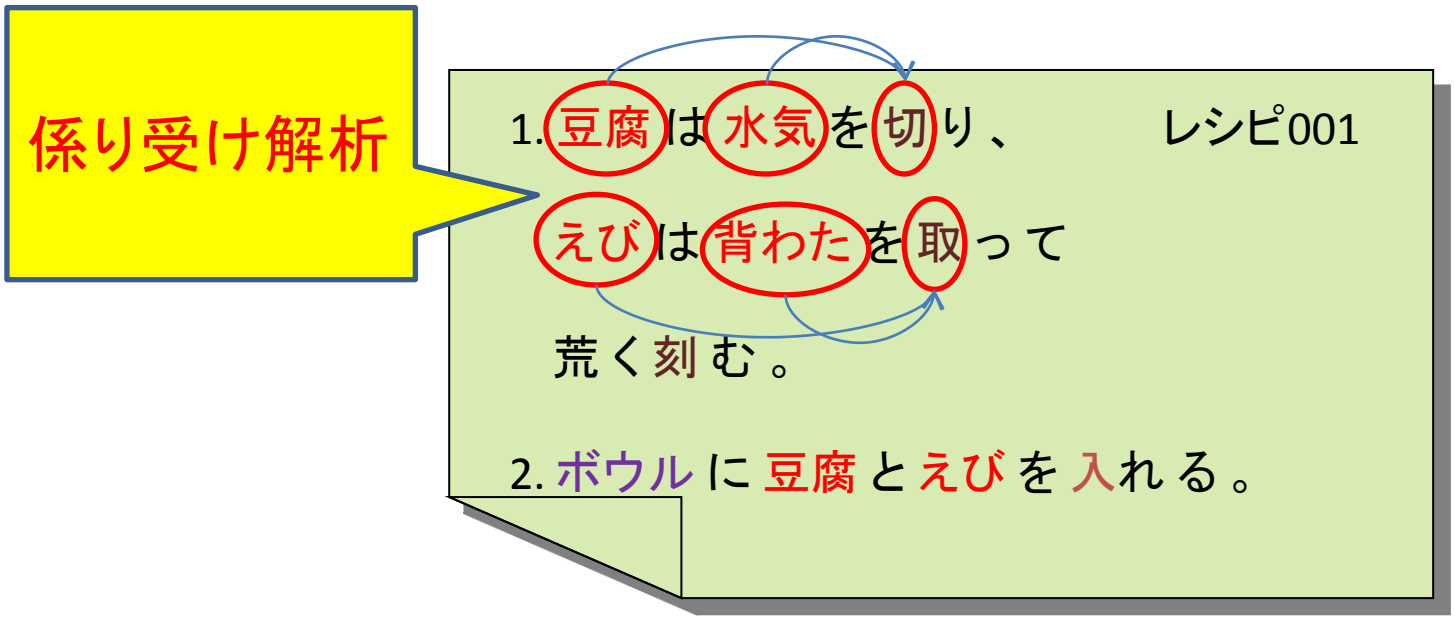
   えび は 背わた を 取って

   荒く 刻む 。

2. ボウル に 豆腐 と えび を 入れる 。

7. まとめ

- レシピの言語処理
  →情報の抽出と構造化



7. まとめ

- レシピの言語処理
  →情報の抽出と構造化

フロー構築
(future work)