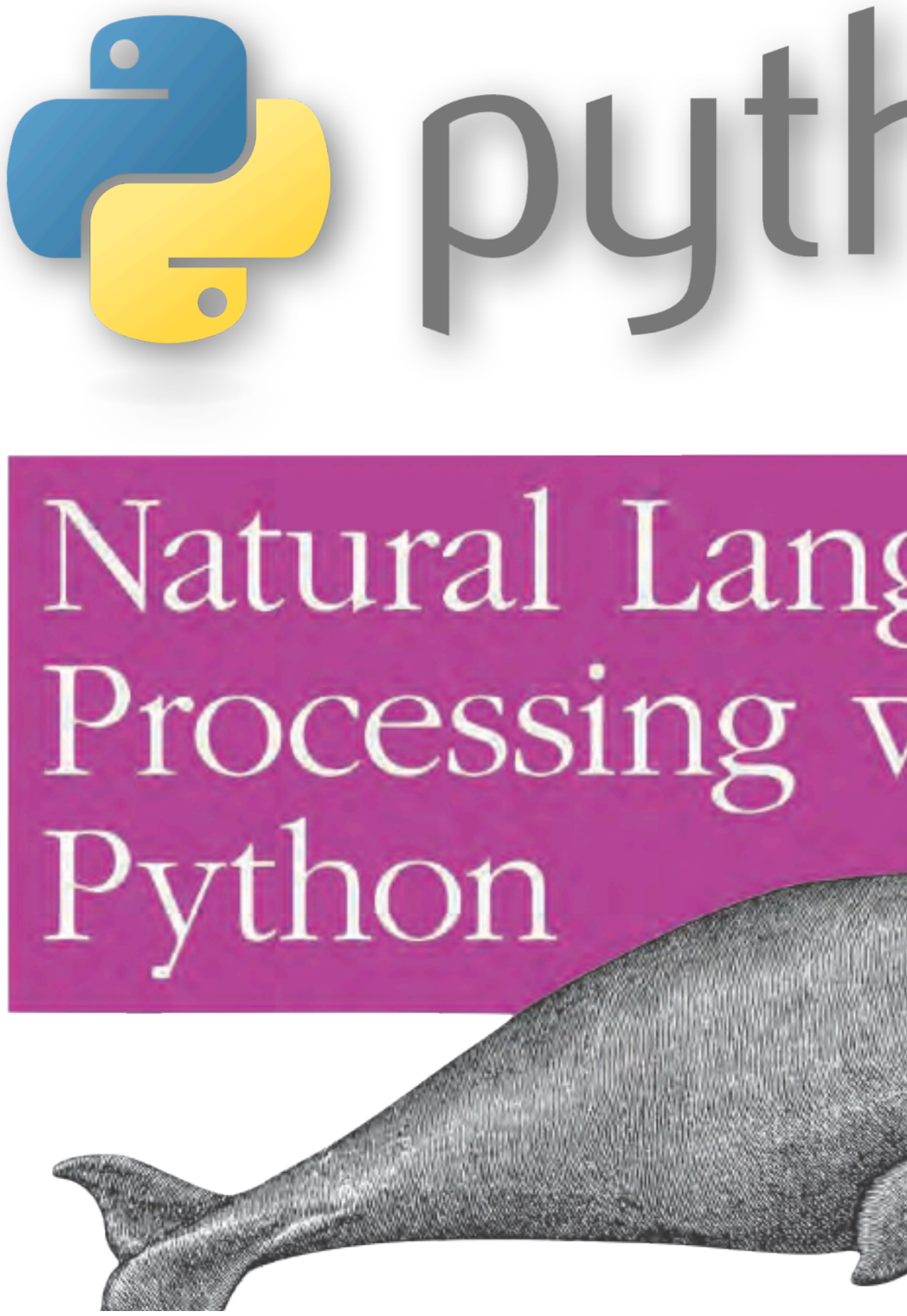
Python/NLTKでつくる
英語前置詞誤り訂正器

奈良先端科学技術大学院大学
自然言語処理学研究室

スプリングセミナー2012

What is NLTK?

イントロダクション



This is NLTK!!!

- Natural Language Tool Kit
- Pythonのための自然言語処理ライブラリ
- ”Natural Language Processing with Python”
  「入門 自然言語処理」
  NLPの認知度を向上, 門戸を広げた本
- 現在, Python 2.5~2.7 で使うことができる
- 気になったら… http://www.nltk.org/

```
pip install nltk
```


3

はじめに: 環境構築

```
$git clone git://github.com/tuxedocat/ss2012.git
$cd ss2012
```

- preprocessor.py, prepchecker.py, feature_extractor.py ss2012_slide, wdiff_prep, README.md が入ります

- お手元のMacbookには...
  - nltk 2.0.1, Python 2.7.2, iPython 0.12が入っています
  - エディタ等はお好きなものをお使いください

大まかな流れ

- 前処理：CLCコーパスを処理しやすいように料理する
- 学習データ・テストデータを作成する
- 素性関数をつくる
- 学習事例をつくる
- MaxentClassifier を学習させる
- テストデータに対して評価を行う
- 考察・エラー分析・改良

5

Cambridge Learner Corpus (CLC)

- Cambridge ESOLが実施している英語試験の回答を集めたもの
  - http://www.cambridgeesol.org/japan/
  - 135,000人の学習者
  - 世界190カ国、130の異なる母語話者
  - 一部のみ自由なアクセスが可能（他はCambridge関係者のみ）

UNIVERSITY OF CAMBRIDGE

CLCのファイルを見てみよう

- 人手で作成されたxmlファイル
- 80種類のエラータグ(例: NS type=“RT”→ 前置詞置換のエ

```
1  <?xml version="1.0" encoding="UTF-8"?>$
2  <learner><head sortkey="TR733*0100*2000*01">$
3    <candidate><personnel><language>Spanish</language><age>21-25</age></personnel><score>33.0</score></candidate>$
4    <text>$
5      <answer1>$
6        <question_number>1</question_number>$
7        <exam_score>4.2</exam_score>$
8          <coded_answer>$
9           <p>Dear Helen Ryan,</p>$
10          <p>I have seen your letter <NS type="RT"><i>for</i><c>about</c></NS> Camp California in the U.S.A and I woul
    d like to travel in July because I am pretty busy with my studies and I am working with my dad in his office now.</p>$
11          <p>First of all, you state in the letter that the accommodation at Camp California is in tents or log cabins
    . I would prefer to stay in <NS type="MD"><c>a</c></NS> <NS type="AGN"><i>tents</i><c>tent</c></NS> because I like bei
    ng in contact with <NS type="UD"><i>the</i></NS> nature <NS type="UN"><i>life</i></NS>.</p>$
12          <p>Secondly, from the list of activities that you <NS type="R"><i>are also state</i><c>provide</c></NS> in t
    he letter I would prefer swimming because when I was a child I <NS type="RV"><i>practised</i><c>did</c></NS> this spor
    t for 8 years. In addition, I would prefer photography too because I love taking photos.</p>$
13          <p>Finally, <NS type="RA"><i>it</i><c>I</c></NS> would be grateful if you could clarify what kind of clothes
     I must <NS type="TV"><i>be taken</i><c>take</c></NS> and how much I will spend during the stay at the Camp.</p>$
14          <p>I look forward to hearing from you</p>$
15          <p>Yours <NS type="DY"><i>sincerelly</i><c>sincerely</c></NS></p>$
16         </coded_answer>$
17     </answer1>$
18     <answer2>$
```

前処理

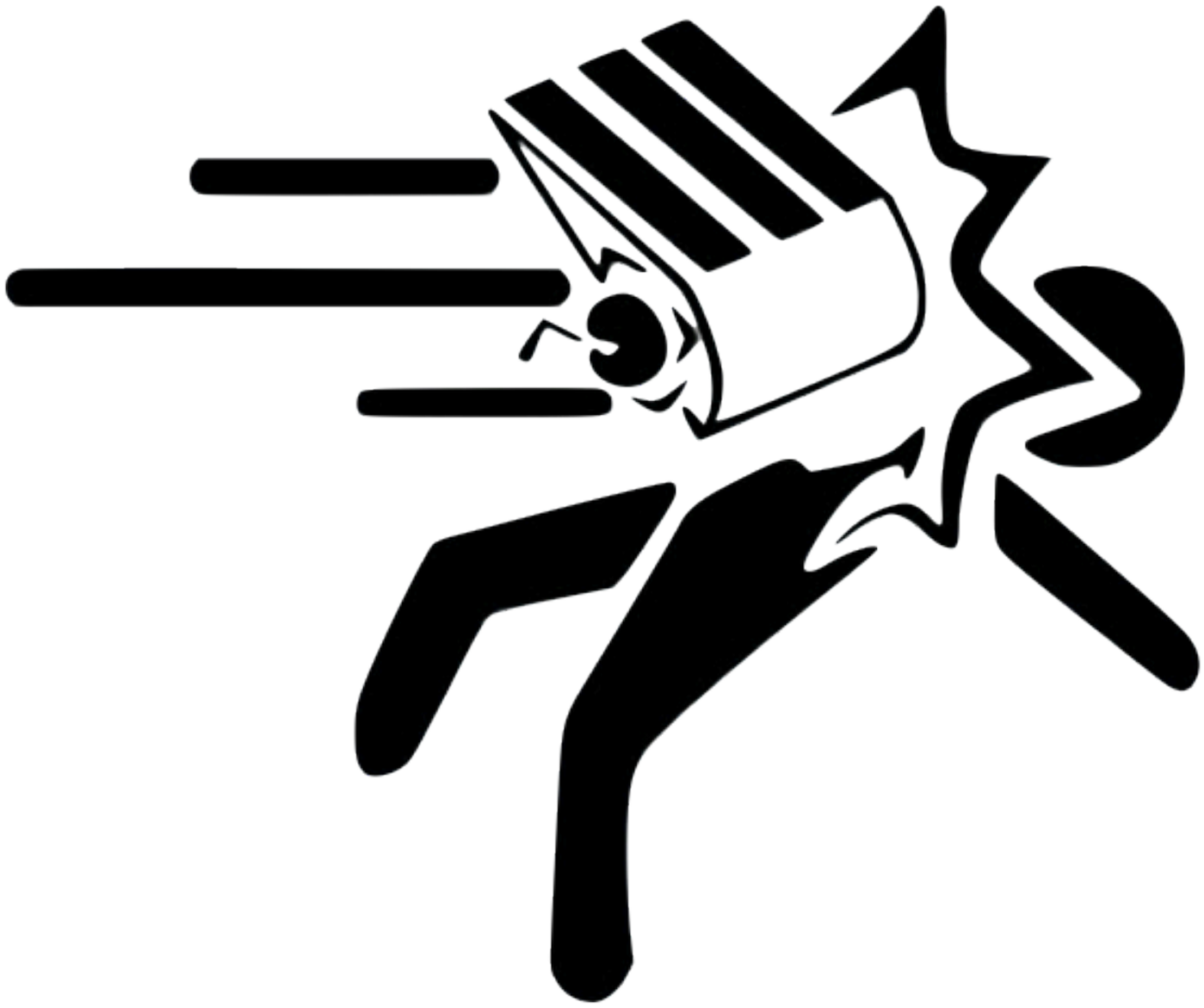
- CLCコーパス（を下ごしらえしたもの：./wdiff_prep ファイル)を読み込み，誤り・訂正情報を抽出する

```
clc = open(“wdiff_prep”, “r”).read()
```

  - これでリストに入る…

- このままでは段落単位なので,文単位に分割する…
  - NLTKには簡易文分割関数がある(nltk.sent_tokenize)
- 文(文字列型)から正規表現を用いて訂正情報などを抽出する
- 後で処理しやすいように整形する…

8



これが意外とめんどくさい

／(^o^)＼ﾅﾝﾃｺｯﾀｲ

...前処理はこちらでやります

```
$python ./preprocessor.py
```

- ”packedcorpus.pkl” が出力されます.
  - 今回はコーパスが小さく処理も簡潔なため Pythonのpickleモジュールを使いました.

- ところで...
  - これは実はほんとうの意味での下拵えではないのです...
  - コーパスの整備や前処理はとても大事な工程です!!!!!!
  - ここで現役NLPerの諸氏から色々なエピソードが語られます・・・

10

コーパス前処理職人の朝は早い

- 「まぁ、好きで始めた仕事ですからね。」

  CLC-FCEコーパスのXMLを眺めながらつぶやく。

  もちろんエディタはVimだ。

- 「なんでこんな間違いをするんや、とか思っても言っちゃダメなんだよ。みんながんばってるからね。」

  膨大な量の言語交流SNSからのデータから

  金脈を見つけ出すのは一流の職人の仕事である。

- 「このデータを使った論文がAcceptされたことを聞く度に、この仕事をしていてよかったと感じるんです。」

コーパス読み込み

- シリアライズされたPython objectを読み込むには...
  - Pickle moduleを使う

```
import cPickle as pickle
corpus = pickle.load(“packedcorpus.pkl”, “rb”)
```

- 一文に対応するdictが入っているlist
  - key “training_words”, “test_words”
    value = [“tokenized”, “sentence”, …]
  - key “correction_pair”, value = (“incorrect prep.”, “correct prep.”)
  - key “ppindex”,
    value = <index of target preposition in tokenized sentence>

12

ここからの流れ

- コーパスを読み込む
- 素性関数をつくる ← これから
- 学習事例をつくる
- Classifierを学習させる
- テストをする
- 結果を眺める→エラー分析

13

Supervised Classification

nltk.MaxentClassifierによる
最大エントロピー法分類器
を用いた教師あり分類問題
…としての前置詞訂正

最大エントロピー法：概略

- エントロピー: 不確実さ、乱雑さの度合い。
  - $$H(p) = \sum -P(x) \log P(x)$$
  - 不確実な状態ほど、エントロピーは高くなる
- 最大エントロピーの原則: 与えられた制約の中で、エントロピーを最大化するモデルを選ぶ。
- 例1：Mさんのじゃんけんパターンをモデル化したい。
  - 制約

(1) P(グー) + P(チョキ) + P(パー) = 1, (2) P(グー) = 0.5

  - 最大エントロピーの原則に従うとすると、どちらのモデルが妥当か。

(ア) P(パー) = 0.1, P(チョキ) = 0.4, P(グー) = 0.5

(イ) P(パー) = 0.25, P(チョキ) = 0.25, P(グー) = 0.5

15

最大エントロピー法：例題

- 例2: Nさん、Lさん、Pさんが、次のように文を書いたとする。
  - Nさん: d_1(block, scissors, block), d_2(paper, scissors, paper)
  - Lさん: d_3(paper, scissors, paper), d_4(block, paper, scissors)
  - Pさん: d_5(block, block, paper), d_6(paper, scissors, block)
- 今、d_x(block, scissors, paper)という文が来たとき、Nさん、Lさん、Pさんによって書かれた確率はそれぞれいくらだろうか。

16

最大エントロピー法：例題

- 素性(特徴)ベクトル**x**とそのラベル(クラス)yを考えてみる
  - 例: **x**(x1,x2,…x9)を次のように定義する

x1 = 単語 “block” が1単語目に出現
x2 = 単語 “block” が2単語目に出現
x3 = 単語 “block” が3単語目に出現
x4 = 単語 “paper” が1単語目に出現
x5 = 単語 “paper” が2単語目に出現
x6 = 単語 “paper” が3単語目に出現
x7 = 単語 “scissors” が1単語目に出現
x8 = 単語 “scissors” が2単語目に出現
x9 = 単語 “scissors” が3単語目に出現

d_1 = (1,0,1,0,0,0,0,1,0), label y: Nさん
d_2 = (0,0,0,1,0,1,0,1,0), label y: Nさん
d_3～d_6を素性ベクトルとラベルで現してみよう。

17

最大エントロピー法：分類・まとめ

- 素性とラベルのペアをもとに、新しい文のラベルを推定する。

$$\overset{*}{\Pr} = \arg\max_{\Pr} H(\Pr) = -\arg\max_{\Pr} \sum_{\mathbf{x},y} \hat{\Pr}[y] \Pr[y|\mathbf{x}] \log \Pr[y|\mathbf{x}]$$

- ※計算は省略しますが気になる人は以下を参考にしてください。
  - 言語処理のための機械学習入門　(高村大也　著)
  - 確率的言語モデル（北研二　著）
  - 朱鷺の杜 (http://ibisforest.org/index.php?最大エントロピー)
- 先ほどの例だと、

  $\Pr[N|d_x] = 0.388,\ \Pr[L|d_x] = 0.302,\ \Pr[P|d_x] = 0.308$

  となり、Nさんによって書かれた可能性が高いことがわかる
- 素性の定義がとても重要

18

nltk.MaxentClassifier

- NLTKには各種の機械学習アルゴリズムが含まれている
  - `nltk.classify.*`
    標準化された素性フォーマット, 簡易的な評価が行える
  - 今回は最大エントロピー法分類器（nltk.MaxentClassifier）を用いる
- MaxentClassifierは教師あり分類器である
  - 正解ラベルと素性集合が学習データとして必要
  - 学習事例（Training Examples）はラベルと素性集合の組
  - CLCコーパスは訂正情報付き！正解ラベルはある
  - となると、残るはどのような素性を使うか...

19

nltk.MaxentClassifierのための学習事例

- nltk.classifyが扱える事例のフォーマット

```
training_set =
[({“feature”:value, “feature”:value,...}, label),
 ({“feature”:value, “feature”:value,...}, label),
...]
```

- 素性集合: {“feature”:value, “feature”:value,...}
- 前置詞の正しさにはどんな素性(特徴量)が関係しているか...
  - 前後の文脈(単語)や品詞は何なのか
  - 他にもっと良い素性は無いのだろうか...

20

素性を抽出する関数

- ある入力に対する素性を抽出する関数をつくる
  - 文中の前置詞に対して, その前後の数単語をとらえたり
    それらの品詞を取得したり…
- 簡単な素性関数(前置詞直後の単語)

```
def successor(words, ppindex):
    return {“succ_%s”%(words[ppindex + 1]): 1}
```

```
words = [“The”, “cat”, “on”, “the” , “box”, “is”…]
ppindex = 2

feature:
{“succ_the”:1}
```

- 今回は二値素性(on/off)のみを用いる

21

素性抽出関数の例: 品詞素性

- NLTKには各種POS Taggerが含まれている
- 今回はPennTreeBankで学習したnltk.pos_tagを使う
- 使用例：
  - sentence: str, words: list

```
words = nltk.word_tokenize(sentence)
nltk.pos_tag(words)
```

  - 入力は単語単位のリスト(nltk.word_tokenize(sentence))

```
words = [“The”, “cat”, “is”, …]
tagged = nltk.pos_tag(words)

tagged:
[(“The”, “DT”), (“cat”, “NN”),…]
```

- feature_extractor.py を参考に作ってみよう

22

評価の前に...

- 学習データとテストデータはあらかじめ分けておく
  - 「正しい評価」をおこなうにはどのような状態が必要だろうか
  - テストデータが学習データに含まれると何がまずい？
- 評価尺度
  - 正解率（accuracy）を評価尺度とする

```
accuracy =
(num. of correct output) / (num. of input instances)
```


23

分類器の学習・テストのしかた

- 学習

```
def train(numexamples = <xxx>, numiter = <xxx>):
    trainset_features = <a list of dictionaries>
    trainset_labels = <a list of strings>
    trainset = zip(trainset_features, trainset_labels)
    classifier =
    nltk.MaxentClassifier.train(trainset,
                                algorithm="IIS", max_iter=numiter)
```

- テスト

```
def test(classifier):
    testset_features = <a list of dictionaries>
    testset_labels   = <a list of strings>
    testset = zip(testset_features, testset_labels)
    accuracy = nltk.classify.accuracy(classifier, testset)
```

休憩？

学習には時間がかかります。

終わらないかも・・・

評価: そのモデルで大丈夫?

- Accuracyはどうでしたか？
- MaxentClassifierは多クラス分類器
  - 本当は “at” だったのに “in” だと出力した場合と “on” だと出力した場合はどちらが多いんだろう...

    分類器が間違いやすいケースを特定したい！

  - そこで混同行列(Confusion Matrix)！

```
classifier_outputs =
    [nltk.MaxentClassifier.classify(t) for t in testset_features]
cm = nltk.ConfusionMatrix(gold_labels, classifier_outputs)
print cm
```


26

より良いモデルを求めて...

- どの素性が精度に寄与しているのか調べたい...

```
print nltk.MaxentClassifier.show_most_informative_features(n = 10)
```

  - 上位10個の素性とそのラベルが表示される

- シンプルな素性では正しい文の特徴を捉えられないときがある
  - 誤り事例や，元のコーパスをもう一度眺めてみる...そして考える
  - 単語の活用形ではなく，原形を入れてみてはどうだろうか
  - より広い範囲の構文的な情報を取り込むとどうなるだろうか
  - 誤りの傾向は？素性に取り込むにはどうするか・・・

27

おつかれさまでした！

また会う日まで・・・