I256:
Applied Natural Language Processing

Marti Hearst
Week 5

Using Context in Tagging

- If only it were easier to say:
  - If “to” is followed by NN, then make its tag IN
  - If “to” is followed by VB, then make its tag TO
- It’s hard to encode this in the ngram tagger since you don’t know what tag you’re going to assign to the following word.
- This type of intuition inspired the **Brill tagger**

Transformation-Based Tagging

- Create an initial assignment
- Go back and make changes if needed
- Iteratively apply a sequence of transformation rules

TBL Templates

```
Change tag a to tag b when:
   w-1 (w+1) is tagged z
   w-2 (w+2) is tagged z
   w-1 or w-2 is tagged z
   etc
```

Non-Lexicalized

```
Change tag a to tag b when:
   w-1 (w+1) is foo
   w-2 (w+2) is bar
   w is foo or w-1 is bar
   etc
```

Lexicalized

TBL Example Rules

He/PRP is/VBZ as/IN tall/JJ as/IN her/PRP$

`Change from IN to RB if w+2 is as`

He/PRP is/VBZ as/RB tall/JJ as/IN her/PRP$

He/PRP is/VBZ expected/VBN to/TO race/NN today/NN

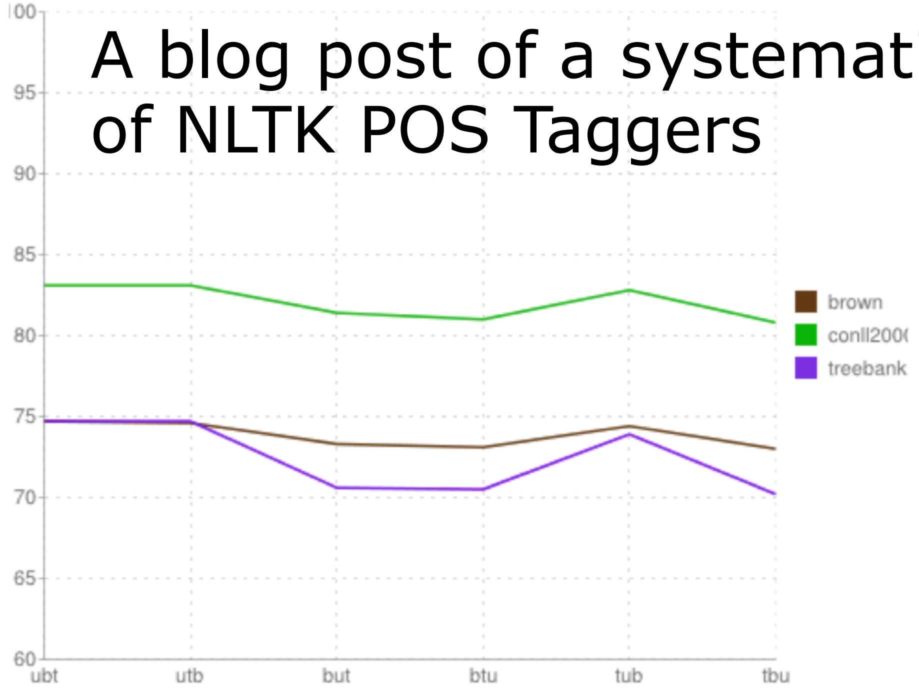
`Change from NN to VB if w-1 is tagged as TO`

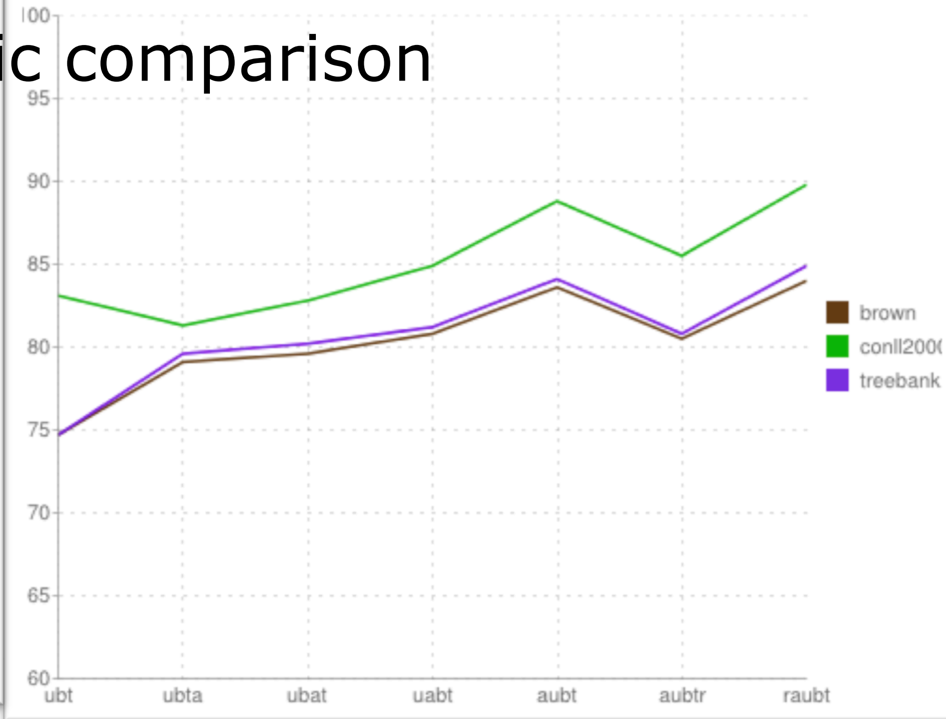
He/PRP is/VBZ expected/VBN to/TO race/VB today/NN

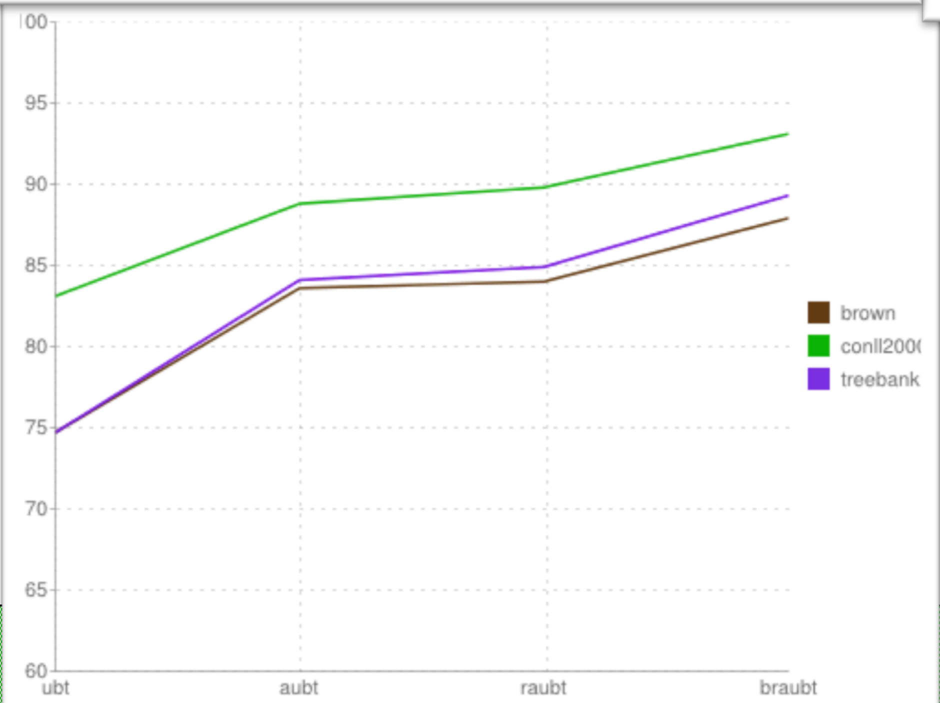
Transformation-Based Tagging

- Rule-based, but data-driven
  - No manual knowledge engineering!
- Training on 600k words, testing on known words only
  - Lexicalized rules: learned 447 rules, 97.2% accuracy
  - Early rules do most of the work:
    - 100 → 96.8%, 200 → 97.0%
  - Non-lexicalized rules: learned 378 rules, 97.0% accuracy
- How good is it?
  - Baseline: 93-94%
  - Upper bound: 96-97%
  - Statistical with 1M words of training data: 96.7%

A blog post of a systematic comparison of NLTK POS Taggers



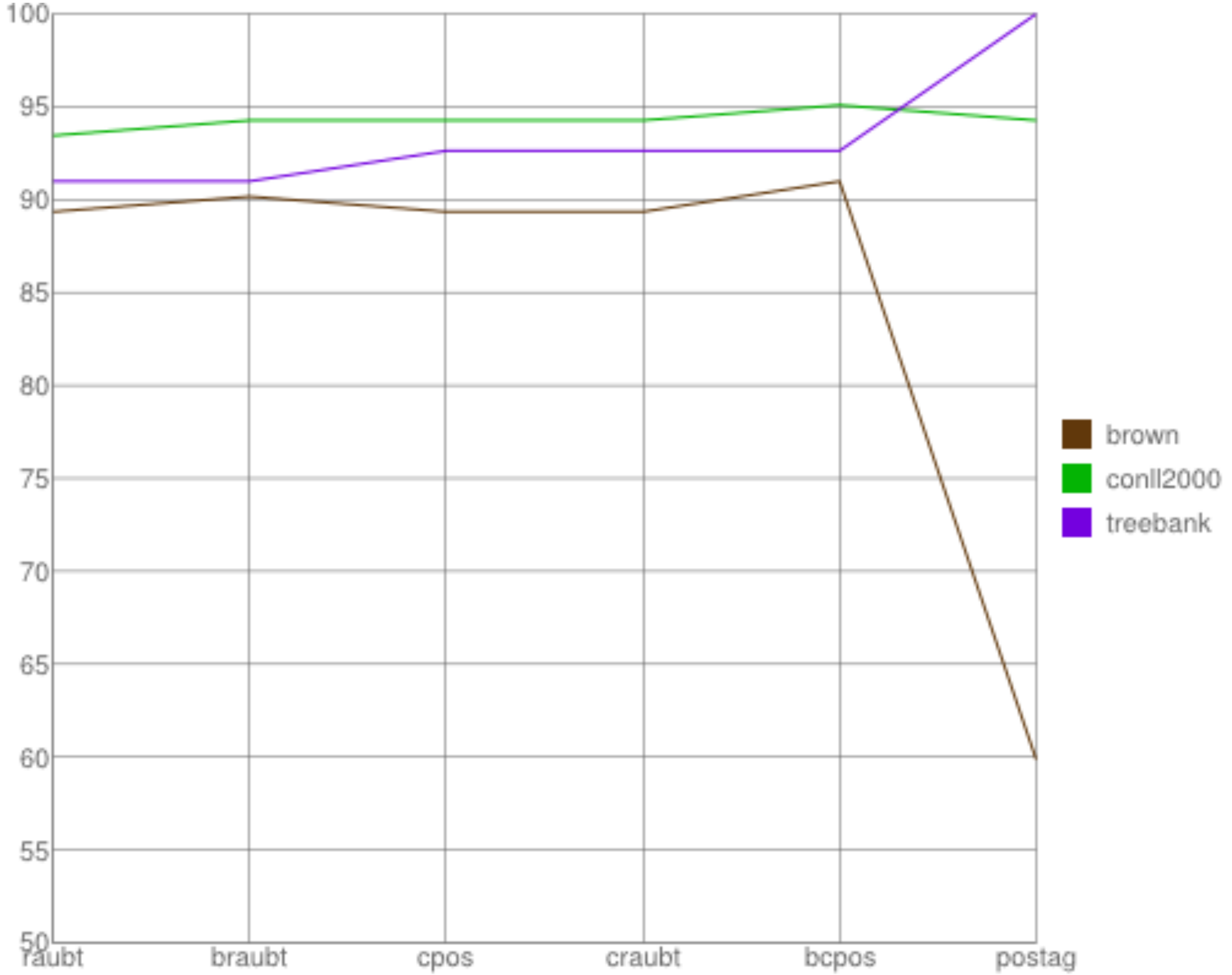




ubt = unigram,bigram,trigram
a = affix, r = regex
second b = brill

very small training sets ~3000 words

http://streamhacker.com/2008/11/03/part-of-speech-tagging-with-nltk-part-1/



ubt = unigram,bigram,trigram
a = affix, r = regex
second b = brill
cpos = NaiveBayes
bcpos = brill + cpos
Pos_tag is nltk’s MaxEnt tagger

98.08% accuracy on Treebank

Not enough training data for ML in brown corpus as used here

http://streamhacker.com/2008/11/03/part-of-speech-tagging-with-nltk-part-1/

Other NLTK Taggers

- Not described in the book, but are described in the blog post:
  - nltk.tag.sequential.ClassifierBasedPOSTagger
  - nltk.tag.sequential.AffixTagger
- To find out more:
  - help(nltk.AffixTagger)

What we covered about tagging

- What are parts of speech
- What is POS tagging
- Methods for automatic POS tagging
  - N-gram (language model) based
  - Rule-based
  - Transformation-based learning
- Starting to see:
  - Evaluation
  - Supervised machine learning

Language Models

- What they are
- Why they are important
- Issues for counting words

Models of word sequences

- Pay attention to the preceding words
  - “Let’s go outside and take a [____]”
    - walk: very likely
    - break: quite likely
    - stone: less likely
- Compute conditional probability as:
  - P(walk | let’s go outside and take a)

Why Language Models?

- POS Tagging:
  - P(n follows det) > P(v follows det)
- Spelling Correction
  - The office is about fifteen **minuets** from my house
    - P(about fifteen minutes from) > P(about fifteen minuets from)
- Speech Recognition
  - P(I saw a van) >> P(eyes awe of an)
- Summarization, question answering, etc etc

What is Language Modeling?

- Goal: compute the probability of a sentence or a sequence of words:
  - $P(W) = P(w_1,w_2,w_3,w_4,w_5 \ldots w_n)$
- Related task: probability of an upcoming word:
  - $P(w_5|w_1,w_2,w_3,w_4)$
- A model that computes either of these two:
  - $P(W)$ or $P(w_n|w_1,w_2 \ldots w_{n-1})$
- is called a **language model**.
  - (Jurafsky thinks a better term is **a grammar**, but LM is standard.)

How to Compute This

The Chain Rule From Probability Theory

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

P("its water is so transparent") =

P(its) × P(water|its) × P(is|its water)

× P(so|its water is) × P(transparent|its water is so)

Probability of a Word Sequence

$$P(\text{the} \mid \text{its water is so transparent that}) =$$

$$\frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

- Problem?
- Too many possible sentences! Not enough data to make good estimates.
- Solution: approximate the probability of a word given all the previous words.

N-gram Approximations

- The Markov Assumption:
  - The probability of a future event depends only on a **limited** history of preceding events.
  - Probability of next word depends only on the prev word.
- Bigram model:
  - Only look at the **preceding** word; multiply these probabilities
- Trigram model:
  - Only look at the **two** preceding words

Simplist Case: Unigram Model

- Just multiply the probability of each word to get the probability of a sentence.
- Simplist way to estimate the probability of the word:
  - 1/frequency in the collection

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

Bigram Model

- Condition on the previous word
- Just look at the **one word** that came before
- Here is the probability of word i given words 1 through word i-1:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

N-gram Models

- We can extend to trigrams, 4-grams, etc
  - Often have a data sparcity issue
  - Works better now that we have “big data”
- This is an insufficient model of language because it has long distance dependencies
- BUT it does do a decent job a lot of the time.

Estimating Bigram Probabilities

The Maximum Likelihood Estimate

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

(Wi-1,Wi) means the words occur consecutively in the text

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

EXERCISE: Compute:

- P(Sam | <s>) =
- P(Sam | am) =
- P(am | I) =
- P(I | <s>) =

Do it in the workbook!

Estimating Bigram Probabilities

The Maximum Likelihood Estimate

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

(Wi-1,Wi) means the words occur consecutively in the text

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$

$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$

More Examples: The Berkeley Restaurant Project

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Raw Bigram Counts

Out of 9222 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Bigram Probabilities in Berkeley Restaurant Data

- After normalizing with unigrams
  - P(chinese|want) = .0065
  - P(to|want) = .66
  - P(eat | to) = .28
  - P(food | to) = 0
  - P(want | spend) = 0
  - P (i | <s>) = .25

Google N-Gram Release, August 2006

AUG
3

All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

```
File sizes: approx. 24 GB compressed (gzip'ed) text files

Number of tokens:     1,024,908,267,229
Number of sentences:     95,119,665,584
Number of unigrams:          13,588,391
Number of bigrams:          314,843,401
Number of trigrams:         977,069,902
Number of fourgrams:      1,313,818,354
Number of fivegrams:      1,176,470,663
```

Google N-Gram Release

- Examples of trigrams
  - ceramics collectables collectibles 55
  - ceramics collectables fine 130
  - ceramics collected by 52
  - ceramics collectible pottery 50
  - ceramics collectibles cooking 45
  - ceramics collection , 144
  - ceramics collection . 247
  - ceramics collection </S> 120
  - ceramics collection and 43
  - ceramics collection at 52
  - ceramics collection is 68
  - ceramics collection of 76

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html