Parts of Speech

Dan Smith

University of East Anglia

POS Taggers: overview

- What do they do?
- How do they work?
- How accurate are they?
- What can we achieve using taggers?
- Examples

Uses of POS tagging

- Limits the range of meanings
- Helps in stemming
- Limits choice of words in speech recognition
- Helps identify terms for IR
- Makes natural language parsing easier
- Makes language pattern recognition easier

What does a tagger do?

- Looks at each word in a sentence.
- Assigns tag to each word.
  - For example: *The man saw the girl.*

the-DET man-NN saw-VPAST the-DET girl-NN

5

Parts of Speech (English)

- Closed classes:
  - **prepositions**: on, under, over, by, with, ...
  - **determiners**: a, an, the
  - **pronouns**: he, she, it, I, we, they, ...
  - **conjunctions**: and, but, or, ...
  - **particles**: up, down, on, off, in, ...
  - **numerals**: 1, 2, 3, ..., first, second, third, ...
  - **auxiliary verbs**: can, may, should, are, ...

6

... more POS

- Open classes
  - **nouns**: people, places, things, ideas
  - **verbs**: actions or processes
  - **adverbs**: modifiers of verbs
  - **adjectives**: describe properties or qualities

Tagsets

- Many different tagsets devised
- Most common for English
  - Penn Treebank, 36 tags
  - Brown corpus tagset, 87 tags
  - C5 (U. Lancaster, CLAWS), 61 tags
  - C7, 146 tags

Penn treebank tags

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd p. singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

9

Users of POS tagging

- NLP researchers
  - using tagging for grammar induction, …
  - need fine-grained tagging
- Users of NLP information
  - using tagging to determine context, meaning, …
  - only interested in major categories

10

Universal POS tags

- Most languages can be tagged with 12 coarser tags

1.	NOUN	nouns
2.	VERB	verbs
3.	ADJ	adjectives
4.	ADV	adverbs
5.	PRON	pronouns
6.	DET	determiners and articles
7.	ADP	prepositions and postpositions
8.	NUM	numbers
9.	CONJ	conjunctions
10.	PRT	particles
11.	.	punctuation
12.	X	anything else (abbreviations, foreign words, …)

(Petrov et al. 2012)

11

Universal vs. specific tagsets

- POS taggers work better with larger tagsets
  - they provide more context
- Applications using POS are often better with universal tags
  - the detail is unhelpful and/or confusing
- Best accuracy if tag with detailed tagset and then map into universal tagset
- Big advantage of universal tagset is that it works well across all major languages

12

The tagging process

- Similar to tokenization of computer languages
- Input:
  - a tagset
  - a string of words
  - dictionary of possible tags for each word
- Output:
  - a string of words each marked with the most likely tag
- Problem is multiple meanings
  - "... book a flight"
  - "... book on the desk"

13

Tag ambiguity

- Brown corpus
- 11.5% of word types ambiguous, but
- 40% of tokens are ambiguous
- Words with
  - 1 tag 35,340
  - 2+ tags 4,100

DeRose S. (1988) Grammatical category disambiguation by statistical optimization, *Comp. Ling.,* 21(2), 31-39

14

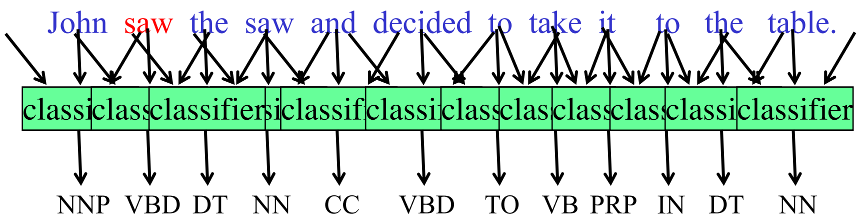
Tagging algorithms

- Rule-based
- Stochastic
- Transformation-based

Tagging algorithms are approximately *O(n)* complexity

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



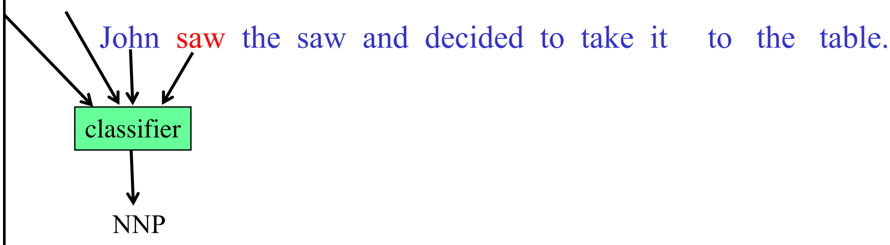
15

Sequence Labeling as Classification Using Outputs as Inputs

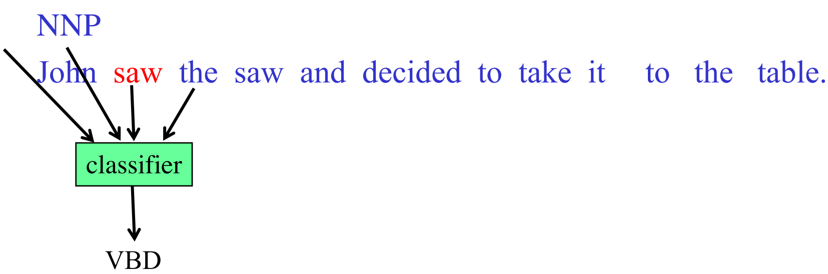
- Better input features are usually the categories of the surrounding tokens, but these are not available yet
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output

16

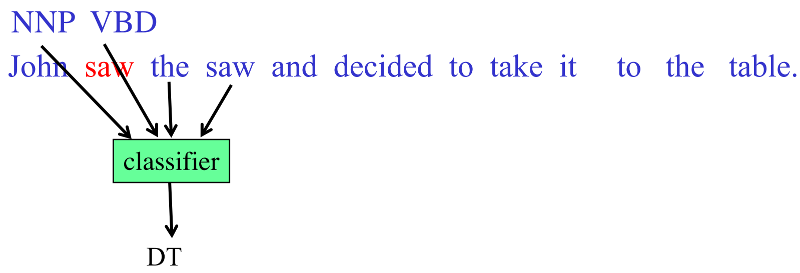
Forward Classification



Forward Classification

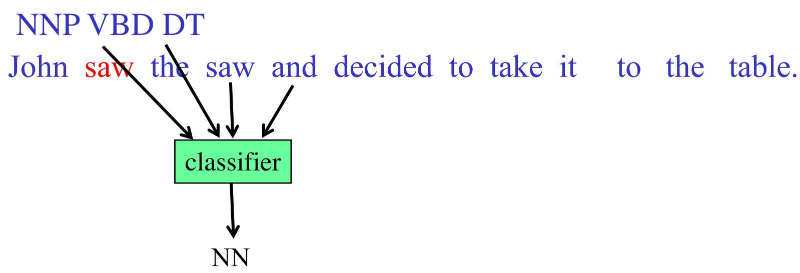


Forward Classification



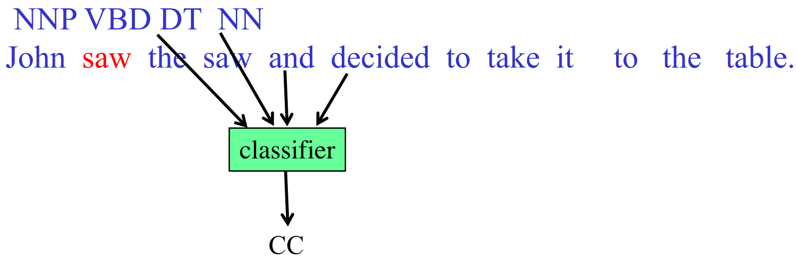
19

Forward Classification



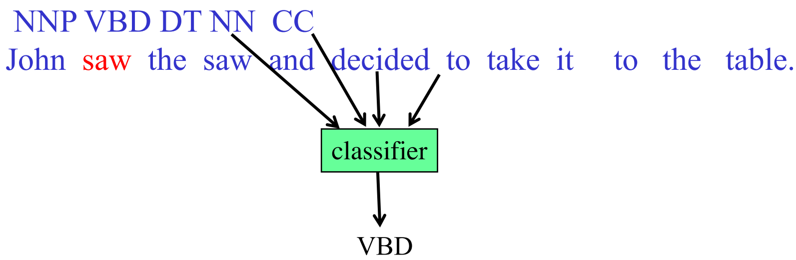
20

Forward Classification



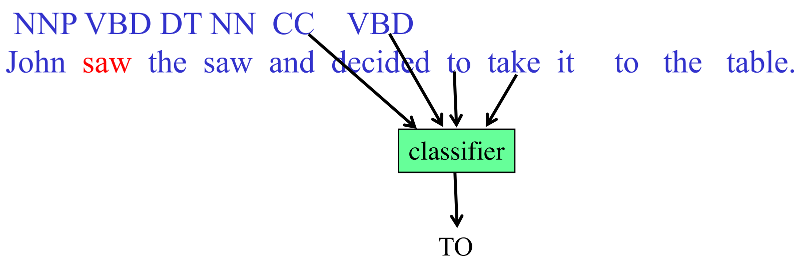
21

Forward Classification



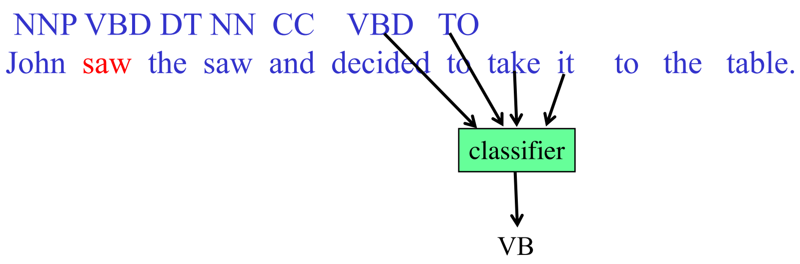
22

Forward Classification



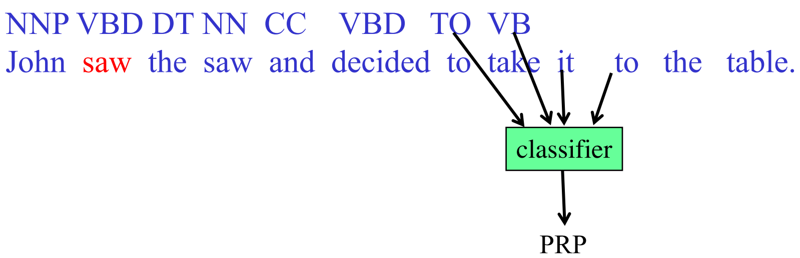
23

Forward Classification



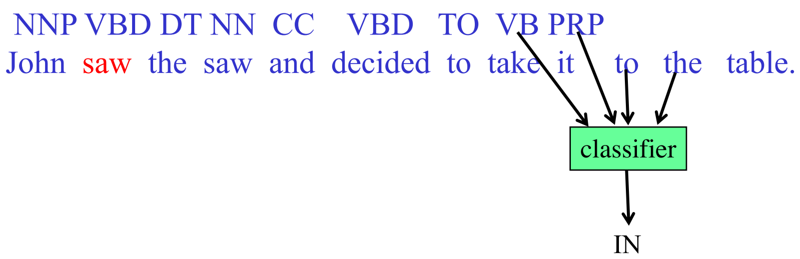
24

Forward Classification



25

Forward Classification

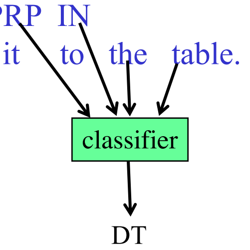


26

Forward Classification

NNP VBD DT NN CC VBD TO VB PRP IN

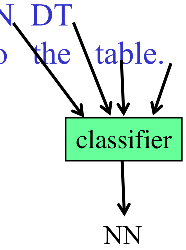
John saw the saw and decided to take it to the table.



Forward Classification

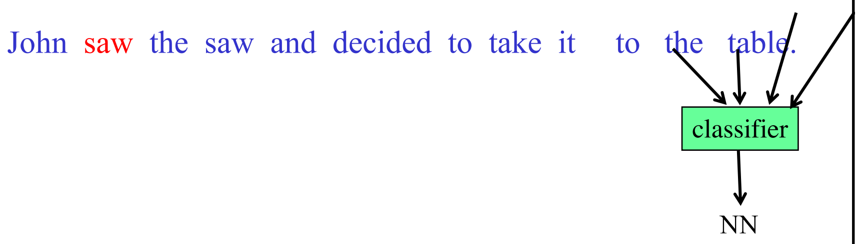
NNP VBD DT NN CC VBD TO VB PRP IN DT

John saw the saw and decided to take it to the table.



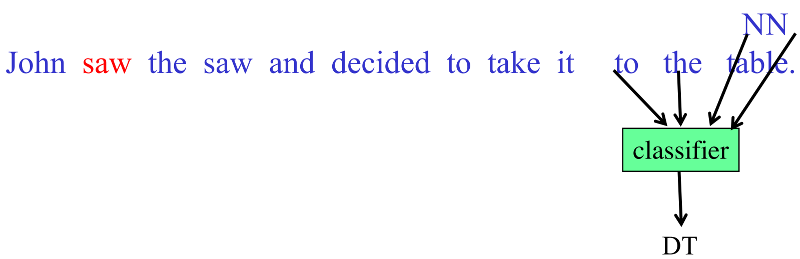
Backward Classification

- Disambiguating "to" in this case would be even easier backward.



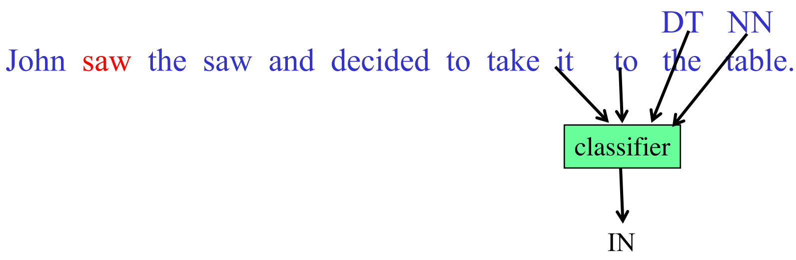
Backward Classification

- Disambiguating "to" in this case would be even easier backward.



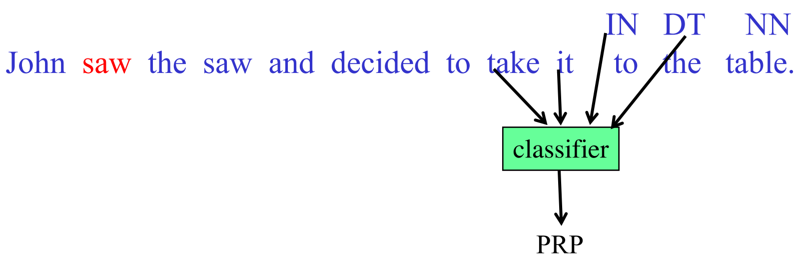
Backward Classification

- Disambiguating "to" in this case would be even easier backward.



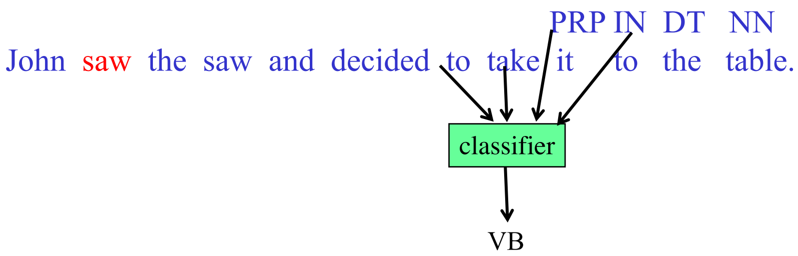
Backward Classification

- Disambiguating "to" in this case would be even easier backward.



Backward Classification

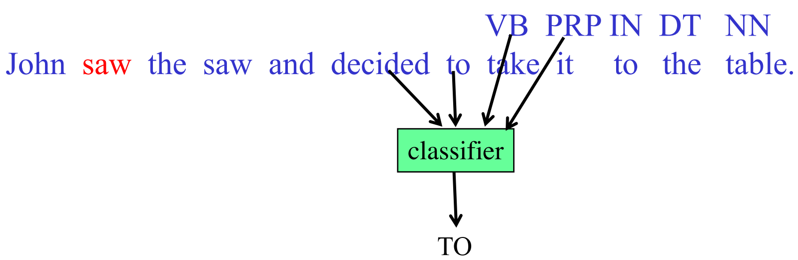
- Disambiguating "to" in this case would be even easier backward.



33

Backward Classification

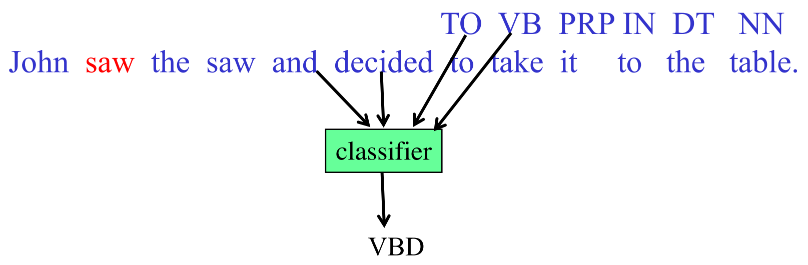
- Disambiguating "to" in this case would be even easier backward.



34

Backward Classification

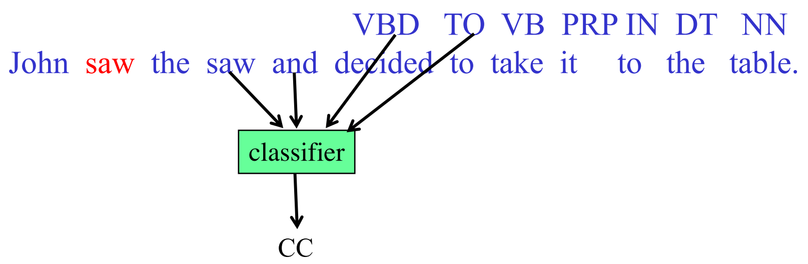
- Disambiguating “to” in this case would be even easier backward.



35

Backward Classification

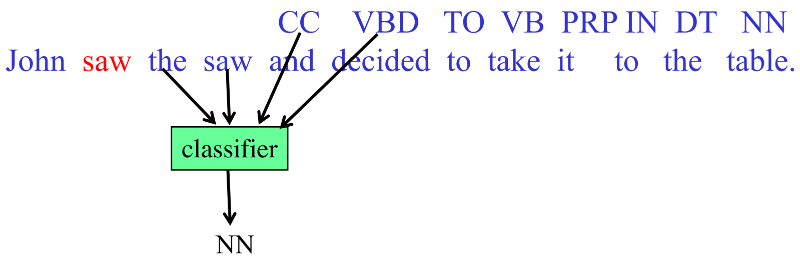
- Disambiguating “to” in this case would be even easier backward.



36

Backward Classification

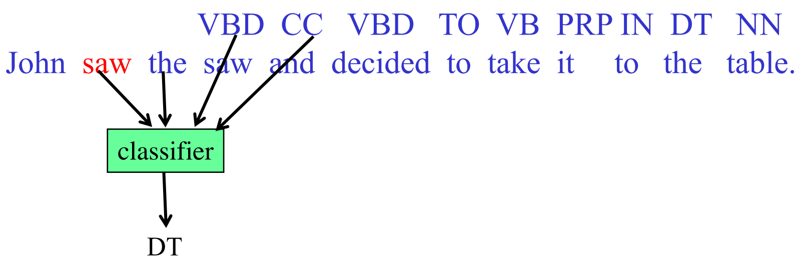
- Disambiguating “to” in this case would be even easier backward.



37

Backward Classification

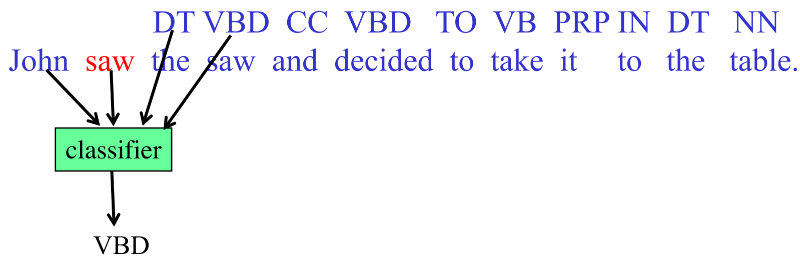
- Disambiguating “to” in this case would be even easier backward.



38

Backward Classification

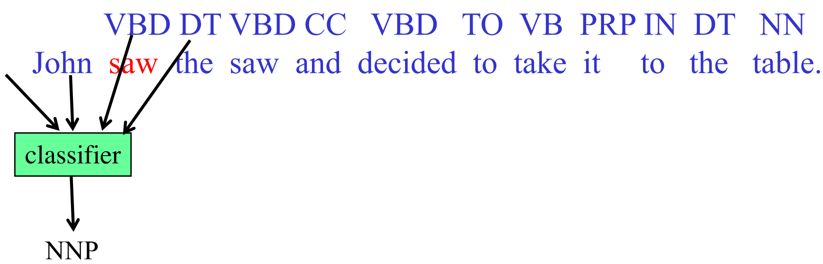
- Disambiguating "to" in this case would be even easier backward.



39

Backward Classification

- Disambiguating "to" in this case would be even easier backward.



40

Problems with Sequence Labeling as Classification

- Not easy to integrate information from category of tokens on both sides.
- Difficult to propagate uncertainty between decisions and "collectively" determine the most likely joint assignment of categories to all of the tokens in a sequence.

41

42

Rule-based tagging

1. Use dictionary to assign list of possible tags to each word
2. Apply hand-written rules to eliminate infeasible tag sequences

Can achieve 99% word-level accuracy, but 95% more typical

43

Rule-based tagging: example

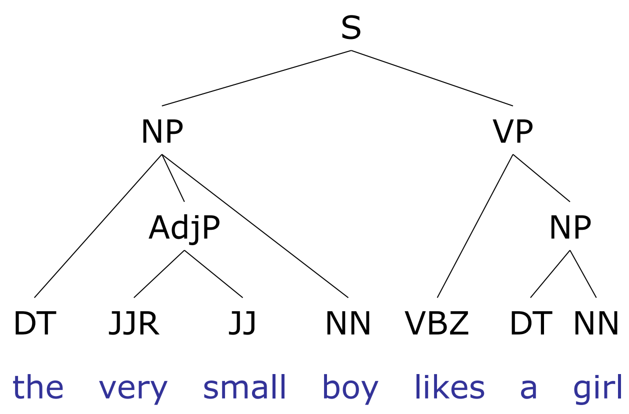
- Rules

  S → NP VP
  VP → V NP
  NP → Det N | AdjP NP
  AdjP → Adj | Adv AdjP
  N → boy | girl
  V → likes
  Adj → small
  Adv → very
  Det → a | the

The very small boy likes a girl

44

Tagging for sentence parsing



45

STOCHASTIC TAGGING

46

Stochastic tagging

- Uses training corpus
- Simple aim is to maximise
  P(tag | word)
- More realistically, maximise
  P(word | tag) * *P*(tag | previous *n* tags)
  - most approaches use Viterbi algorithm
  - prunes search tree using Maximum Likelihood Estimates
- HMM most popular, combines these

HMM tagging

- Bigram HMM tagger chooses most probable tag t_i for word w_i given tag t_{i-1}

$$t_i = \underset{j}{\operatorname{argmax}}\ P(t_j \mid t_{i-1}, w_i)$$

(i.e. "the t_j such that $P(t_j\ldots)$ is maximised")

- Basic HMM for a single tag:

$$t_i = \underset{j}{\operatorname{argmax}}\ \underbrace{P(t_j \mid t_{i-1}, w_i)}_{\text{tag sequence probability}}\ \underbrace{P(w_i \mid t_j)}_{\text{word likelihood}}$$

TRANSFORMATION-BASED TAGGING

Transformation-Based Tagging

- Commonly called Brill tagging
- Hybrid approach
  - uses rules
  - supervised learning from pre-tagged training set
- Rules are applied in broadest to narrowest order

TBL rules

- Tag frequency:
  - p(NN | race) = 0.98
  - p(VB | race) = 0.02
- Text:
  - Baxter is expected to race tomorrow
  - The race for outer space will probably be set back because of the global financial crisis

51

Tagging results

- First, apply most likely tag:
  - Baxter/NNP is/VBZ expected/VBN to/TO race/NN tomorrow/NN
  - People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN
- Next, apply narrower rules, including:
  - *Change NN to VB when previous tag is TO*
- This gives:
  - Baxter/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NN

52

TBL learning

- Given a correctly tagged training set
- Label every word with most likely tag
- Apply rules
  1. Examine all transformations
  2. Select the one giving most improved tagging
  3. Re-tag using this rule
- Repeat until improvement is insufficient to justify continuing
- Gives ordered list of rules

53

TBL rules

- Based on templates:
- "Change tag **a** to tag **b** when ...
  - the previous (following) word is tagged **z**
  - the word two before (after) is tagged **z**
  - one of the two previous (following) words is tagged **z**
  - one of the three previous (following) words is tagged **z**
  - the previous word is tagged **z** and the following word is tagged **w**
  - the previous (following) word is tagged **z** and the word two before (after) is tagged **w**
- The variables **a, b, w, z** range over the parts of speech

54

Unknown words

- Unknown words always occur
- Proper nouns and acronyms are most common new words
  1. Can assume all tags equally probable – hard work
  2. Better to assume distribution is similar to singleton words
  3. Best to use rules:
     - words ending in -s are most probably NNS
     - words ending in -ed are most probably VBN
     - capitalised words are most probably NNP
- Brill used set of orthographic templates in TBL to induce rules

55

POS tagger accuracy

- Baseline:
  - most common tag for a given word
  - accuracy c. 91% (Brown corpus)
- Taggers achieve word accuracy rates of 95-99%
  - Vary according to text/type/genre
    - Of pre-tagged corpus
    - Of text to be tagged
- Worst case scenario: assume success rate of 95%
    - Prob(one-word sentence) = .95
    - Prob(two-word sentence) = .95 * .95 = 90.25%
    - Prob(ten-word sentence) = 59% approx

56

Tagger evaluation

- Evaluate accuracy against baseline of "give every word its most common tag"
- Look at errors
  - a confusion matrix is the best summary of where the errors occur

	VB	TO	NN
VB			
TO			
NN			

57

NLTK

POS TAGGING IN PYTHON

58

NLTK overview

- NLTK is a collection of Python resources and corpora for natural language processing
  - Available for download
    - the whole collection is about 7GB
    - better to download only the bits you need
- Main tools
  - tokenizers
  - POS taggers
  - parsers
  - classifiers
  - clustering
  - …

59

POS tagging with NLTK

- Simple example

```
import nltk
s = "Really, Dinah ought to have taught you better manners!”
st = nltk.word_tokenize(s)
sp = nltk.pos_tag(st)
```

```
sp =>
[('Really', 'RB'), (',', ','), ('Dinah', 'NNP'), ('ought', 'MD'), ('to', 'TO'), ('have', 'VB'), ('taught', 'VBN'), ('you', 'PRP'), ('better', 'JJR'), ('manners', 'NNS'), ('!', '.')]
```

60

A more difficult example

```
s = 'He skis on short skis. She skis on long old skis’
st = nltk.word_tokenize(s)
sp = nltk.pos_tag(st)
```

```
sp =>
[('He', 'PRP'), ('skis', 'VBZ'), ('on', 'IN'), ('short', 'JJ'),
 ('skis', 'NN'), ('.', '.'), ('She', 'PRP'), ('skis', 'VBD'),
 ('on', 'IN'), ('long', 'JJ'), ('old', 'JJ'), ('skis', 'NN')]
```

61

NLTK resources

- http://nltk.org/install.html
- https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/
- http://www.nltk.org/book/ch01.html

62

Resources

- Manning *et al*. 2008, Ch 2
- For an in-depth treatment of these topics:

  Jurafsky D., Martin J. (2000) *Speech and Language Processing*, Prentice Hall

  Slav Petrov, Dipanjan Das, Ryan T. McDonald (2012). A Universal Part-of-Speech Tagset, *LREC 2012*, 2089-2096
- HMM tutorial by Roger Boyle – highly recommended
  http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html
- Resources, including Stanford tagger
  http://www-nlp.stanford.edu/links/statnlp.html

some slides in this presentation from R. J. Mooney, U. Texas Austin