

CSE 842 Natural Language Processing

Lecture 4: Smoothing

1/26/2009

CSE842, Spring 2009, MSU

1

Chain Rule

The probability of a word sequence is the probability of a conjunctive event.

$$P(w_1^n) = P(w_1)P(w_2 | w_1)P(w_3 | w_1^2) \dots P(w_n | w_1^{n-1})$$
$$= \prod_{k=1}^n P(w_k | w_1^{k-1})$$

Unfortunately, that's really not helpful in general. Why?

1/26/2009

CSE842, Spring 2009, MSU

2

Markov Assumption

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-N+1}^{n-1})$$

- $P(w_n)$ can be approximated using only N-1 previous words of context
- This lets us collect statistics in practice
- Markov models are the class of probabilistic models that assume that we can predict the probability of some future unit without looking too far into the past
- Order of a Markov model: length of prior context

1/26/2009

CSE842, Spring 2009, MSU

3

MLE for N-gram

N-gram models can be trained by **counting** and **normalization**

Bigram: $P_{ML}(w_n | w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$

Ngram: $P_{ML}(w_n | w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$

1/26/2009

CSE842, Spring 2009, MSU

4

Log Probability

- Individual probabilities can be small, and the product of many probabilities can be VERY small.
- It is usually best to compute N-gram probabilities in log space, e.g.,
$$\log_2 P(w_1 \dots w_n) = \log_2 \prod P(w_i | w_{i-1})$$
$$= \sum \log_2 P(w_i | w_{i-1})$$
- In the end, the actual probability can be restored by taking the anti-log: $2^{\log(X)} = X$

1/26/2009

CSE842, Spring 2009, MSU

5

Some Useful Empirical Observations

- A small number of events occur with high frequency
- A large number of events occur with low frequency
- You can quickly collect statistics on the high frequency events
- You might have to wait an arbitrarily long time to get valid statistics on low frequency events
- Some of the zeroes in the table are really zeroes. But others are simply low frequency events you haven't seen yet.

1/26/2009

CSE842, Spring 2009, MSU

6

Problem with MLE estimate - Example

Suppose in a corpus, 10 instances with "come across .."

Come across as: 8
Come across more: 1
Come across a: 1

Using MLE estimation:
 $P(\text{as}|\text{come across}) = 0.8$
 $P(\text{more}|\text{come across}) = 0.1$
 $P(\text{a}|\text{come across}) = 0.1$

Any problem?

1/26/2009

CSE842, Spring 2009, MSU

7

Problem with MLE estimate - Example

Suppose in a corpus, 10 instances with "come across .."

Come across as: 8
Come across more: 1
Come across a: 1

Using MLE estimation:
 $P(\text{as}|\text{come across}) = 0.8$
 $P(\text{more}|\text{come across}) = 0.1$
 $P(\text{a}|\text{come across}) = 0.1$
 $P(x|\text{come across}) = 0$ for x not among the above 3 words

The MLE does not capture the fact that there are other words which can follow "come across", e.g., "the", "some", etc, but just do not appear in the training set

1/26/2009

CSE842, Spring 2009, MSU

8

Problem with MLE estimate

While there are a limited number of frequent events in language, there is a seemingly never ending tail to the probability distribution of rarer events, and we can never collect enough data to get to the end of the tail: Zipf's law

Devise better estimators that allow for the possibility that we will see events that we didn't see in the training text

1/26/2009

CSE842, Spring 2009, MSU

9

Discounting Method

Suppose we've seeing the following counts

X	$count(x)$	$P_{ML}(w_i w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$
the	50	
the flower	16	16/50
the dog	14	14/50
the park	8	8/50
the tree	6	6/50
the woman	3	3/50
The path	1	1/50
The pond	1	1/50
The afternoon	1	1/50

The ML estimates are systematically high, particularly for low count items

1/26/2009

CSE842, Spring 2009, MSU

10

Discounting Method

Now define "discounted" counts, $Count^*(x) = Count(x) \cdot 0.5$

New estimates:

X	$count(x)$	$count^*(x)$	$P_{ML}(w_i w_{i-1}) = \frac{C^*(w_{i-1}w_i)}{C(w_{i-1})}$
the	50		
the flower	16	15.5	15.5/50
the dog	14	13.5	13.5/50
the park	8	7.5	7.5/50
the tree	6	5.5	5.5/50
the woman	3	2.5	2.5/50
The path	1	0.5	0.5/50
The pond	1	0.5	0.5/50
The afternoon	1	0.5	0.5/50

1/26/2009

CSE842, Spring 2009, MSU

11

Discounting Method

- We now have some "miss probability mass"

$$\alpha(w_{i-1}) = 1 - \sum_w \frac{Count^*(w_{i-1}, w)}{Count(w_{i-1})}$$

E.g., in our example $\alpha(w_{i-1}) = 4/50$

- Divide the remaining probability mass between word w for which $Count(w_{i-1}, w) > 0$

1/26/2009

CSE842, Spring 2009, MSU

12

Smoothing Techniques

Smoothing: re-evaluate some zero-probability and low-probability N-grams, and assign them non-zero values.

- Laplace
- Good Turing
- Backoff

1/26/2009

CSE842, Spring 2009, MSU

13

Laplace Smoothing

- Also add one smoothing:

- Add 1 to the count
- Original Prob: $P_i = \frac{c_i}{N}$
- Smoothed Prob: $P_i^* = \frac{c_i + 1}{N + V}$

- Discount: Lowering some non-zero counts

$$c^* = (c + 1) \frac{N}{N + V}$$

1/26/2009

CSE842, Spring 2009, MSU

14

Laplace Smoothing

Given a corpus, suppose vocabulary size $V = 15,000$
 "...was...": 3000 times; "...was not...": 608 times

Original: $P(not | was) = \frac{C("was not")}{C("was")} = \frac{608}{3000} = 0.203$

After add one smoothing:

$$P(not | was) = \frac{C("was not") + 1}{C("was") + V} = \frac{608 + 1}{3000 + 15000} = 0.034$$

Any problem?

1/26/2009

CSE842, Spring 2009, MSU

15

Laplace Smoothing

Original counts

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Discounted counts

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

1/26/2009

CSE842, Spring 2009, MSU

16

Laplace Smoothing

- In general, add one smoothing is a poor method of smoothing
- We'd like to find methods that don't change the original counts/probabilities so drastically

1/26/2009

CSE842, Spring 2009, MSU

17

Good-Turing Discounting

- Intuition: use the count of things you've seen once to help estimate the count of things you've never seen.
- Re-estimate amount of probability mass for zero (or low count) Ngrams by looking at Ngrams with higher counts
- Define N_c = number of elements x for which $\text{Count}(x) = c$.
- Modify count for any x with $\text{Count}(x) = c$ and $c > 0$

$$c^* = (c + 1) \frac{N_c + 1}{N_c}$$

- Leading to the following estimate of "missing mass", $\frac{N_1}{N}$ where N is the size of the sample (why? HW assignment)

1/26/2009

CSE842, Spring 2009, MSU

18

Good-Turing Discounting

A corpus of 30000 English words, Suppose a word “spartan” only appears once.

$$P_{ML}(\text{“spartan”}) = 1/30000$$

Suppose there are 10000 different words that appear once and 3000 words that appear twice, what is $P(\text{“spartan”})$ using GT discount?

$$c^* = (1+1) \frac{N_2}{N_1} = 2 * 3000 / 10000$$

$$p^* = \frac{c^*}{N} = \frac{2 * 3000}{10000 * 30000} = 2 \times 10^{-5}$$

1/26/2009

CSE842, Spring 2009, MSU

19

Examples: Bigram Frequencies of Frequencies and GT Re-estimates

AP Newswire			Berkeley Restaurant—		
c (MLE)	N_c	c^* (GT)	c (MLE)	N_c	c^* (GT)
0	74,671,100,000	0.0000270	0	2,081,496	0.002553
1	2,018,046	0.446	1	5315	0.533960
2	449,721	1.26	2	1419	1.357294
3	188,933	2.24	3	642	2.373832
4	105,668	3.24	4	381	4.081365
5	68,379	4.22	5	311	3.781350
6	48,190	5.19	6	196	4.500000

1/26/2009

CSE842, Spring 2009, MSU

20

Complications

- In practice, assume large counts ($c > k$ for some k) are reliable:

$$c^* = c \text{ for } c > k$$

- That complicates c^* , making it:

$$c^* = \frac{(c+1) \frac{N_{c+1}}{N_c} - c \frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}}, \text{ for } 1 \leq c \leq k.$$

- We need the N_k to be non-zero, so we need to smooth (interpolate) the N_k counts before computing c^* from them (Gale and Sampson: Good-turning Frequency Estimation without Tears)
- GT discounting is used in combination with the backoff and interpolation algorithms.

1/26/2009

CSE842, Spring 2009, MSU

21

Backoff and Interpolation

- If we are estimating:
 - trigram $p(z|x,y)$
 - but $\text{count}(xyz)$ is zero
- Use info from:
 - Bigram $p(z|y)$
- Or even:
 - Unigram $p(z)$
- How to combine this trigram, bigram, unigram info in a valid fashion?

1/26/2009

CSE842, Spring 2009, MSU

22

Backoff Vs. Interpolation

- Backoff:** use trigram if you have it, otherwise bigram, otherwise unigram
- Interpolation:** mix all three

1/26/2009

CSE842, Spring 2009, MSU

23

Backoff

- Use N-gram “hierarchy” to “back off” to a lowered order N-gram if there is zero evidence for a higher-order N-gram
- Backoff methods (e.g. Katz)
 - Where trigram unavailable **back off** to bigram if available, otherwise, unigram probability

1/26/2009

CSE842, Spring 2009, MSU

24

Backoff

Is the following right?

~~$$P_{\text{backoff}}(w_i | w_{i-2}w_{i-1}) = \begin{cases} P_{ML}(w_i | w_{i-2}w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ P_{ML}(w_i | w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) = 0 \\ P_{ML}(w_i), & \text{and } C(w_{i-1}w_i) > 0 \\ & \text{otherwise} \end{cases}$$~~

1/26/2009

CSE842, Spring 2009, MSU

25

Backoff

Any backoff language model must also be discounted

$$P_{\text{backoff}}(w_i | w_{i-2}w_{i-1}) = \begin{cases} \tilde{P}(w_i | w_{i-2}w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha_1 \tilde{P}(w_i | w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) = 0 \\ & \text{and } C(w_{i-1}w_i) > 0 \\ \alpha_2 \tilde{P}(w_i), & \text{otherwise} \end{cases}$$

1/26/2009

CSE842, Spring 2009, MSU

26

Katz Backoff Model (Bigram)

For a bigram model, define two sets

$A(w_{i-1}) = \{w: \text{Count}(w_{i-1}w, w) > 0\}$, $B(w_{i-1}) = \{w: \text{Count}(w_{i-1}w, w) = 0\}$

$$P_{\text{Katz}}(w_i | w_{i-1}) = \begin{cases} \frac{\text{Count}^*(w_{i-1}w_i)}{\text{Count}(w_{i-1})}, & \text{if } w_i \in A(w_{i-1}) \\ \alpha(w_{i-1}) \frac{P_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} P_{ML}(w)}, & \text{if } w_i \in B(w_{i-1}) \end{cases}$$

What is $\alpha(w_{i-1})$

1/26/2009

CSE842, Spring 2009, MSU

27

Katz Backoff Model (Bigram)

For a bigram model, define two sets

$A(w_{i-1}) = \{w: \text{Count}(w_{i-1}w, w) > 0\}$, $B(w_{i-1}) = \{w: \text{Count}(w_{i-1}w, w) = 0\}$

$$P_{\text{Katz}}(w_i | w_{i-1}) = \begin{cases} \frac{\text{Count}^*(w_{i-1}w_i)}{\text{Count}(w_{i-1})}, & \text{if } w_i \in A(w_{i-1}) \\ \alpha(w_{i-1}) \frac{P_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} P_{ML}(w)}, & \text{if } w_i \in B(w_{i-1}) \end{cases}$$

$$\alpha(w_{i-1}) = 1 - \sum_{w \in A(w_{i-1})} \frac{\text{Count}^*(w_{i-1}, w)}{\text{Count}(w_{i-1})}$$

1/26/2009

CSE842, Spring 2009, MSU

28

Linear Interpolation

$$P_{LI}(w_i | w_{i-2}, w_{i-1}) = \lambda_1 \times P_{ML}(w_i | w_{i-2}, w_{i-1}) \\ + \lambda_2 \times P_{ML}(w_i | w_{i-1}) \\ + \lambda_3 \times P_{ML}(w_i)$$

Where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and $\lambda_i \geq 0$ for all i .

1/26/2009

CSE842, Spring 2009, MSU

29

Linear Interpolation

Why does this estimation correctly define a distribution?

$$\sum_{w_i \in V} P_{LI}(w_i | w_{i-2}, w_{i-1}) \\ = \sum_{w_i \in V} [\lambda_1 \times P_{ML}(w_i | w_{i-2}, w_{i-1}) + \lambda_2 \times P_{ML}(w_i | w_{i-1}) + \lambda_3 \times P_{ML}(w_i)] \\ = \lambda_1 \sum_{w_i \in V} P_{ML}(w_i | w_{i-2}, w_{i-1}) + \lambda_2 \sum_{w_i \in V} P_{ML}(w_i | w_{i-1}) + \lambda_3 \sum_{w_i \in V} P_{ML}(w_i) \\ = \lambda_1 + \lambda_2 + \lambda_3 \\ = 1$$

1/26/2009

CSE842, Spring 2009, MSU

30

How to Set the Lambdas?

- Use a **held-out, or development**, corpus
- Choose lambdas which maximize the probability of some held-out data
 - i.e. fix the N gram probabilities, search for lambda values that when plugged into previous equation, give largest probability for held out set
 - use EM to do this search

1/26/2009

CSE842, Spring 2009, MSU

31

Part of Speech (POS) Tagging

- POS
- Tagsets
- Tagging algorithms
 - Rule based
 - HMM Viterbi Algorithm

1/26/2009

CSE842, Spring 2009, MSU

32

Part of Speech

- Eight basic POS:
 - Noun, verb, pronoun, preposition, adjective, conjunction, article, adverb
 - Based on morphological and syntactic function
- Called: parts of speech, lexical categories, word classes, morphological classes, lexical tags...
- Lots of debate within linguistics about the number, nature, and universality of these
 - We'll completely ignore this debate.

1/26/2009

CSE842, Spring 2009, MSU

33

Open and Closed Classes

- Closed class: a small fixed membership
 - Prepositions: of, in, by, ...
 - Auxiliaries: may, can, will had, been, ...
 - Pronouns: I, you, she, mine, his, them, ...
 - Usually **function words** (short common words which play a role in grammar)
- Open class: new ones can be created all the time
 - English has 4: Nouns, Verbs, Adjectives, Adverbs
 - Many languages have these 4, but not all!

1/26/2009

CSE842, Spring 2009, MSU

34

Open Class Words

- Nouns
 - Proper nouns (Boulder, Granby, Eli Manning)
 - English capitalizes these.
 - Common nouns (the rest).
 - Count nouns and mass nouns
 - Count: have plurals, get counted: goat/goats, one goat, two goats
 - Mass: don't get counted (snow, salt, communism) (*two snows)
- Adverbs: tend to modify things
 - **Unfortunately**, John walked home **extremely slowly** yesterday
 - Directional/locative adverbs (here, home, downhill)
 - Degree adverbs (extremely, very, somewhat)
 - Manner adverbs (slowly, slinkily, delicately)
- Verbs
 - In English, have morphological affixes (eat/eats/eaten)

1/26/2009

CSE842, Spring 2009, MSU

35

Closed Class Words

Examples:

- prepositions: *on, under, over, ...*
- particles: *up, down, on, off, ...*
- determiners: *a, an, the, ...*
- pronouns: *she, who, I, ..*
- conjunctions: *and, but, or, ...*
- auxiliary verbs: *can, may should, ...*
- numerals: *one, two, three, third, ...*

1/26/2009

CSE842, Spring 2009, MSU

36

Prepositions from CELEX

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

1/26/2009

CSE842, Spring 2009, MSU

37

English Particles

aboard	aside	besides	forward(s)	opposite	through
about	astray	between	home	out	throughout
above	away	beyond	in	outside	together
across	back	by	inside	over	under
ahead	before	close	instead	overhead	underneath
alongside	behind	down	near	past	up
apart	below	east, etc.	off	round	within
around	beneath	eastward(s),etc.	on	since	without

1/26/2009

CSE842, Spring 2009, MSU

38

Conjunctions

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in so far as	0
as	54,608	once	2,826	whereupon	85	inasmuch as	0
if	53,917	unless	2,205	seeing	63	insomuch as	0
when	37,975	why	1,333	directly	26	insomuch that	0
because	23,626	now	1,290	ere	12	like	0
so	12,933	neither	1,120	notwithstanding	3	neither nor	0
before	10,720	whenever	913	according as	0	now that	0
though	10,329	whereas	867	as if	0	only	0
than	9,511	except	864	as long as	0	provided that	0
while	8,144	till	686	as though	0	providing that	0
after	7,042	provided	594	both and	0	seeing as	0
whether	5,978	whilst	351	but that	0	seeing as how	0
for	5,935	suppose	281	but then	0	seeing that	0
although	5,424	cos	188	but then again	0	without	0
until	5,072	supposing	185	either or	0		

1/26/2009

CSE842, Spring 2009, MSU

39

Part of Speech Tagging

- What is POS tagging?
- Associate with each word a lexical tag
 - 45 classes from Penn Treebank
 - 87 classes from Brown Corpus
 - 146 classes from C7 tagset (CLAWS system)

1/26/2009

CSE842, Spring 2009, MSU

40

Why is POS Tagging Useful?

- First step of a vast number of practical tasks
- Speech synthesis
 - How to pronounce "lead"?
 - INsult inSULT
 - OBject obJECT
 - OVERflow overFLOW
 - DIScount disCOUNT
 - CONtent conTENT
- Parsing
 - Need to know if a word is an N or V before you can parse
- Information extraction
 - Finding names, relations, etc.

1/26/2009

CSE842, Spring 2009, MSU

41

Penn Treebank

- Large Corpora of 4.5 million words of American English
 - POS Tagged
 - Syntactic Bracketing
- : <http://www.cis.upenn.edu/~treebank>
- /user/cse842/Corpora/PennTreebank

1/26/2009

CSE842, Spring 2009, MSU

42

Penn Treebank

Description	Tagged for Part-of-Speech (Tokens)	Skeletal Parsing (Tokens)
Dept. of Energy abstracts	231,404	231,404
Dow Jones Newswire stories	3,065,776	1,061,166
Dept. of Agriculture bulletins	78,555	78,555
Library of America texts	105,652	105,652
MUC-3 messages	111,828	111,828
IBM Manual sentences	89,121	89,121
WBUR radio transcripts	11,589	11,589
ATIS sentences	19,832	19,832
Brown Corpus, retagged	1,172,041	1,172,041
Total:	4,885,798	2,881,188

1/26/2009

CSE842, Spring 2009, MSU

43

POS Tags from Penn Treebank

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	and, but, or	SYM	symbol	+, %, &
CD	cardinal number	one, two, three	TO	to	to
DT	determiner	a, the	UH	interjection	ah, oops
EX	existential 'there'	there	VB	verb, base form	eat
FW	foreign word	mea culpa	VBD	verb, past tense	ate
IN	preposition/sub-conj	of, in, by	VBG	verb, gerund	eating
JJ	adjective	yellow	VBN	verb, past participle	eaten
JJR	adj., comparative	bigger	VBP	verb, non-3sg pres	eat
JJS	adj., superlative	widest	VBZ	verb, 3sg pres	eats
LS	list item marker	1, 2, One	WDT	wh-determiner	which, that
MD	modal	can, should	WP	wh-pronoun	what, who
NN	noun, sing. or mass	llama	WPS	possessive wh-	whose
NNS	noun, plural	llamas	WRB	wh-adverb	how, where
NNP	proper noun, singular	IBM	\$	dollar sign	\$
NNPS	proper noun, plural	Carolinas	#	pound sign	#
PDT	predeterminer	all, both	"	left quote	' or "
POS	possessive ending	's	"	right quote	' or "
PRP	personal pronoun	I, you, he	(left parenthesis	[, (, {, <
PRPS	possessive pronoun	your, one's)	right parenthesis], }, >
RB	adverb	quickly, never	,	comma	,
RBR	adverb, comparative	faster	.	sentence-final punc	! ?
RBS	adverb, superlative	fastest	:	mid-sentence punc	:: ... --
/RP	particle	up, off			

1/26/2009

CSE842, Spring 2009, MSU

44

POS Tagging

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

We have:

- A set of tags (tagset)
- A dictionary with all possible tags for each word
- Input text

1/26/2009

CSE842, Spring 2009, MSU

45

How Hard is POS Tagging

- Most words are unambiguous
- Many of the most common English words are ambiguous

Unambiguous (1 tag)	35,340
Ambiguous (2-7 tags)	4,100
2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1 ("still")

1/26/2009

CSE842, Spring 2009, MSU

46

Tagging Method

- Rule-based tagging
- Statistical tagging
- Transformation-based tagging

1/26/2009

CSE842, Spring 2009, MSU

47

Rule-Based Tagging

- Start with a dictionary
- Assign all possible tags to words from the dictionary
- Write rules by hand to selectively remove tags
- Leaving the correct tag for each word.

1/26/2009

CSE842, Spring 2009, MSU

48

Start With a Dictionary

- she: PRP
 - promised: VBN,VBD
 - to TO
 - back: VB, JJ, RB, NN
 - the: DT
 - bill: NN, VB
- Etc... for the ~100,000 words of English with more than 1 tag

1/26/2009

CSE842, Spring 2009, MSU

49

Assign Every Possible Tag

NN
 RB
 VBN JJ VB
 PRP VBD TO VB DT NN
She promised to back the bill

1/26/2009

CSE842, Spring 2009, MSU

50

Write Rules to Eliminate Tags

Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"

VBN NN
 RB
 JJ VB
 PRP VBD TO VB DT NN
She promised to back the bill

1/26/2009

CSE842, Spring 2009, MSU

51

Rule-based Tagging: ENGTWOL

- A two stage architecture
 - Use dictionary (lexicon) to assign each word a list of potential POS
 - Use large lists of hand-written disambiguation rules to identify a single POS for each word.
- ENGTWOL tagger (Voutilainen,'95)
 - 56000 English word stems
 - Example: "Pavlov had shown that salivation" (p. 138)
- Advantage: high precision
- Disadvantage: needs a lot of rules

1/26/2009

CSE842, Spring 2009, MSU

52

Stage 1 of ENGTWOL Tagging

- First Stage: Run words through FST morphological analyzer to get all parts of speech.
- Example: *Pavlov had shown that salivation ...*

Pavlov	PAVLOV N NOM SG PROPER
had	HAVE V PAST VFIN SVO
shown	HAVE PCP2 SVO
that	SHOW PCP2 SVOO SVO SV
	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	N NOM SG

1/26/2009

CSE842, Spring 2009, MSU

53

Stage 2 of ENGTWOL Tagging

- Second Stage: Apply NEGATIVE constraints.
- Example: Adverbial "that" rule
 - Eliminates all readings of "that" except the one in
 - "It isn't *that* odd"

Given input: "that"

If

(+1 A/ADV/QUANT) ;if next word is adj/adv/quantifier

(+2 SENT-LIM) ;following which is E-O-S

(NOT -1 SVOC/A) ; and the previous word is not a

; verb like "consider" which

; allows adjective complements

; in "I consider that odd"

Then eliminate non-ADV tags

Else eliminate ADV

1/26/2009

CSE842, Spring 2009, MSU

54