CS-AD 220 – Spring 2016

Natural Language Processing

Session 12: 8-Mar-16

Prof. Nizar Habash

Some slides are adapted from Jurafsky and Martin's course slides on Speech and Language Processing

NYUAD CS-AD 220 – Spring 2016
Natural Language Processing

Assignment #2
Finite State Machines
Assigned Feb 18, 2016
Due Mar 10, 2016 (11:59pm)

I. Grading & Submission

This assignment is about the development of finite state machines using the OpenFST and Thrax toolkits. The assignment accounts for 15% of the full grade. It consists of three exercises. The first is a simple "machine translation" system for animal sounds to help with learning the tools. The second is about modeling how numbers are read in English and French. And the third is about Spanish verb conjugation. The answers should be placed in a zipped folder with separate sub-directories for each exercise.

The assignment is due on March 10 before midnight (11:59pm). For late submissions, 10% will be deducted from the homework grade for any portion of each late day. The student should upload the answers in a single zipped to NYU Classes (Assignment #2).

Assignment #2 posted on NYU Classes

Moving Legislative Day Class

- Spring Break is March 18 – 25, 2016
- Sat March 26, 2016 is a Legislative *Thursday*
- Move to

Sat April 2, 2016 at 10am

Same Classroom C2-E049

Final Exam!

Monday May 16th

1pm-4pm

CR-002

Interview a Professor

Mock Lectures on

- Thursday March 10 @ 11am in ERB 045
- Sunday March 13 @ 11am in ERB 045

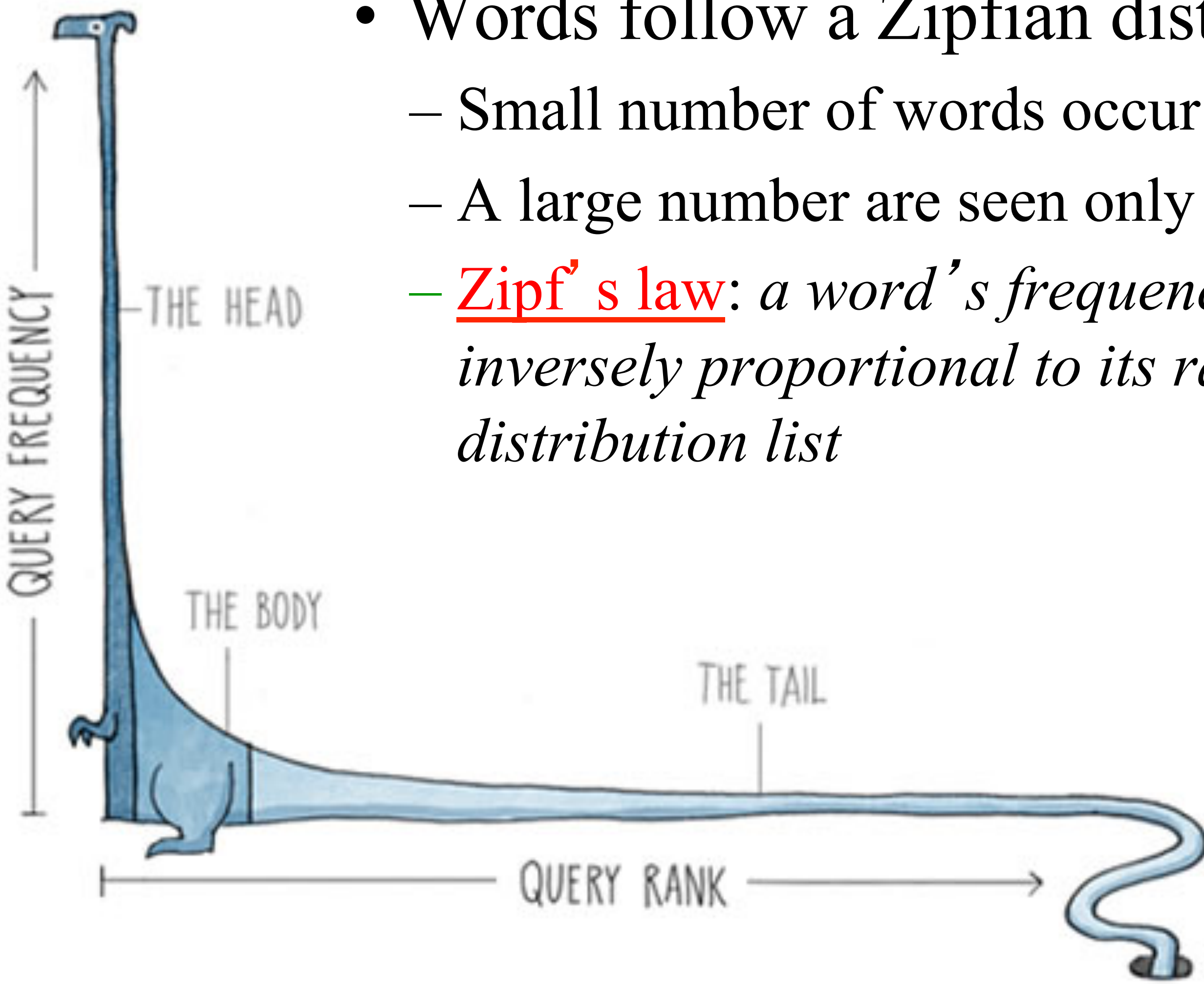
Extra credit for doing this (1%)

Must provide a short review (250 words) by Monday March 14.

Midterm in one week!

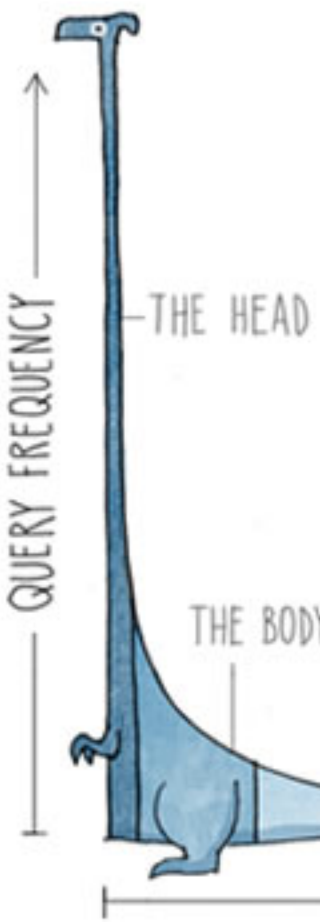
- Tuesday March 15
  - 5 questions; 75 minutes.

Smoothing



- Words follow a Zipfian distribution
  - Small number of words occur very frequently
  - A large number are seen only once
  - Zipf's law: *a word's frequency is approximately inversely proportional to its rank in the word distribution list*

Smoothing



- Words follow a Zipfian distribution
  - Small number of words occur very frequently
  - A large number are seen only once
  - Zipf's law: *a word's frequency is approximately inversely proportional to its rank in the word distribution list*
- Problem with unseen n-grams → zero probabilities
- So….how do we estimate the likelihood of unseen n-grams?

Laplace (Add-1) Smoothing

- For unigrams:
  - Add 1 to every word (type) count to get an adjusted count c^*
  - Normalize by N (#tokens) + V (#types)
  - Original unigram probability

$$P(w_i) = \frac{c_i}{N}$$

  - New unigram probability

$$P_{LP}(w_i) = \frac{c_i + 1}{N + V}$$

- ***So, we lower some (larger) observed counts in order to include unobserved vocabulary***
- For bigrams:
  - Original $$P(w_n|w_{n-1}) = \frac{c(w_n|w_{n-1})}{c(w_{n-1})}$$
  - New $$P(w_n|w_{n-1}) = \frac{c(w_n|w_{n-1})+1}{c(w_{n-1})+V}$$
  - ***But this change counts drastically***:
    - ***Too much weight*** given to unseen ngrams
    - In practice, unsmoothed bigrams often work better!
    - Can we smooth more usefully?

Good-Turing Discounting

- Re-estimate amount of probability mass for zero (or low count) ngrams by looking at ngrams with higher counts
  - Estimate

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

- N_c = the count of things we've seen c times – Frequency of frequency c

Sam I am I am Sam I do not eat green eggs

I 3	*do 1*	*green 1*
sam 2	*not 1*	*eggs 1*
am 2	*eat 1*	

N_1 = 5

N_2 = 2

N_3 = 1

Good-Turing smoothing intuition

- You are fishing (a scenario from Josh Goodman), and caught:
  - 10 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel = 18 fish
- How likely is it that next species is trout?
  - 1/18
- How likely is it that next species is new (i.e. catfish or bass)
  - Let's use our estimate of things-we-saw-once to estimate the new things.
  - 3/18 (because N_1=3)
- Assuming so, how likely is it that next species is trout?
  - Must be less than 1/18
  - How to estimate?

Good Turing calculations

$$P^*_{GT}(\text{things with zero frequency}) = \frac{N_1}{N}$$

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

- Unseen (bass or catfish)
  - c = 0:
  - MLE p = 0/18 = 0
  - P^*_{GT} (unseen) = N_1/N = 3/18
  - Two unseen fish => each get 3/(18*2)

- Seen once (trout)
  - c = 1
  - MLE p = 1/18
  - C*(trout) = (1+1) * N_2/N_1
    = 2 * 1/3
    = 2/3
  - P^*_{GT}(trout) = 2/3 / 18
    = 1/27

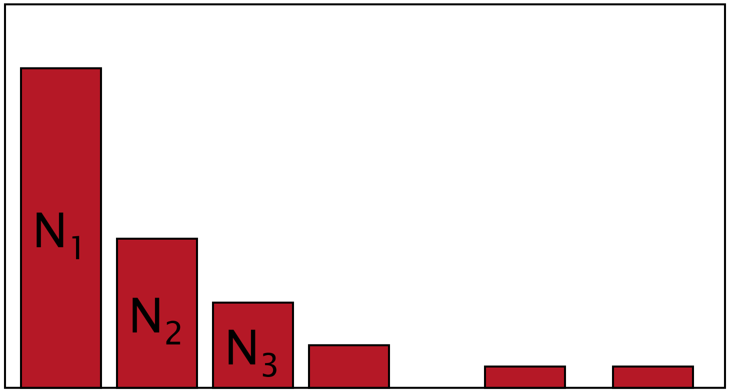
10 carp, 3 perch, 2 whitefish,
1 trout, 1 salmon, 1 eel = 18 fish

N_0=?? N_1=3 N_2=1 N_3=1 N_{10}=1

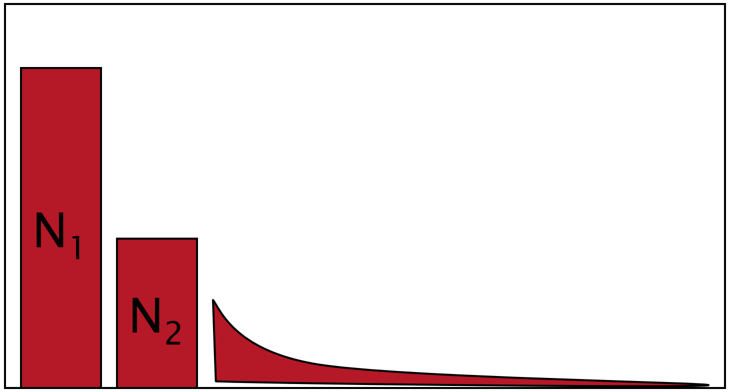
Good-Turing complications

(slide from Dan Klein)

- Problem: what about “the”? (say c=4417)
  - For small k, $N_k > N_{k+1}$
  - For large k, too jumpy, zeros wreck estimates



  - Simple Good-Turing [Gale and Sampson]: replace empirical N_k with a best-fit power law once counts get unreliable



Resulting Good-Turing numbers

- Numbers from Church and Gale (1991)
- 22 million words of AP Newswire

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

Count c	Good Turing c*
0	.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

Resulting Good-Turing numbers

- Numbers from Church and Gale (1991)
- 22 million words of AP Newswire

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

- It sure looks like c* = (c - .75)

Count c	Good Turing c*
0	.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

Backoff and Interpolation

- Sometimes it helps to use **less** context
  - Condition on less context for contexts you haven't learned much about
- **Backoff:**
  - use trigram if you have good evidence,
  - otherwise bigram, otherwise unigram

$$P_{katz}(w_n \mid w_{n-N+1}^{n-1}) = \begin{cases} P^*(w_n \mid w_{n-N+1}^{n-1}) & \text{if } C(w_{n-N+1}^{n}) > 1 \\ \alpha(w_{n-N+1}^{n-1}) P_{katz}(w_n \mid w_{n-N+2}^{n-1}) & \text{otherwise} \end{cases}$$

  - *P^* is a discounted probability estimate to reserve mass for unseen events and α's are back-off weights.*

Backoff and Interpolation

- Sometimes it helps to use **less** context
  - Condition on less context for contexts you haven't learned much about
- **Interpolation:**
  - mix unigram, bigram, trigram

$$\begin{aligned}\hat{P}(w_n|w_{n-1}w_{n-2}) &= \lambda_1 P(w_n|w_{n-1}w_{n-2}) \\ &+\lambda_2 P(w_n|w_{n-1}) \\ &+\lambda_3 P(w_n)\end{aligned}$$

$$\sum_i \lambda_i = 1$$

- Interpolation works better

Smoothing Summed Up

- Add-one smoothing (easy, but inaccurate)
  - Add 1 to every word count (Note: this is type)
  - Increment normalization factor by Vocabulary size: N (tokens) + *V (types)*
- Good-Turing
  - Re-estimate amount of probability mass for zero (or low count) ngrams by looking at ngrams with higher counts
- Backoff models
  - When a count for an n-gram is 0, back off to the count for the (n-1)-gram
  - These can be weighted – trigrams count more
- Many other advanced methods
  - Kneser-Ney Smoothing
  - …

Part-of-Speech Tagging

- *What's the plural of "Part-of-Speech"?*

  → *Parts-of-Speech*

  not Part-of-Speeches ☺

- *Abbreviation: POS*

Parts of Speech

- 8 (ish) traditional parts of speech
  - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc
  - 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.

POS examples

N	noun
V	verb
ADJ	adjective
ADV	adverb
P	preposition
PRO	pronoun
DET	determiner

POS examples

- N noun *chair, bandwidth, pacing*
- V verb *study, debate, munch*
- ADJ adjective *purple, tall, ridiculous*
- ADV adverb *unfortunately, slowly*
- P preposition *of, by, to*
- PRO pronoun *I, me, mine*
- DET determiner *the, a, that, those*

POS Tagging

- The process of assigning a part-of-speech or lexical class marker to each word in a collection.

WORD	tag
the	**DET**
koala	**N**
put	**V**
the	**DET**
keys	**N**
on	**P**
the	**DET**
table	**N**

Why is POS Tagging Useful?

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

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 four: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have these four too

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
  - 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)

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, …*

Prepositions from CELEX
frequencies from COBUILD 16M word corpus

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

English Particles
(Quirk et al., 1985)

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

POS Tagging
Choosing a Tagset

- There are so many parts of speech, potential distinctions we can draw
- To do POS tagging, we need to choose a standard set of tags to work with
- Could pick very coarse tagsets
  - N, V, Adj, Adv.
- More commonly used set is finer grained, the "Penn TreeBank tagset", 45 tags
  - PRP$, WRB, WP$, VBG
- Even more fine-grained tagsets exist

Penn TreeBank POS Tagset

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	*wildest*	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*	WP$	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	[, (, {, <
PRP$	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*			

Using the Penn Tagset

- The**/DT** grand**/JJ** jury**/NN** commented**/VBD** on**/IN** a**/DT** number**/NN** of**/IN** other**/JJ** topics**/NNS** ./**.**
- Prepositions and subordinating conjunctions marked IN ("although/IN I/PRP..")
- Except the preposition/complementizer "to" is just marked "TO".

POS Tagging

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

These examples from Dekang Lin

How Hard is POS Tagging? Measuring Ambiguity

		45-tag Treebank Brown	
Unambiguous (1 tag)		**38,857**	
Ambiguous (2–7 tags)		**8844**	
Details:	2 tags	6,731	
	3 tags	1621	
	4 tags	357	
	5 tags	90	
	6 tags	32	
	7 tags	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags	4	(*'s*, *half*, *back*, *a*)
	9 tags	3	(*that*, *more*, *in*)

1 → 81%
2 → 14%
3 → 3%
4+ → ~1%

Baseline = ?

Assuming equal probabilities for ambiguous POS tags

1 → 81%
2 → +7%
3 → +1%
4+ → ~+0.2%

Baseline = ~89.2%

Two Methods for POS Tagging

1. Rule-based tagging
   - Large databases of hand-written rules
     - EngCG / ENGTWOL
2. Stochastic
   - Probabilistic sequence models
     - HMM (Hidden Markov Model) tagging
     - …

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.

Start With a Dictionary

- she: PRP
- promised: VBN,VBD
- to TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB

- Etc… for the words of English with more than 1 tag

Assign Every Possible Tag

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

Write Rules to Eliminate Tags

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

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

EngCG/ENGTWOL Tagging Stage 1

- 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**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**

EngCG/ENGTWOL Tagging Stage 2

- 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

- EngCG has around 3,700 such constraints!

Next Time

- Read J+M Chap 5 (5.8 to end);
  handout (Pasha et al., 2014)
- Assignment #2 due March 10 midnight
- Midterm in one week!