6.863J Natural Language Processing
Lecture 5: What's my line?

Instructor: Robert C. Berwick
berwick@ai.mit.edu

The Menu Bar

- Administrivia
  Lecture 3 posted; Lab 1a (aka "component II") due yesterday; Lab 1b, due next Monday
- Postmortem: Complexity of Kimmo/fst's – too weak? Too strong? What makes a good computational linguistics representation? A good linguistic representation? A good algorithm?
- Alternatives: morphology w/o a dictionary
- What's my line: take a chance

Is Kimmo <u>necessary</u>?

- Does it explain why many non-human systems never occur (ruling them out)
- Or does it overshoot?

- Ans: it seems to overshoot, in at least 2 ways
- Overshoots detected by <u>computational</u> analysis

Overshoot #1: too powerful with dependencies

- More powerful than well-known grammars in linguistics (and computational linguistics)
- We can use kimmo to 'count' – but natural languages don't (or cannot) do this…
- (Recall: we can use Kimmo to output a language with <u>one</u> counting relation: $a^n b^n$ – not a finite-state language)
- But we can do more… nothing stops us from producing a language with *m* counting relations, e.g, for any n, $\{(x, (cx)^n) \mid x \in \{a^* b^*\}\}$, e.g., for n=3, cababcababcabab, cbbbcbbbcbbb…

Not captured by context-free language

- (Familiar): $a^n b^n c^n$
- Intuition: use of pushdown stack – can catch <u>one</u> such pairing, but not more

So: Kimmo admits more than context-free languages!

- So Kimmo is more powerful than this!

  But how powerful is it? We can still parse context-free languages in cubic time (in the length of sentences)
- We shall see that Kimmo is <u>more complex</u> than this!
- Conjecture: all the context-sensitive languages

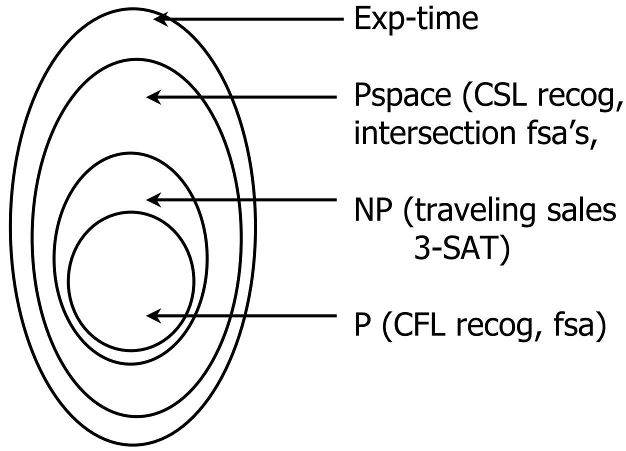
Complexity of Kimmo word recognition

- All these finite-state devices, working in parallel
- There is backup
- Is it intrinsic to the system? Or eradicable? Or, doesn't matter in practice?

Litmus test #2 – computational complexity of Kimmo – word parsing is intractable!

- Kimmo Recognition Problem (KRP):
  Given a language defined by an arbitrary (finite) Kimmo dictionary (lexical automata) and a finite set of Kimmo rules, how long in the worst case will it take to recognize whether a form is or is not in the language?
- Kimmo recognition problem is NP-hard
- As hard as any other problem solvable by a nondeterminstic Turing machine in polynomial time
- No known det polytime (eg, cubic) algorithm for NP-hard problems…

Complexity hierarchy



Parsing words with Kimmo is computationally intractable

- Intuition: what if the characters on the surface don't give any clues as to what 'features' they ought to have underlyingly? (e.g., whether a Noun or a Verb, as in *police police police*)
- This seems awfully close to the famous 3-SAT problem: is there an assignment of T(rue), F(alse) to the literals of an arbitrary Boolean formula in 3-conjunctive normal form s.t. the formula evaluates to *true*?
- In fact, we can simulate this problem using Kimmo

3-Sat (3-satisfiability) is NP-complete

- *Given* (arb) 3-Sat formula, e.g.,
- There is no known deterministic Turing machine that can figure out quickly (in polynomial time) whether there is an assignment of *true* or *false* to literals *x,y, z* in order to make the formula evaluates to true <u>just</u> by inspecting the local surface string
- We could <u>guess</u> this in polynomial time – i.e., Nondeterministic Polynomial, or NP time (time measured in length of the formula)

Reduction of 3-Sat to Kimmo recognition problem

- For every 3-Sat problem, we can find, in polynomial time, a corresponding Kimmo word recognition problem where there's a valid word if the 3-Sat problem was satisfiable
- If Kimmo recognition could be done in deterministic polynomial time (P) then so could 3-SAT

Reduction

Any 3-Sat problem

↓ ← Efficient (polynomial time) transformation

Equivalent Kimmo recognition problem

↓

Answer to original SAT problem

The reduction:

Given: arbitrary 3-SAT problem instance, e.g.,

(x v ¬y v z) (¬x v ¬z) (x v y)

Fast (polytime) transformation

(fixed) Lexicon, *L*

Fst's, 1 per variable

word∈L if Sat instance satisfiable

If we could solve Kimmo recognition easily,

Then we could solve 3-Sat easily

Two components to 3-Sat

- The fact that an *x* that has a truth assignment in one place, must have the same truth assignment everywhere - what morphological process is that like?

- The fact that every triple must have at least 1 'T' underlyingly (so that the triple is true) - what morphological process is that like?

How the reduction works

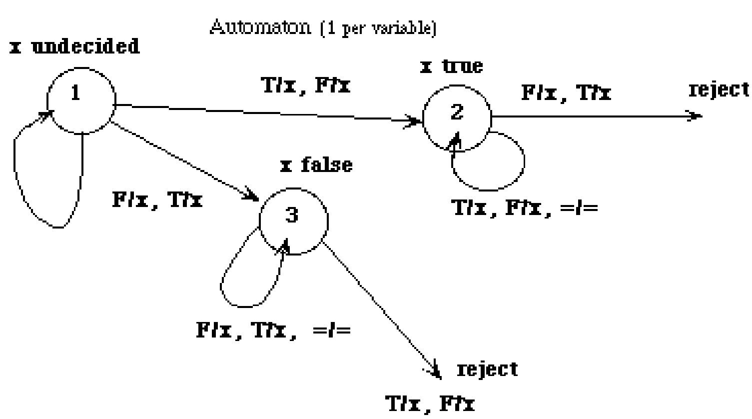
- Given arbitrary 3-sat formula ϕ, e.g.,
  (x v ¬y v z) (¬x v ¬z) (x v y)
- Represent in the form, a 'word':
  x-yz,-xz,xy

- For each variable x, we have an 'assignment machine' that ensures that x is mapped to T or F throughout the whole formula
- We have one machine (and a fixed dictionary) to checks each disjunction to make sure that at least one disjunct is true in every conjunct

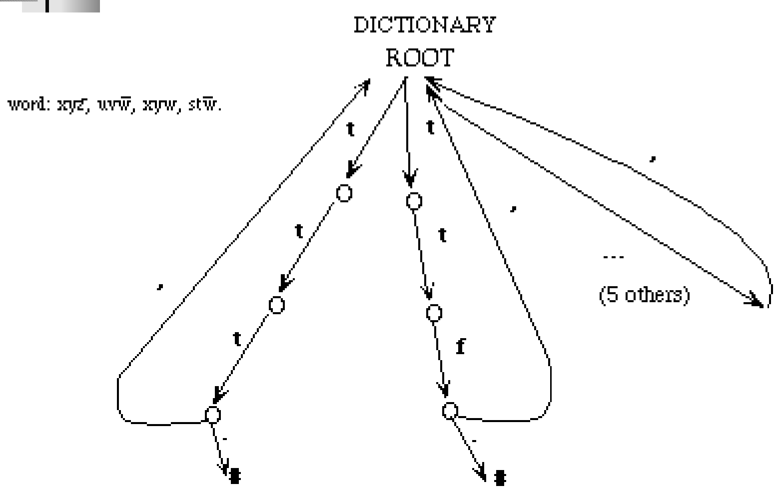
Two components

- Agreement: vowel harmony (if round at some point, round everywhere)
- Ambiguity: we can't tell what the underlying value of *x* is from the surface, but if there's at least one "t" per 'part of word', then we can spell out this constraint in dictionary
- Note that words (like Sat formulas) must be arbitrarily long… (pas de probleme)
- Dictionary is fixed…
- # of Vowel harmony processes corresponds to # of distinct literals

Reduce until done: assignment consistency



Reduce until done – formula must eval to *true*



What are the implications?

- FTNs inherently require backup if simulated (in the worst case) – Kimmo at least NP-hard (proof later on)
- Empty elements cause computational complexity (unless restricted – equal length condition) – true in <u>all</u> areas of linguistics
- Composition can save us, but then rule ordering must be watched carefully

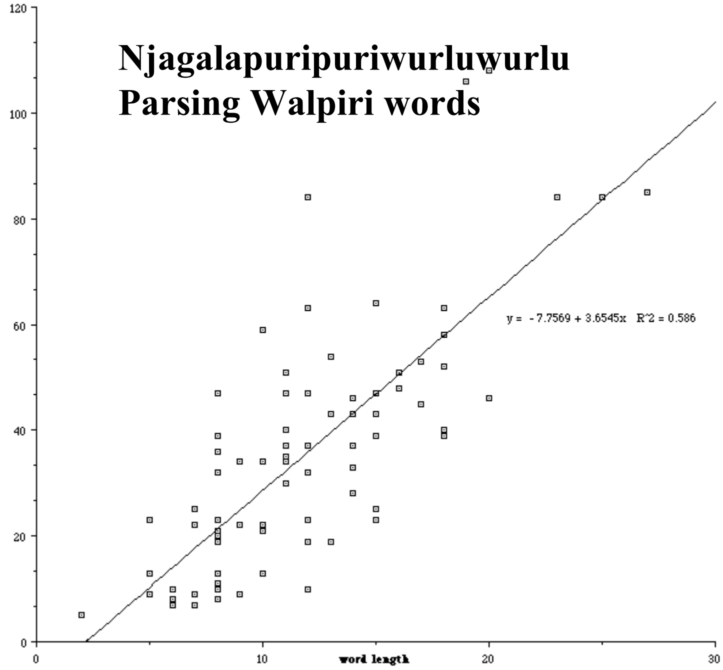
Implications

- Do we need a machine powerful enough to represent intractable problems?
- No evidence for unbounded # of counting dependencies or harmony processes…
- Performance? Or do we need something this powerful??

Why should we care?

- This is *typical* of a combination of ‘agreement and ambiguity’ that trickles through all of natural language
- The agreement part – like Turkish vowel harmony
- The ambiguity part – like the *police police police* example
- Suggests that speed won’t come from the formalism *all by itself*

Njagalapuripuriwurluwurlu
Parsing Walpiri words



What if we don't have a dictionary?

- Don't use one
- Learn one from data

Method 1: don't use a dictionary

- Best known method – Porter stemming (Porter, 1980)
- http://www.tartarus.org/~martin/PorterStemmer/
  http://snowball.tartarus.org/
  - For English
  - Most widely used system
  - Manually written rules
  - 5 stage approach to extracting roots
  - Considers suffixes only
  - May produce non-word roots

Porter output

Sample Output (English):

consigned	consign	knack	knack
consignment	consign	knackeries	knackeri
consolation	consol	knaves	knavish
consolatory	consolatori	knavish	knavish
consolidate	consolid	knif	knif
consolidating	consolid	knife	knife
consoling	consol	knew	knew

Why?

Algorithmic stemmers can be fast (and lean):

E.g.: 1 Million words in 6 seconds on 500 MHz PC

- It is more efficient not to use a dictionary (don’t have to maintain it if things change).
- It is better to ignore irregular forms (exceptions) than to complicate the algorithm (not much lost in practice).

Output - German

aufeinander	aufeinand	kategorie	kategori
auferlegen	auferleg	kategorien	kategori
auferlegt	auferlegt	kater	kat
auferlegten	auferlegt	katers	kat
auferstanden	auferstand	katze	katz
auferstehen	auferstand	katzen	katz
aufersteht	aufersteht	kätzchen	katzch

Method

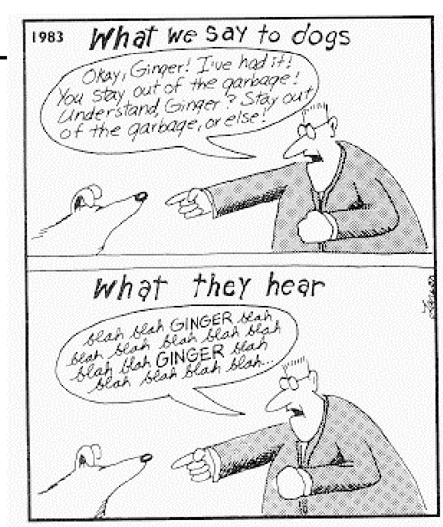
Porter Stemmers use simple algorithms to determine which affixes to strip in which order and when to apply repair strategies.

Input	Strip -ed Affix	Repair
hoped	hop	hope (add -e if word is short)
hopped	hopp	hop (delete one if doubled)

Samples of the algorithms are accessible via the Web and can be programmed in any language.

Advantage: easy to see understand, easy to implement.

Words are fine – but we need more



Paradigmatic example for NLP

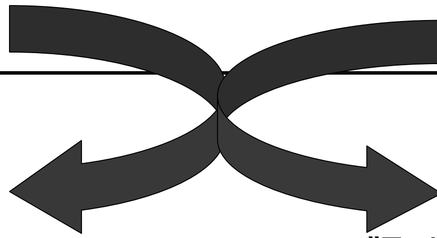
- Morphophonemic parsing
- Given surface form, recover underlying form:

Two ways

- Generative model – concatenate then fix up joints
  - stop + -ing = stopping,   fly + s = flies
  - Use a cascade of transducers to handle all the fixups
- Probabilistic model - some constraints on morpheme sequences using prob of one character appearing before/after another

  prob(ing | stop) vs. prob(ly| stop)

The Great Divide in NLP:



"Knowledge Engineering" approach
Rules built by hand w/ K of Language
"Text understanding"

"Trainable Statistical" Approach
Rules inferred from lots of data ("corpora")
"Information retrieval"

the red pill or the blue pill?

Another example: generating language

- Variations in style, syntactic form…
- Where do these come from?

- Jane Austin writes differently from Charlotte Brontë

The Red Pill

- How to fold statistical information in to the symbolic models?
- Let’s try a classic and simple way to define a statistical model for language…

Language ID

- “Rabbit and Lukasiewicz are on the menu”

- Is this English or Polish or what?
- Is it “good” (= likely) English?
- Is it “good” (= likely) Polish?

Text Categorization

- Automatic Yahoo classification, etc.
- Similar to language ID …
  - Topic 1 sample: In the beginning God created …
  - Topic 2 sample: The history of all hitherto existing society is the history of class struggles. …
- Input text: Matt's Communist Homepage. Capitalism is unfair and has been ruining the lives of millions of people around the world. The profits from the workers' labor …
- Input text: And they have beat their swords to ploughshares, And their spears to pruning-hooks. Nation doth not lift up sword unto nation, neither do they learn war any more. …

Contextual Spelling Correction

- Which is most probable?
  - … I think they're okay …
  - … I think there okay …
  - … I think their okay …
- Which is most probable?
  - … by the way, are they're likely to …
  - … by the way, are there likely to …
  - … by the way, are their likely to …

Topic Segmentation

- Break big document or media stream into indexable chunks
- From NPR's *All Things Considered*:

  The U. N. says its observers will stay in Liberia only as long as West African peacekeepers do, but West African states are threatening to pull out of the force unless Liberia's militia leaders stop violating last year's peace accord after 7 weeks of chaos in the capital, Monrovia … Human rights groups cite peace troops as among those smuggling the arms. I'm Jennifer Ludden, reporting. Whitewater prosecution witness David Hale began serving a 28-month prison sentence today. The Arkansas judge and banker pleaded guilty two years ago to defrauding the Small Business Administration. Hale was the main witness in the Whitewater-related trial that led to the convictions …

Contextual Spelling Correction

- Which is most probable?
  - … I think they're okay …
  - … I think there okay …
  - … I think their okay …
- Which is most probable?
  - … by the way, are they're likely to …
  - … by the way, are there likely to …
  - … by the way, are their likely to …

Speech Recognition

- How do you wreck a nice beach?
- How do you recognize speech?

- Put the file in the folder
- Put the file and the folder

Language generation

- Choose randomly among outputs:
  - Visitant which came into the place where it will be Japanese has admired that there was Mount Fuji.
- Top 10 outputs according to bigram probabilities:
  - Visitors who came in Japan admire Mount Fuji.
  - Visitors who came in Japan admires Mount Fuji.
  - Visitors who arrived in Japan admire Mount Fuji.
  - Visitors who arrived in Japan admires Mount Fuji.
  - Visitors who came to Japan admire Mount Fuji.
  - A visitor who came in Japan admire Mount Fuji.
  - The visitor who came in Japan admire Mount Fuji.
  - Visitors who came in Japan admire Mount Fuji.
  - The visitor who came in Japan admires Mount Fuji.
  - Mount Fuji is admired by a visitor who came in Japan.

Basic Morphology

Basic Affix Typology (don’t seem to need more):

- i-suffix: inflectional suffix

  English: *cheer+ed = cheered, fit+ed = fitted, love+ed = loved*

- d-suffix: derivational suffix, changes word type

  English: *walk(V)+er = walker(N), happy(A)+ness=happiness(N)*

- a-suffix: attached suffix (enclitics).

  Italian *mandargli= mandare+gli* = to send + to him

Algorithmic Method

General Strategy:

- Normal order of suffixes seems to be *d, i, a.*
- Remove from right in order *a, i, d.*
- Generally remove all the *a* and *i* suffixes, sometimes leave the *d* one.

Types of Errors

• Conflation:	reply, rep.	rep
• Overstemming:	wander	wand
	news	new
• Misstemming:	relativity	relative
• Understemming:	knavish	knavish

Algorithmic Method

Strategy for German:

- Leave prefixes alone because they can change meaning.
- Put everything in small caps.
- Get rid of *ge-.*
- Get rid of *i* type: *e, em, en, ern, er, es, s, est,* (e.g, *armes > arm*)
- Get rid of *d* type*: end, ung, ig, ik, isch, lich, heit, keit*

Information Retrieval

Does stemming indeed improve IR?

- No: Harman (1991), Krovetz (1993)

- Possibly: Krovetz (1995)
  Depends on type of text, and the assumption is that once one moves beyond English, the difference will prove significant.

Crosslinguistic Applicability

- Can this type of stemming be applied to all languages?
  - Not to Chinese, for example (doesn't need it).

- Do all languages have the same kind of morphology?
  - No. Stemming assumes basically agglutinative morphology. This is not true crosslinguistically (but the algorithms seem to work pretty well within Indo-

Stemming: Methods

- Dictionary approach not enough
  - Example: (Porter, 1991)
    - routed -> route/rout
      - At Waterloo, Napoleon's forces were routed
      - The cars were routed off the highway
    - Here, the (inflected) verb form is polysemous

Stemming: Errors

- Understemming: failure to merge
  - Adhere/adhesion
- Overstemming: incorrect merge
  - Probe/probable
    - Claim: -*able* irregular suffix, root: *probare* (Lat.)

- Mis-stemming: removing a non-suffix (Porter, 1991)
  - reply -> rep

Stemming: Interaction

- Interacts with noun compounding:
  - Example:
    - operating systems
    - negative polarity items
  - For IR, compounds need to be identified first…

Stemming: Porter Algorithm

- Rule format:
  - (condition on stem) $suffix_1$ -> $suffix_2$
  - In case of conflict, prefer longest suffix match
- "Measure" of a word is *m* in:
  - $(C)\ (VC)^m\ (V)$
  - C = sequence of one or more consonants
  - V = sequence of one or more vowels
  - Examples:
    - *tree* $C(VC)^0V$
    - *troubles* $C(VC)^2$

Stemming: Porter Algorithm

- Step 1a: remove plural suffixation
  - SSES -> SS (caresses)
  - IES -> I (ponies)
  - SS -> SS (caress)
  - S -> (cats)
- Step 1b: remove verbal inflection
  - (m>0) EED -> EE (agreed, feed)
  - (*v*) ED -> (plastered, bled)
  - (*v*) ING -> (motoring, sing)

Stemming: Porter Algorithm

- Step 1b: (contd. for -*ed* and -*ing* rules)
  - AT -> ATE (conflat*ed*)
  - BL -> BLE (troubl*ed*)
  - IZ -> IZE (siz*ed*)
  - (*doubled c & ¬(*L v *S v *Z)) -> single c (hopp*ing*, hiss*ing*, fall*ing*, fizz*ing*)
  - (m=1 & *cvc) -> E (fil*ing*, fail*ing*, slow*ing*)
- Step 1c: Y and I
  - (*v*) Y -> I (happy, sky)

Stemming: Porter Algorithm

- Step 2: Peel one suffix off for multiple suffixes
  - (m>0) ATIONAL -> ATE (relational)
  - (m>0) TIONAL -> TION (conditional, rational)
  - (m>0) ENCI -> ENCE (valenci)
  - (m>0) ANCI -> ANCE (hesitanci)
  - (m>0) IZER -> IZE (digitizer)
  - (m>0) ABLI -> ABLE (conformabli) - *able* (step 4)
  - …
  - (m>0) IZATION -> IZE (vietnamization)
  - (m>0) ATION -> ATE (predication)
  - …
  - (m>0) IVITI -> IVE (sensitiviti)

Stemming: Porter Algorithm

- Step 3
  - (m>0) ICATE -> IC (triplicate)
  - (m>0) ATIVE -> (formative)
  - (m>0) ALIZE -> AL (formalize)
  - (m>0) ICITI -> IC (electriciti)
  - (m>0) ICAL -> IC (electrical, chemical)
  - (m>0) FUL -> (hopeful)
  - (m>0) NESS -> (goodness)

Stemming: Porter Algorithm

- Step 4: Delete last suffix
  - (m>1) AL -> (revival) - *revive*, see step 5
  - (m>1) ANCE -> (allowance, dance)
  - (m>1) ENCE -> (inference, fence)
  - (m>1) ER -> (airliner, employer)
  - (m>1) IC -> (gyroscopic, electric)
  - (m>1) ABLE -> (adjustable, mov(e)able)
  - (m>1) IBLE -> (defensible,bible)
  - (m>1) ANT -> (irritant,ant)
  - (m>1) EMENT -> (replacement)
  - (m>1) MENT -> (adjustment)
  - …

Stemming: Porter Algorithm

- Step 5a: remove *e*
  - (m>1) E -> (probate, rate)
  - (m>1 & ¬*cvc) E -> (cease)

- Step 5b: *ll* reduction
  - (m>1 & *LL) -> L (controll*er*, roll)

Stemming: Porter Algorithm

- Misses (understemming)
  - Unaffected:
    - *agreement* $(VC)^1VCC$ - step 4 (m>1)
    - *adhesion*
  - Irregular morphology:
    - drove, geese
- Overstemming
    - *relativity* - step 2
- Mis-stemming
    - *wander* $C(VC)^1VC$

The Great Divide in NLP: the red pill or the blue pill?

"Knowledge Engineering" approach
Rules built by hand w/ K of Language
"Text understanding"

"Trainable Statistical" Approach
Rules inferred from lots of data ("corpora")
"Information retrieval"

A simple example

- We consider a sentence as a sequence of *n* words; we want to find P($w_1, \ldots, w_n$)
- We can use this to model all the 'noise' that gets into language (the leaks)
- So one idea is to combine the symbolic models (like kimmo) with the 'noise' components, to do better (eg, like econometrics…)

Language models, probability & info

- Given a string *w,* a language model gives us the probability of the string P(*w*), e.g.,
  - *P(the big dog) > (dog big the) > (dgo gib eth)*
  - Easy for humans; difficult for machines
  - Let P(*w*) be called a <u>language model</u> L(M)
- "I have a <u>gub"</u> (Woody Allen)

A simple example – which is 'right?'

physical Brainpower not plant is chief , now a 's asset , . firm ,

a Brainpower not now chief asset firm 's is . plant physical ,

chief a physical , . firm not , Brainpower plant is asset 's now

now not plant Brainpower now physical 's . a chief , asset firm , is

Brainpower , not physical plant , is now a firm 's chief asset .

Each sentence is a sequence $w_1, \ldots, w_n$.
Task is to find P($w_1, \ldots, w_n$).

N-grams

A simple model of language

- Computes a probability for observed input
- Probability is likelihood of observation

being generated by the same source as training data

- Such a model is often called a *language model* (LM)

How can we compute this?

- Each of this pr's can be estimated (using frequency counts) from training data
- Can this work directly?
- No – not in practice…why?

What's this good for?

- What's my line?

Language ID

- "Rabbit and Lukasiewicz are on the menu"

- Is this English or Polish or what?
- We had some notion of using n-gram models …

- Is it "good" (= likely) English?
- Is it "good" (= likely) Polish?

- Space of events will be not races but character sequences (x_1, x_2, x_3, …) where x_n = EOS (nb, "BOS")

Language ID?

- Let p(X) = probability of text X in English
- Let q(X) = probability of text X in Polish
- Which probability is higher?

"Rabbit and Lukasiewicz are on the menu"

$p(x_1=\texttt{r}, x_2=\texttt{a}, x_3=\texttt{b}, x_4=\texttt{b}, x_5=\texttt{i}, x_6=\texttt{t}, \ldots)$

Data needs

- How many possible distinct probabilities will be needed?, i.e. parameter values
- Total number of word tokens in our training data
- Total number of unique words: word types is our vocabulary size

How can we compute this?

- Each of this pr's can be estimated (using frequency counts) from training data
- Can this work?
- No – not in practice…why?

What are the tools we need?

- Crash….

Probability you should know..

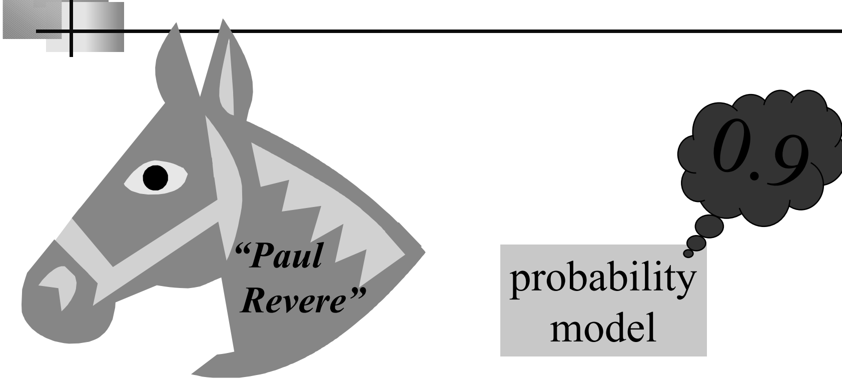
- Probability notation like p(X | Y):
  - What does this expression mean?
  - How can I manipulate it?
  - How can I estimate its value in practice?
- Probability models:
  - What is one?
  - Can we build one for language?
  - How do I know if my model is any good?

3 Kinds of Statistics

- descriptive: mean MIT SAT (or median)
- confirmatory: statistically significant?
- predictive: wanna bet?

this course – why?

Notation for Greenhorns



p(Paul Revere wins | weather's clear) = 0.9

What does that *really* mean?

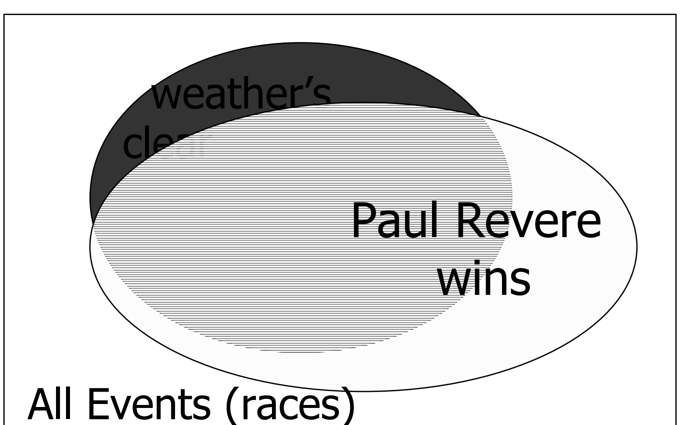
p(Paul Revere wins | weather's clear) = 0.9

- Past performance?
  - Revere's won 90% of races with clear weather
- Hypothetical performance?
  - If he ran the race in many parallel universes …
- Subjective strength of belief?
  - Would pay up to 90 cents for chance to win $1
- Output of some computable formula?
  - Ok, but then which formulas should we trust?

    p(X | Y) versus q(X | Y)

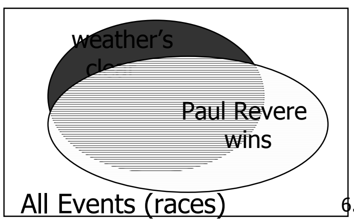
p is a function on event sets

p(win | clear) ≡ p(win, clear) / p(clear)



p is a function on event sets

$$\underbrace{p(\text{win} \mid \text{clear})}_{\text{syntactic sugar}} \equiv \underbrace{p(\text{win, clear})}_{\text{logical conjunction of predicates}} / \underbrace{p(\text{clear})}_{\text{predicate selecting races where weather's clear}}$$



p measures total probability of a set of events.

The Chain rule – factoring joint events

- P(GC in Hawaii,GC alone,GC low in polls|GC drives drunk)=

  P(GC in Hawaii|GC alone, GC low in polls,GC drives drunk)×

  P(GC alone|GC low in polls,GC drives drunk)×

  P(GC low in polls|GC drives drunk)

  Why does this work?

Chain rule – why does it work?

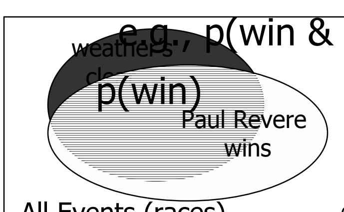
- Remember: $P(X|Y)=P(X,Y)/P(Y)$
- $\frac{HALD}{D} = \frac{HALD}{ALD} \times \frac{ALD}{LD} \times \frac{LD}{D}$
- Simply cancel out the matching terms

most of the

Required Properties of p (axioms)

- $p(\emptyset) = 0$ $\quad$ p(all events) = 1
- $p(X) \leq p(Y)$ for any $X \subseteq Y$
- $p(X) + p(Y) = p(X \cup Y)$ provided $X \cap Y=\emptyset$

e.g., p(win & clear) + p(win & $\neg$clear) =



p measures total probability of a set of events

Commas denote conjunction

p(Paul Revere wins, Valentine places, Epitaph shows | weather's clear)

what happens as we add conjuncts to left of bar ?

- probability can only <u>decrease</u>
- numerator of historical estimate likely to go to zero:

$$\frac{\text{\# times Revere wins AND Val places… AND weather's clear}}{\text{\# times weather's clear}}$$

Commas denote conjunction

~~p(Paul Revere wins, Valentine places, Epitaph~~ shows | weather's clear)

p(Paul Revere wins | weather's clear, ground is dry, jockey getting over sprain, Epitaph also in race, Epitaph was recently bought by Gonzalez, race is on May 17, …)

what happens as we add conjuncts to right of bar ?

- probability could increase or decrease
- probability gets more relevant to our case (less *bias*)
- probability *estimate* gets less reliable (more *variance*)

$$\frac{\text{\# times Revere wins AND weather clear AND … it's May 17}}{\text{\# times weather clear AND … it's May 17}}$$

Simplifying Right Side: Backing Off

p(Paul Revere wins | weather's clear, ground is dry, jockey getting over sprai~~n, Epitaph also in race, Epitaph was re~~cently bought by Go~~nzalez, race is on May 17, ...~~)

not exactly what we want but at least we can get a reasonable estimate of it!

(i.e., more bias but less variance)

try to *keep* the conditions that we suspect will have the most influence on whether Paul Revere wins

Simplifying Right Side: Backing Off

p(Paul Revere wins, ~~Valentine places, Epitaph shows~~ | weather's clear)

NOT ALLOWED!

but we can do something similar to help …

Factoring Left Side: The Chain Rule

p(Revere, Valentine, Epitaph | weather's clear) RVEW/W
= p(Revere | Valentine, Epitaph, weather's clear) = RVEW/VEW
* p(Valentine | Epitaph, weather's clear) * VEW/EW
* p(Epitaph | weather's clear) * EW/W

True because numerators cancel against denominators

Makes perfect sense when read from bottom to top

Moves material to right of bar so it can be ignored

If this prob is unchanged by backoff, we say Revere was CONDITIONALLY INDEPENDENT of Valentine and Epitaph (conditioned on the weather's being clear). Often we just ASSUME conditional independence to get the nice product above.

Language ID?

- Let p(X) = probability of text X in English
- Let q(X) = probability of text X in Polish
- Which probability is higher?

"Rabbit and Lukasiewicz are on the menu"

$p(x_1=\mathtt{r}, x_2=\mathtt{a}, x_3=\mathtt{b}, x_4=\mathtt{b}, x_5=\mathtt{i}, x_6=\mathtt{t}, \ldots)$

How do we calculate this?

- Use the chain rule in probability…
- But there's a hitch…

Apply the Chain Rule

$p(\mathbf{x_1=r, x_2=a, x_3=b, x_4=b, x_5=i, x_6=t, \ldots})$	
$= p(x_1=\mathbf{r})$	4470/ 52108
$* \; p(x_2=\mathbf{a} \mid x_1=\mathbf{r})$	395/ 4470
$* \; p(x_3=\mathbf{b} \mid x_1=\mathbf{r}, x_2=\mathbf{a})$	5/ 395
$* \; p(x_4=\mathbf{b} \mid x_1=\mathbf{r}, x_2=\mathbf{a}, x_3=\mathbf{b})$	3/ 5
$* \; p(x_5=\mathbf{i} \mid x_1=\mathbf{r}, x_2=\mathbf{a}, x_3=\mathbf{b}, x_4=\mathbf{b})$	3/ 3
$* \; p(x_6=\mathbf{t} \mid x_1=\mathbf{r}, x_2=\mathbf{a}, x_3=\mathbf{b}, x_4=\mathbf{b}, x_5=\mathbf{i})$	0/ 3
$* \; \ldots \quad = 0$	

counts from Brown corpus

How can we compute this?

- Each of this pr's can be estimated (using frequency counts) from training data
- Can this work?
- No – not in practice…why?

Forming classes

- "*n-gram*" = sequence of n "words"
  - unigram
  - bigram
  - trigram
  - four-gram…
- In language modeling, the conditioning variables are sometimes called the "history" or the "context."
- The Markov assumption says that the prediction is conditionally independent of ancient history, given recent history.
- I.e., we divide all possible histories into equivalence classes based on the recent history

The Markov assumption

- We *approximate* p(word | all previous words) Instead of

  p(rabbit|Follow the white…) we use:

  P(rabbit|white)
- This is a *<u>Markov assumption</u>* where past memory is limited to immediately previous state – just 1 state corresponding to the previous word or tag

N-grams: limiting history – the Markov assumption

- 0^{th} order Markov model: $P(w_i)$ called a <u>unigram</u> model
- 1^{st} order Markov model: $P(w_i \mid w_{i-1})$ called a <u>bigram</u> model
- 2^{nd} order Markov model: $P(w_i \mid w_{i-2}, w_{i-1})$ called a <u>trigram</u> model

Calculation

$p(x_1=\mathbf{r}, x_2=\mathtt{a}, x_3=\mathtt{b}, x_4=\mathtt{b}, x_5=\mathtt{i}, x_6=\mathtt{t}, \ldots)$

$\approx p(x_1=\mathbf{r})$	4470/ 52108
$* \; p(x_2=\mathbf{a} \mid x_1=\mathbf{r})$	395/ 4470
$* \; p(x_3=\mathbf{b} \mid x_1=\mathbf{r}, x_2=\mathbf{a})$	5/ 395
$* \; p(x_4=\mathbf{b} \mid x_2=\mathbf{a}, x_3=\mathbf{b})$	12/ 919
$* \; p(x_5=\mathbf{i} \mid x_3=\mathbf{b}, x_4=\mathbf{b})$	12/ 126
$* \; p(x_6=\mathbf{t} \mid x_4=\mathbf{b}, x_5=\mathbf{i})$	3/ 485

$* \ldots = 7.3\text{e-}10 * \ldots$

counts from Brown corpus

Another Independence Assumption

$p(x_1=\mathtt{h}, x_2=\mathtt{o}, x_3=\mathtt{r}, x_4=\mathtt{s}, x_5=\mathtt{e}, x_6=\mathtt{s}, \ldots)$

$\approx p(x_1=\mathtt{h})$	4470/ 52108
$* \; p(x_2=\mathtt{o} \mid x_1=\mathtt{h})$	395/ 4470
$* \; p(x_i=\mathtt{r} \mid x_{i-2}=\mathtt{h}, x_{i-1}=\mathtt{o})$	1417/ 14765
$* \; p(x_i=\mathtt{s} \mid x_{i-2}=\mathtt{o}, x_{i-1}=\mathtt{r})$	1573/ 26412
$* \; p(x_i=\mathtt{e} \mid x_{i-2}=\mathtt{r}, x_{i-1}=\mathtt{s})$	1610/ 12253
$* \; p(x_i=\mathtt{s} \mid x_{i-2}=\mathtt{s}, x_{i-1}=\mathtt{e})$	2044/ 21250

$* \ldots = 5.4\text{e-}7 * \ldots$

counts from Brown corpus

Simplify the Notation

$p(x_1=\texttt{h}, x_2=\texttt{o}, x_3=\texttt{r}, x_4=\texttt{s}, x_5=\texttt{e}, x_6=\texttt{s}, \ldots)$

$\approx p(x_1=\texttt{h})$	4470/52108
$* \; p(x_2=\texttt{o} \mid x_1=\texttt{h})$	395/ 4470
$* \; p(\texttt{r} \mid \texttt{h}, \texttt{o})$	1417/14765
$* \; p(\texttt{s} \mid \texttt{o}, \texttt{r})$	1573/26412
$* \; p(\texttt{e} \mid \texttt{r}, \texttt{s})$	1610/12253
$* \; p(\texttt{s} \mid \texttt{s}, \texttt{e})$	2044/21250
$* \; \ldots$	

counts from Brown corpus

Simplify the Notation

$p(\mathbf{x_1=r, x_2=a, x_3=b, x_4=b, x_5=i, x_6=t, \ldots})$

$\approx p(\mathbf{r} \mid \mathbf{BOS, BOS})$	4470/52108
$* \; p(\mathbf{a} \mid \mathbf{BOS, r})$	395/ 4470
$* \; p(\mathbf{b} \mid \mathbf{r, a})$	1417/14765
$* \; p(\mathbf{b} \mid \mathbf{a, b})$	1573/26412
$* \; p(\mathbf{i} \mid \mathbf{b, b})$	1610/12253
$* \; p(\mathbf{t} \mid \mathbf{i, b})$	2044/21250
$* \; \ldots$	

the parameters of a trigram generator! Same assumptions about language.

values of those parameters, as naively estimated from Brown corpus.

These basic probabilities are used to define p(rabbit)

counts from Brown corpus

Simplify the Notation

$p(x_1=\texttt{r}, x_2=\texttt{a}, x_3=\texttt{b}, x_4=\texttt{b}, x_5=\texttt{i}, x_6=\texttt{t}, \ldots)$

$\approx t_{\mathbf{BOS, BOS, r}}$

$* \; t_{\mathbf{BOS, r, a}}$

$* \; t_{\mathbf{a, BOS, r}}$

$* \; t_{\mathbf{b, r, s}}$

$* \; t_{\mathbf{r, s, e}}$

$* \; t_{\mathbf{t, b, i}}$

$* \; \ldots$

the parameters of trigram generator! Same assumptions about language

This notation emphasizes that they're just real variables whose value must be estimated

4470/ 52108

395/ 4470

1417/ 14765

1573/ 26412

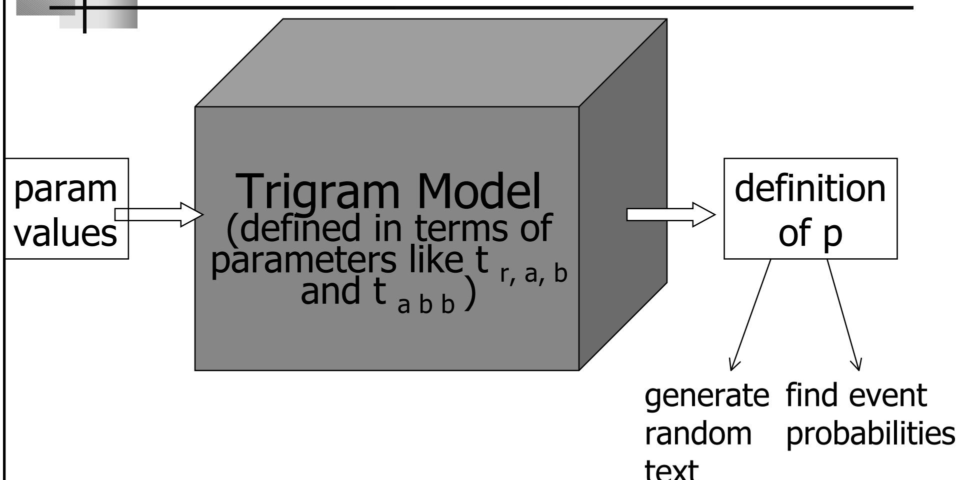
1610/ 12253

2044/ 21250

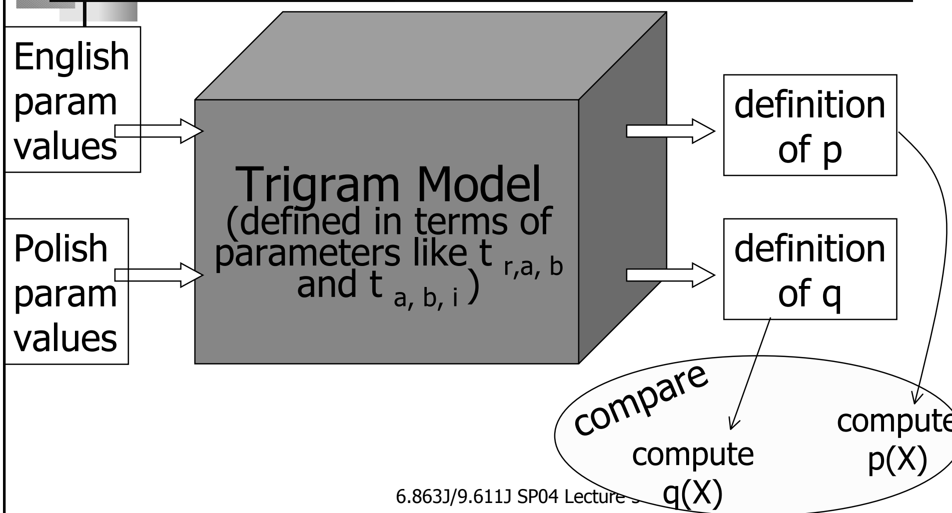
values of those parameters, as naively estimated from Brown corpus.

counts from Brown corpus

Definition: Probability Model

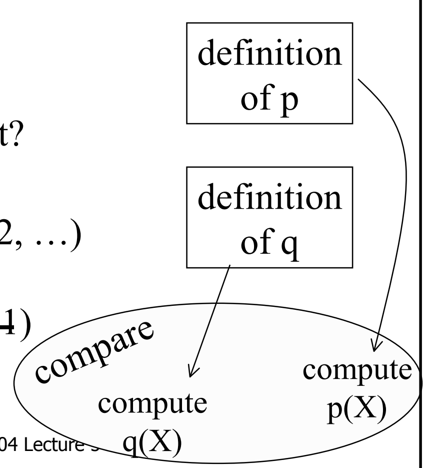


English vs. Polish



What is "X" in p(X)?

- Element of some implicit "event space"
  - e.g., race
  - e.g., sentence
- What if event is a whole text?
  - p(text)
    = p(sentence 1, sentence 2, …)
    = p(sentence 1)
    * p(sentence 2 | ~~sentence 1~~)
    * …



Writing like Jane Austen

- Three novels by Jane Austen: *Emma*, *Sense and Sensibility, Pride and Prejudice*
- Remove punctuation, keep paragraphs
- Train trigram model on this text

Writing like Jane Austen

f(3gram)	f(2gram)	f(1gram)	w_0	w_1	w_2
378	518	10381	I	do	not
366	1366	10381	I	am	sure
214	1917	9182	in	the	world
202	572	6917	she	could	not
189	462	2751	would	have	been
174	184	10381	I	dare	say
173	179	5758	as	soon	as
173	357	11135	a	great	deal
171	332	7573	it	would	be
155	945	3017	could	not	be

Conversion

3gram $\frac{f(w_0,w_1,w_3)}{f(w_0,w_1)}$	2gram $\frac{f(w_0,w_1)}{f(w_0)}$	1gram $\frac{f(w_0)}{N}$	w_0	w_1	w_2
0.72	0.04	0.016	I	do	not
0.26	0.13	0.016	I	am	sure
0.11	0.20	0.014	in	the	world
0.35	0.08	0.011	she	could	not
0.40	0.16	0.004	would	have	been
0.94	0.01	0.016	I	dare	say
0.96	0.03	0.009	as	soon	as
0.48	0.03	0.018	a	great	deal
0.51	0.04	0.012	it	would	be
0.16	0.31	0.004	could	not	be

3-gram

[Genmetheyesse orils of Ted you doorder [6], the Grily Capiduatent pildred and For thy werarme: nomiterst halt i,what production the Covers, in calt cations on wile ars, was name conch rom the exce of the man, Winetwentagaint up, and and Al1. And of Ther so i hundal panite days th the res of th rand ung into the forD six es, wheralf the hie soulsee, frelatche rigat. And the LOperact camen unismelight fammedied: and nople,

4-gram

[1] By the returall benefit han every familitant of all Thou go? And At the eld to parises of the nursed by thy way of all histantly be the ~aciedfag . to the narre gread abrasa of thing, and vas these conwuning clann com to one language; all Lah, which for the greath othey die. -

5-gram

[Gen 3:1] In the called up history of its opposition of bourgeOIS AND Adam to rest, that the existing of heaven; and land the bourgeoiS ANger anything but concealed, the land whethere had doth know ther: bury thy didst of Terature their faces which went masses the old society [2] is the breaks out of oppressor of all which, the prolETARiat goest, unto German pleast twelves applied in manner with t hese, first of this polities have all

3-word-gram

[Gen 4:25] And Adam gave naines to ail feudal, patriarchal, idyllic relations. It bas but -established new classes, new conditions of oppression, new forme of struggle in place of the West? The bourgeoisie keeps more and more splitting up into two great lights; the greater light to rule the day of my house is this Eliezer of Damascus.

How far can we go??