6.863J Natural Language Processing
Lecture 1: Introduction – walking the walk, talking the talk

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

The Menu Bar

- Administrivia
  - All on web page: www.ai.mit.edu/courses/6.863/
- What this course is all about
- What you will learn & what you have to do in the course
- Why NLP is hard, and interesting
- The ingredients of language
- Why language and computation?
- Words, words, words…

What is this course all about?

- Computational methods for working with natural (human) languages
- Applications of computer science & AI
- Linguistic theory
- Natural (psycholinguistics) or artificial computation (natural language processing, NLP)
- Tools for building such systems

A few applications of NLP

- Spelling correction, grammar checking …
- Better search engines
- Information extraction
- Psychotherapy; Harlequin romances; etc.

- New interfaces:
  - Speech recognition (and text-to-speech)
  - Dialogue systems (USS Enterprise onboard computer)
  - Machine translation (the Babel fish)

Goals of the course

- Introduce you to NLP problems & solutions
- Relation to linguistics, cognitive science, & statistics

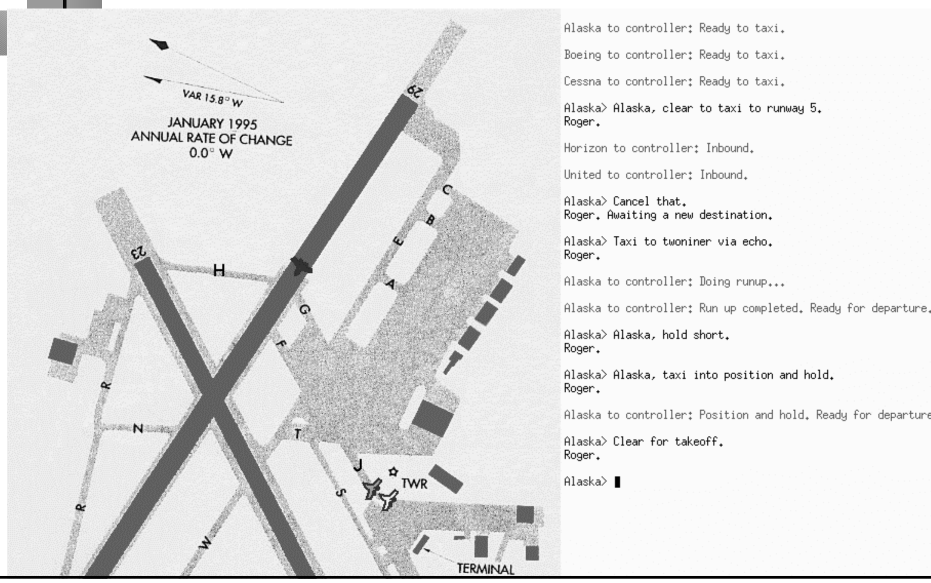
- At the end you should:
  - Agree that language is subtle & interesting
  - Feel some ownership over the formal & statistical models
  - Understand research papers in the field

What you will be able to do

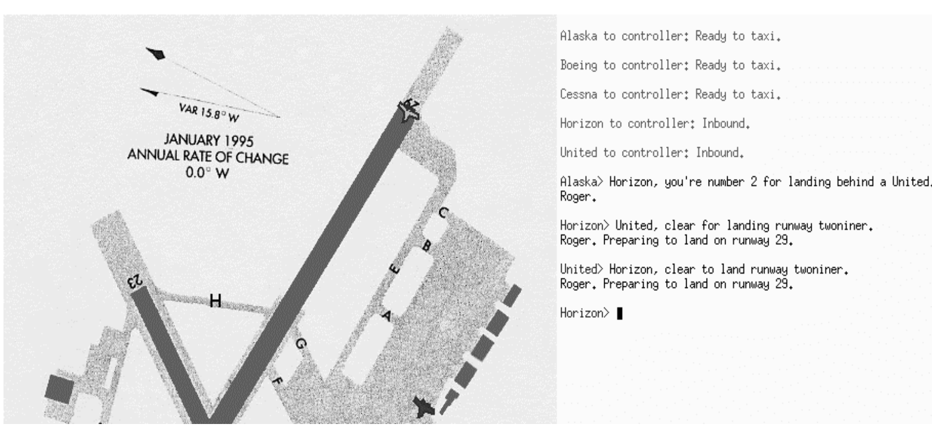
- Build your own natural language systems

- Become the next Google…?

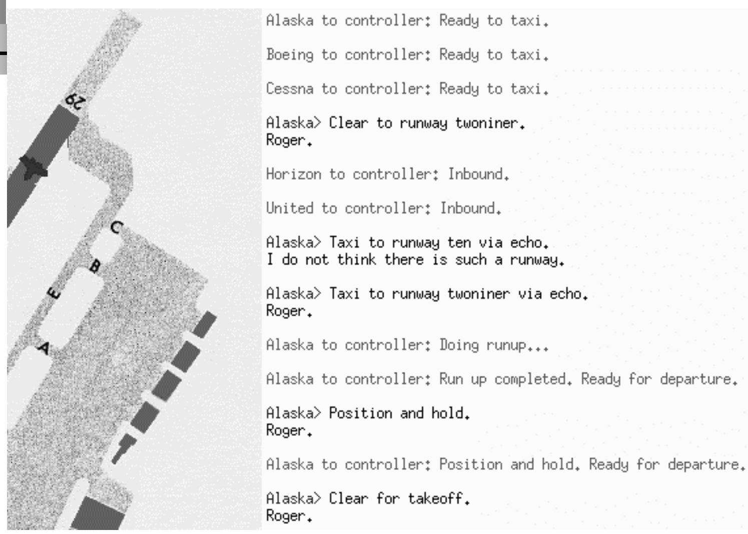
Build your own system



Landing is always harder…



And then there's…



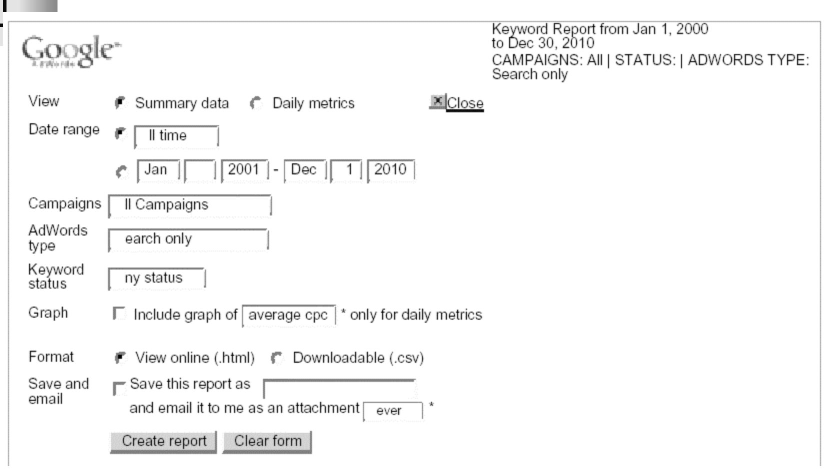
The next Google? Words, words, words

- How many (English) words on the web? Est. from 2001
- 76,598,718,000
- Many repetitive repetitive words

The next Google? Words, words, words

- Keywords match to match search terms to advertisers: ‘hair follicle’ →‘hair follicle stimulation’ → Rogaine
- Advertisers pay Google: $3B for keyword advertising in 2003, expected to grow to over $8B
- Easy to guess some: ‘Rogaine’, ‘hair loss’
- Hard to guess others, very low frequency: ‘hair falling out’
- Can’t maximize marketing reach

Actual example



Example – too low frequency

DOWNLOAD (.csv) Save this report as [ ] and email me this report: ever Submit

Keyword	Keyword Matching	Keyword Status	Destination URL	Ad Group	Campaign	Maximum CPC	Impressions	Clicks	CTR	Avg CPC	Cos
Totals and Overall Averages:							54,409	988	1.82%	$1.11	$1,100.8
PostgreSQL web hosting	Keyword	Active	default URL	Ad Group #1	Google Trial	$10.00	2	0	0.00%	$0.00	$0.0
apache postgresql	Keyword	Active	default URL	Ad Group #1	Campaign #2	$5.00	1,188	6	0.51%	$0.43	$2.5
business web site	Keyword	Active	default URL	Ad Group #1	Campaign #3	$5.00	1	0	0.00%	$0.00	$0.0
challenger circuit breaker	Keyword	Active	default URL	Ad Group #1	Campaign #4	$5.00	240	36	15.00%	$1.12	$40.4
challenger circuit breakers	Keyword	Active	default URL	Ad Group #1	Campaign #4	$5.00	250	32	12.80%	$0.83	$26.4
cheap web design	Keyword	Active	default URL	Ad Group #1	Campaign #3	$5.00	2	0	0.00%	$0.00	$0.0
free hosting	Keyword	Active	default URL	Ad Group	Campaign	$5.00	571	30	5.25%	$1.95	$58.3

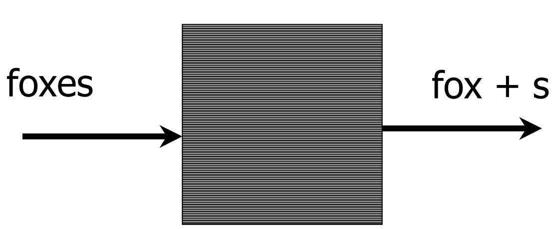
Goal

- Input: “Rogaine is a drug that helps prevent hair loss”
- Output: keywords causally relevant: “stop hair loss”; “hair loss”; “hair falling out”

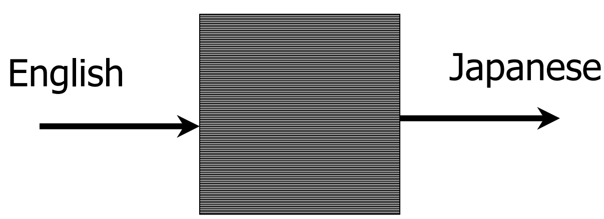
Lightweight vs. AI-complete natural language problems

But what's inside the black box? Lightweight / AI-complete

Lightweight:



AI-complete



But the most important reason of all…

It’s the year 2004 and still…still….

Dave Bowman: Open the pod bay doors, HAL
HAL: I’m sorry Dave, I’m afraid I can’t do that.

Natural language at the heart of human intelligence

- The first Turing test:

“Rabbah Zoreh made a Gollum and brought it to Rabbah; he bid it to talk.

Rabbah replied: ‘It cannot speak; return it unto the flames’

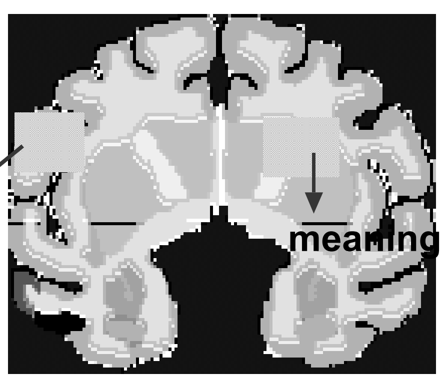
(Manhet Sahedrin, Babylonian Talmud, approx. 400 BCE)

‘Simple’ model: sound-meaning relation

sound →  → ‘meaning’

Aristotle, e.g., only 2500 years old…

Language = Pairing sound & meaning



sound (outside)

meaning (inside world)

Only interfaces ‘remain’

Why is nlp hard?

- Human language is <u>not</u> 'wysiwyg'
- Human language is <u>ambiguous</u>

Why is nlp hard?

- Human language is <u>not</u> 'wysiwyg'
  - What you see 'on the surface' is <u>not</u> the 'underlying representation' people (or computers) manipulate
  - You can't just write down the representation from a simple surface examination of the data
  - (Compare: edges in machine vision)

Example of hidden structure

- English plural:
- Toy+s → toyz ; add *z*
- Book+s → books ; add *s*
- Church+s → churchiz ; add *iz*
- Box+s → boxiz ; add *iz*
- (Sheep+s → sheep/sheeps ; add nothing; or *s*
- What if a *novel* word?
  - Bach's many cantatas
  - Which pronounciation is it? *S* or IZ ?

Bachs many cantatas NOT BachIZ despite

Analogy/similarity to 'box' - why?

Invisible knowledge

- I want to hold your hand
- I wanna hold your hand
- Displacement: I understand these students
- These students I understand
- I want these students to solve the problem
- These students I want [x] to solve the problem [x]= these students

Notice that contraction of want+to *is now blocked!*

Figuring out the right representation

- *What* is the right representation (knowledge of language)
- This is what linguists figure out
- This alone is a daunting task…but wait, there's more
- Computation = data structures + algorithms

Sentence knowledge is subtle

- A book was given Mary
- Mary was given a book
- A book was given to Mary
- Mary was given a book to

Word knowledge is subtle

- He arrived at the lecture
- He chuckled at the lecture

- He arrived drunk
- He chuckled drunk

- He chuckled his way through the lecture
- He arrived his way through the lecture

What is the character of this knowledge?

- Some of it must be memorized (obviously so):
  - Singing→ Sing+ing; Bringing → bring+ing
  - Cantare, portare; singen, holen
- We must know the endings (suffixes) of words
- What else?
- We must know the roots (stems) of words
  - Duckling → Duckl + ing ?
- What else?
  - We must know which endings go on which roots
  - Doer → do+er
  - Beer → ??

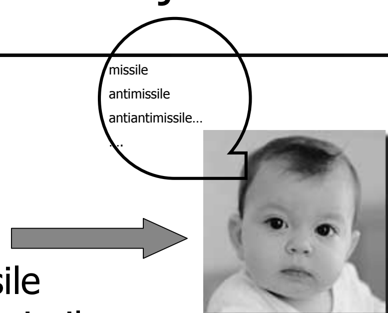
But there is too much to memorize!

- Missile
- Antimissile
- Antiantimissile
- Antiantiantimissile…

- Conclusion: we must have a <u>generative</u> system – i.e., a set of <u>rules</u> to do this

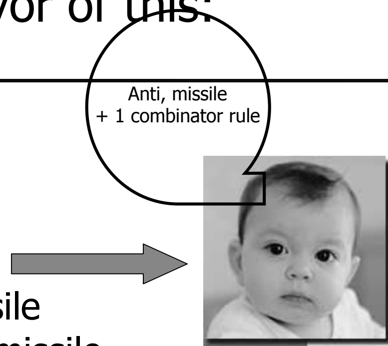
So we can reject this:

Sound:
missile
antimissile
antiantimissile…



In favor of this:

Sound:
missile
antimissile
antiantimissile…



The non-WYSIWYG view!

Other languages…

```
Lexical:  Paris+mut+nngau+juma+niraq+lauq+sima+nngit+junga
Surface:  Pari  mu  nngau juma nira  lauq sima nngit tunga

          Paris     = (root = Paris)
          +mut      = terminalis case ending
          +nngau    = go (verbalizer)
          +juma     = want
          +niraq    = declare (that)
          +lauq     = past
          +sima     = (added to -lauq- indicates "distant past")
          +nngit    = negative
          +junga    = 1st person sing. present indic (nonspecific)
```

Figure 2: Inuktitut: *Parimunngaujumaniralauqsimanngittunga* = “I never said I wanted to go to Paris”

An ancient tradition

- Insight of P̄anini (Sanskrit grammarians): circa 400BCE: system of morphological analysis, based on cascaded rules (we will see how to implement this later on)
- Nice to have whole book written to reveal this published in year 2000
- Still, have we made progress in the intervening two millennia…?

Panini

- *Astadhyayi: (400-700BCE?)* Panini gives formal production rules and definitions to describe Sanskrit grammar.
- Starting with about 1700 basic elements like nouns, verbs, vowels, consonants he put them into classes. The construction of sentences, compound nouns etc. is explained as ordered rules operating on underlying structure

What's more…

- On the basis of just under 4000 sutras [rules expressed as aphorisms], he built virtually the whole structure of the Sanskrit language
- Uses a notation precisely as powerful as Backus normal form - an algebraic notation to represent numeral (and other patterns) by letters
- So, we know something about <u>what</u> the representation for language might be

Figuring out the right algorithm

- *How* is that knowledge is put to use?
- *What* and *how* are the cornerstones – the key questions

How can be difficult – why?

- Police police police

- I know that
- I know that block
- I know that blocks the sun
- I know that block blocks the sun
- In a word: ambiguity

Ambiguity

- Iraqi Head Seeks Arms
- Juvenile Court to Try Shooting Defendant
- Teacher Strikes Idle Kids
- Stolen Painting Found by Tree
- Kids Make Nutritious Snacks
- Local HS Dropouts Cut in Half
- Obesity Study Looks for Larger Test Group

Ambiguity

- British Left Waffles on Falkland Islands
- Red Tape Holds Up New Bridges
- Man Struck by Lightning Faces Battery Charge
- Bush Wins on Budget, but More Lies Ahead
- Hospitals Are Sued by 7 Foot Doctors

Subtler Ambiguity

- Q: Why does my high school give me a suspension for skipping class?
- A: Administrative error. They're supposed to give you a suspension for auto shop, and a jump rope for skipping class. (*rim shot*)

What’s hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

What’s hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

To get a donut (spare tire) for his car?

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

store where donuts shop? or is run by donuts?

or looks like a big donut? or made of donut?

or has an emptiness at its core?

What's hard about this story?

I stopped smoking freshman year, but

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

Describes where the store is? Or when he stopped?

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

Well, actually, he stopped there from hunger and exhaustion, not just from work.

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

At that moment, or habitually?

(Similarly: Mozart composed music.)

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

That's how often he thought it?

What’s hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

But actually, a coffee only stays good for about 10 minutes before it gets cold.

What’s hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

the particular coffee that was good every few hours? the donut store? the situation?

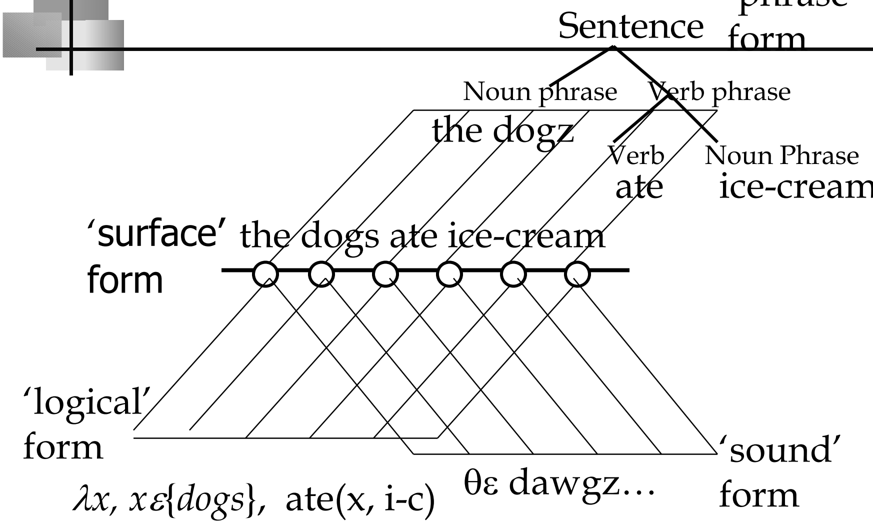
What's hard about this story?

John stopped at the donut store on his way home from work. He thought a coffee was good every few hours. But it turned out to be too expensive there.

too expensive for what? what are we supposed to conclude about what John did?

how do we connect "it" to "expensive"?

The "spiral notebook" picture



Levels of language

- Phonetics/phonology/morphology: what words (or subwords) are we dealing with?
- Syntax: What phrases are we dealing with? Which words modify one another?
- Semantics: What's the literal meaning?
- Pragmatics: What should you conclude from the fact that I said something? How should you react?

Levels of representation

- Primitives
- Rules of combination (syntax – from Greek σψνταξισ, 'too arrange together')
- Generative system to produce expressions in the representation language
- Examples: words, phrases,….

The basic computational problem

- Mapping from (external) representation to an (internal) representation
- True for *all* representational levels of
- Examples: cats→ cat-Noun-Plural)

  cat-Noun-Plural → cats

  the cat slept → Noun phrase Verb

The central problem: parsing

- The problem of mapping from one representational level to another is called <u>parsing</u>
- If there is > 1 possible outcome (the mapping is not a function) then the input expression is <u>ambiguous</u>

dogs $\rightarrow$ dog-Noun-plural <u>or</u>
dog-Verb-presT

Word parsing

- We begin here: Lab 1
- Why?

Start with words: they illustrate all the problems (and solutions) in NLP

- Parsing words
  Cats → CAT + N(oun) + PL(ural)
- Used in:
  - Traditional NLP applications
  - Finding word boundaries (e.g., Latin, Chinese)
  - Text to speech (*boathouse*)
  - Document retrieval (example next slide)
- In particular, all the problems of *parsing*, *ambiguity,*and *computational efficiency* arise (as well as the problems of *how people do it*)

Example from information retrieval

- Keywork retrieval: *marsupial* or *kangaroo* or *koala*
- Trying to form equivalence classes - ending not important
- Can try to do this without *extensive* knowledge, but then:

organization → organ   Europe~~an~~ → Europe
generalization → generic   noise → noisy

Morphology

- <u>Morphology</u> is the study of how *words* are *built up* from smaller *meaningful* units called *morphemes* (morph= shape; logos=word)

What about other languages?

Present indicative	Imper	Imperf Indic.	Future	Preterite	Present Subjun	Cond	Imp. Subj.	Future Subj.
amo		*amaba*	*amaré*	*amé*	*ame*	*amaría*	*amara*	*amare*
amas	*ama*	*amabas*	*amarás*	*amaste*	*ames*	*amarías*	*amaras*	*amares*
	ames							
ama		*amamba*	*amará*	*amó*	*ame*	*amaría*	*amara*	*amáreme*
amamos								
amáis	*amad*	*amambais*	*amremos*	*amomos*	*amemos*	*amaríanos*	*amarais*	*amareis*
	amáis							
aman		*amamban*	*amarán*	*amaron*	*amen*	*amarían*	*amarain*	*amaren*

How to love in Spanish…incomplete…you can finish it after Valentine's Day…

What about other processes?

- Stem: core meaning unit (morpheme) of a word
- Affixes: bits and pieces that combine with the stem to modify its meaning and grammatical functions

  Prefix: *un- , anti-,* etc.

  Suffix: -*ity, -ation,* etc.

  Infix:

  Tagalog: *um+hinigi* → *h<u>um</u>ingi* (borrow)

  Any infixes in 'nonexotic' language like English?

Here's one: *un-f******-believable*

OK, now how do we deal with this computationally?

- <u>*What*</u> knowledge do we need?
- <u>*How*</u> is that knowledge put to use?

- <u>*What:*</u>

  duckling; beer (implies what K…?)

  chase + *ed* → *chased* (implies what K?)

  breakable + un →*unbreakable* ('prefix')
- <u>*How:*</u> a bit trickier, but clearly we are at least doing this kind of mapping…

Why not a great big dictionary?

1. Impractical: some languages associate a single meaning w/ a Sagan number of distinct surface forms (600 billion in Turkish)
2. Chinese compounding: about 3000 'words,' combine to yield tens of thousands
3. Speakers don't represent words as a list

   Wug test (Berko, 1958)

   Juvenate is rejected slower than *pertoire* (real prefix matters)

Two parts to the "what"

1. Which units can glue to which others (roots and affixes) (or stems and affixes)= "morphotactics"
2. What 'spelling changes' (orthographic changes) occur – like dropping the *e* in 'chase + ed' (morpheme 'shape' depends on its context – like plural)

IDEA: MODEL EACH AS A FINITE-STATE MACHINE, then combine.

WHY is this a good model?

What are we modeling – part I

- Linear arrangement of morphemes – beads on a string:

Lebensversichergungesellschaftsangestellter

Leben+s+versichergun+gesellschaft+s+angestellter

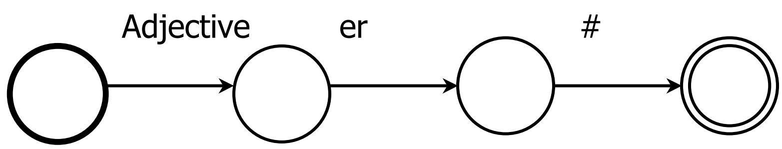
life+CmpAug+insurance+CmpAug+company+CompAug+employee

English examples

- As an example, consider adjectives

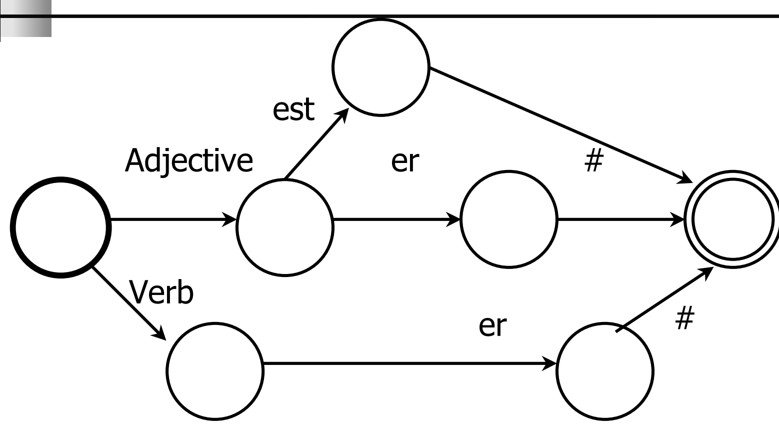
  Big, bigger, biggest
  Cool, cooler, coolest, coolly
  Red, redder, reddest
  Clear, clearer, clearest, clearly, unclear, unclearly
  Happy, happier, happiest, happily
  Unhappy, unhappier, unhappiest, unhappily
  Real, unreal, silly

Finite-state machine

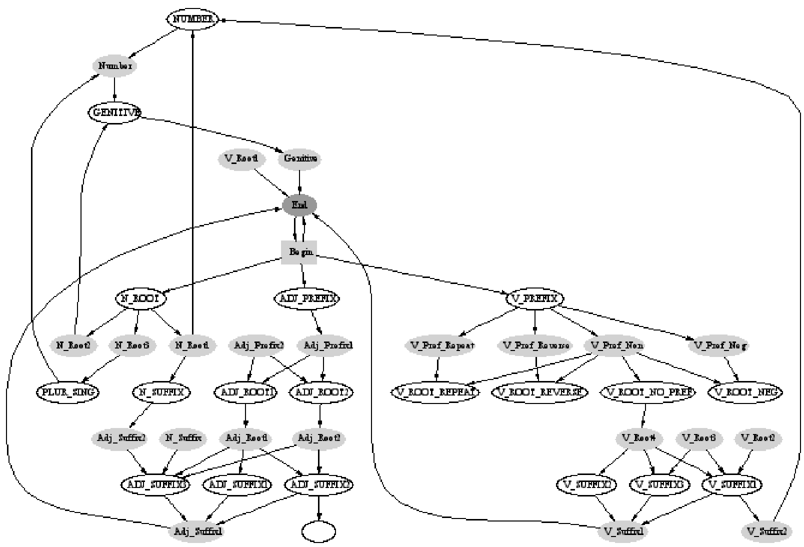


Pure model of linear concatenation

More states & arcs for more discrimination



Linear concatenation of morphemes as fsa



What are we modeling – part 2

- Morphemes surface differently in different contexts
- Fox+s $\rightarrow$ foxes; fly+s $\rightarrow$flies; quiz+s $\rightarrow$quizzes; dog+s $\rightarrow$ dogs
- We model this as an extension of std finite-state machine
- Do this in two steps: (1) since the words are spelled out as characters, form an fsa for spelling, a <u>prefix tree</u> automaton:

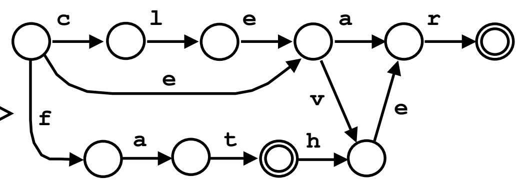
Representing possible spell-out

Wordlist

`clear`
`clever`
`ear`
`ever`
`fat`
`father`

compile →

Network



`/usr/dict/words`

25K words
206K chars

2 sec →

FSM

17728 states,
37100 arcs

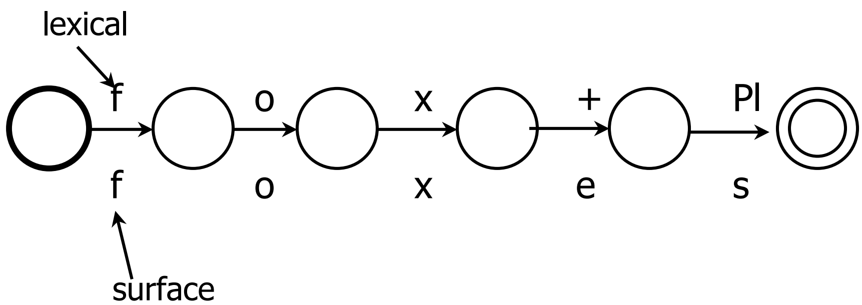
Representing spelling changes

- This gives the <u>surface</u> spelling
- Now we add a second pair on the arc labels to specify what the <u>underlying</u> or <u>lexical</u> 'spelling' is
- Example: d o g s (surface)

  d o g PL (lexical)

This is called a finite-state <u>transducer</u>

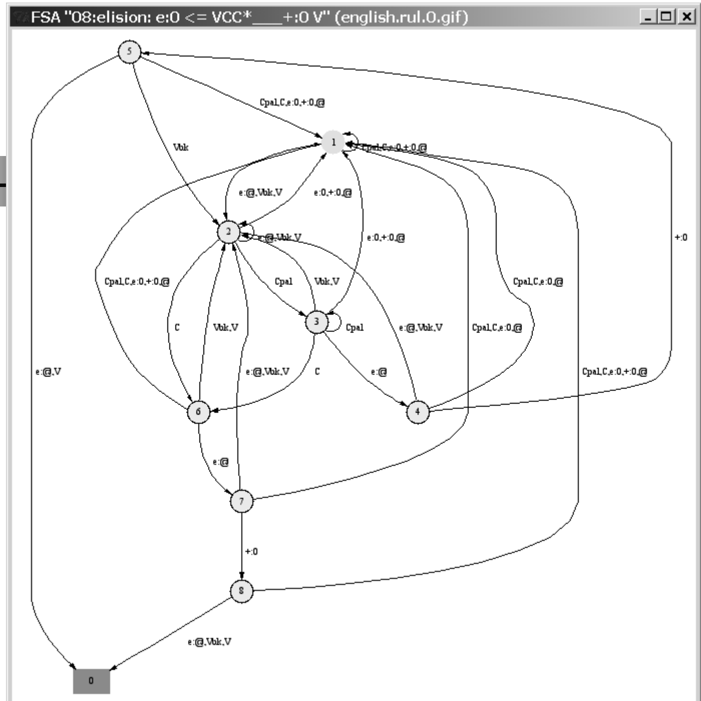
Finite state transducer (fst) as way to represent lexical/surface relation



(lexical, surface) pairs for transitions
Alternative notation: (f:f), (+:e)

How was this done in linguistic theory?

- Insert e after ‘sh’, ‘x’, etc: “epthenthesis”
- Statement of rule is actually quite complex:
  - Rewrite rule: $x \rightarrow y \mid \alpha\_\_\_\beta$ (Chomsky & Halle)
  - 0:e ==> [Csib (c h) (s h) y:i] +:0 _ s (transducer notation)



Ah, but there's more than 1 change..

- quiz+s → quizzes
- And… they interact: spy+s
- What do we do?
- Ans: more than one FTN, one for <u>each</u> kind of change
  - Epenthesis (e insertion)
  - consonant doubling (gemination)

Example from English

underlying quiz + s

↓ Rule A: *s -> es after z*

intermediate quiz + es

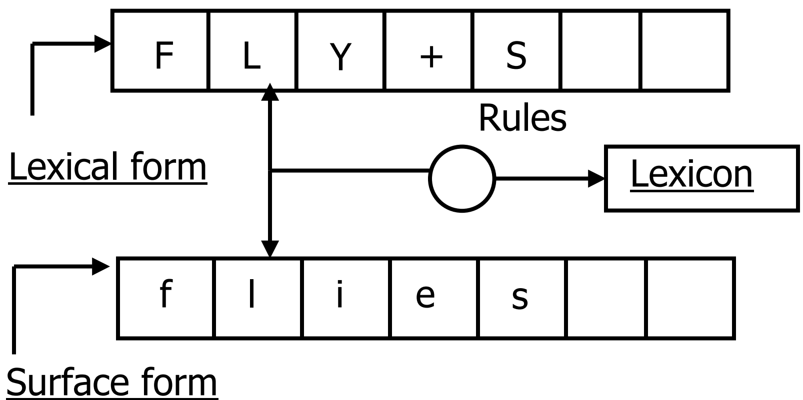
↓ Rule B: *z doubles before Suffix beginning with vowel*

underlying quizzes

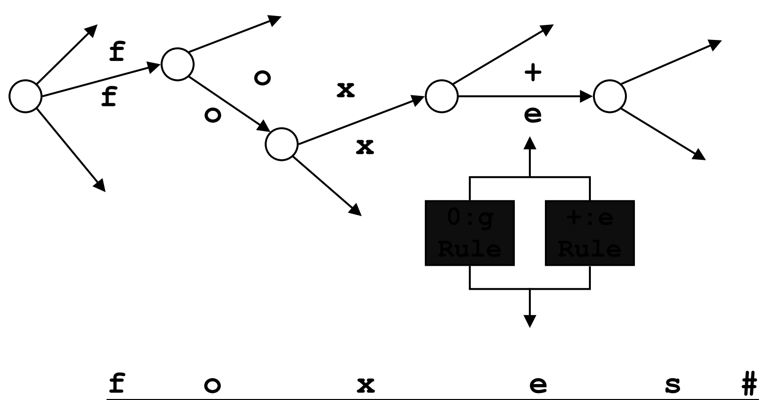
How to handle > 1 rule?

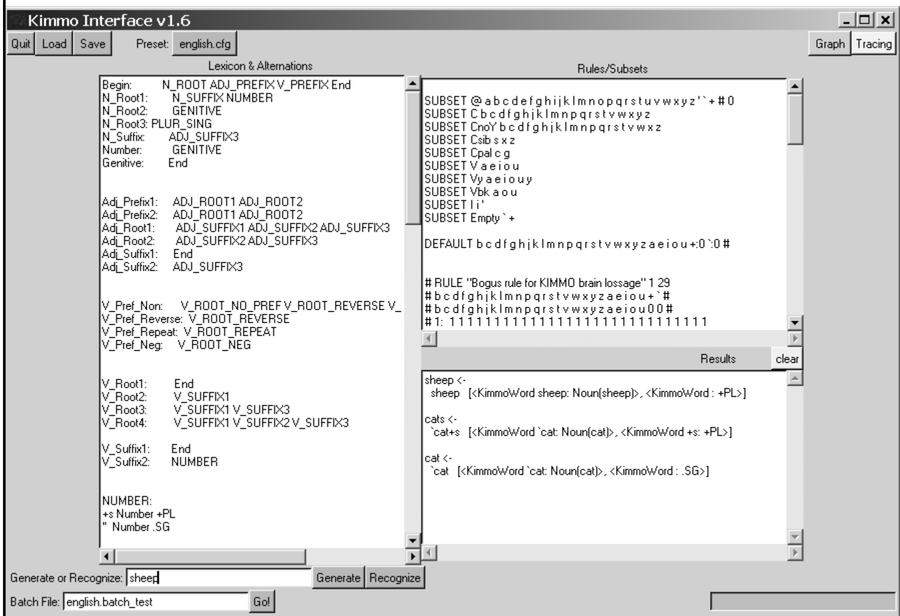
- Ans: intersect FTNs…
- So we get this picture…

Two-level morphology

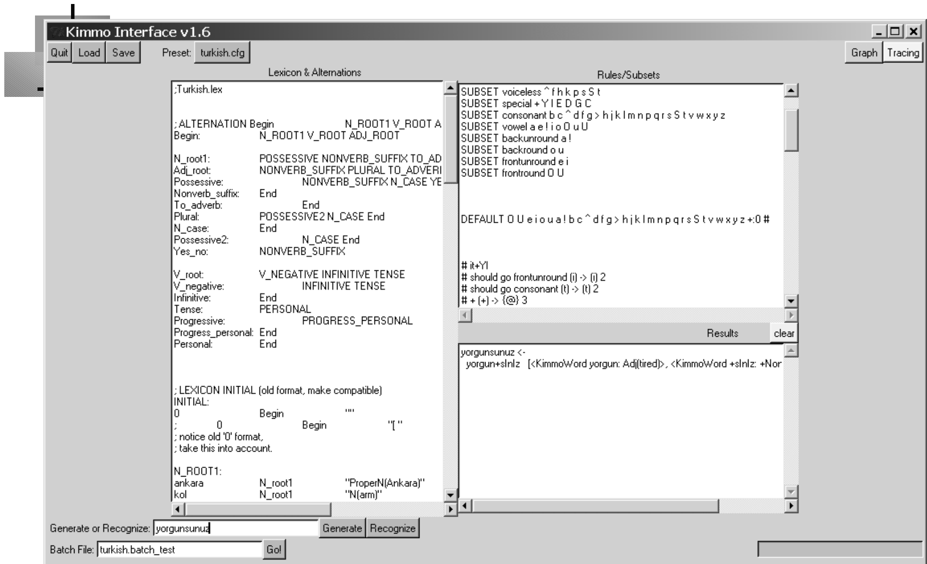


Lookup and Analysis in Tandem




Kimmo Interface v1.6
Quit Load Save Preset: english.cfg
Graph Tracing
Lexicon & Alternations
Begin: N_ROOT ADJ_PREFIX V_PREFIX End
N_Root1: N_SUFFIX NUMBER
N_Root2: GENITIVE
N_Root3: PLUR_SING
N_Suffix: ADJ_SUFFIX3
Number: GENITIVE
Genitive: End
Adj_Prefix1: ADJ_ROOT1 ADJ_ROOT2
Adj_Prefix2: ADJ_ROOT1 ADJ_ROOT2
Adj_Root1: ADJ_SUFFIX1 ADJ_SUFFIX2 ADJ_SUFFIX3
Adj_Root2: ADJ_SUFFIX2 ADJ_SUFFIX3
Adj_Suffix1: End
Adj_Suffix2: ADJ_SUFFIX3
V_Pref_Non: V_ROOT_NO_PREF V_ROOT_REVERSE V_
V_Pref_Reverse: V_ROOT_REVERSE
V_Pref_Repeat: V_ROOT_REPEAT
V_Pref_Neg: V_ROOT_NEG
V_Root1: End
V_Root2: V_SUFFIX1
V_Root3: V_SUFFIX1 V_SUFFIX3
V_Root4: V_SUFFIX1 V_SUFFIX2 V_SUFFIX3
V_Suffix1: End
V_Suffix2: NUMBER
NUMBER:
+s Number +PL
'' Number .SG
Rules/Subsets
SUBSET @ a b c d e f g h i j k l m n o p q r s t u v w x y z ' ` + # 0
SUBSET C b c d f g h j k l m n p q r s t v w x y z
SUBSET CnoY b c d f g h j k l m n p q r s t v w x z
SUBSET Csib s x z
SUBSET Cpal c g
SUBSET V a e i o u
SUBSET Vy a e i o u y
SUBSET Vbk a o u
SUBSET I i '
SUBSET Empty ` +
DEFAULT b c d f g h j k l m n p q r s t v w x y z a e i o u +:0 `:0 #
RULE "Bogus rule for KIMMO brain lossage" 1 29
b c d f g h j k l m n p q r s t v w x y z a e i o u + `
b c d f g h j k l m n p q r s t v w x y z a e i o u 0 0
1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Results clear
sheep <-
sheep [<KimmoWord sheep: Noun(sheep)>, <KimmoWord : +PL>]
cats <-
`cat+s [<KimmoWord `cat: Noun(cat)>, <KimmoWord +s: +PL>]
cat <-
`cat [<KimmoWord `cat: Noun(cat)>, <KimmoWord : .SG>]
Generate or Recognize: sheep
Generate Recognize
Batch File: english.batch_test
Go!

…and Turkish…


Kimmo Interface v1.6
Quit Load Save Preset: turkish.cfg
Graph Tracing
Lexicon & Alternations
;Turkish.lex
;ALTERNATION Begin N_ROOT1 V_ROOT A
Begin: N_ROOT1 V_ROOT ADJ_ROOT
N_root1: POSSESSIVE NONVERB_SUFFIX TO_AD
Adj_root: NONVERB_SUFFIX PLURAL TO_ADVERI
Possessive: NONVERB_SUFFIX N_CASE YE
Nonverb_suffix: End
To_adverb: End
Plural: POSSESSIVE2 N_CASE End
N_case: End
Possessive2: N_CASE End
Yes_no: NONVERB_SUFFIX
V_root: V_NEGATIVE INFINITIVE TENSE
V_negative: INFINITIVE TENSE
Infinitive: End
Tense: PERSONAL
Progressive: PROGRESS_PERSONAL
Progress_personal: End
Personal: End
; LEXICON INITIAL (old format, make compatible)
INITIAL:
0 Begin ""
; 0 Begin "["
; notice old '0' format,
; take this into account.
N_ROOT1:
ankara N_root1 "ProperN(Ankara)"
kol N_root1 "N(arm)"
Rules/Subsets
SUBSET voiceless ^ f h k p s S t
SUBSET special + Y I E D G C
SUBSET consonant b c ^ d f g > h j k l m n p q r s S t v w x y z
SUBSET vowel a e ! i o O u U
SUBSET backunround a !
SUBSET backround o u
SUBSET frontunround e i
SUBSET frontround O U
DEFAULT O U e i o u a ! b c ^ d f g > h j k l m n p q r s S t v w x y z +:0 #
it+YI
should go frontunround (i) -> (i) 2
should go consonant (t) -> (t) 2
+ (+) -> (@) 3
Results clear
yorgunsunuz <-
yorgun+sInIz [<KimmoWord yorgun: Adj(tired)>, <KimmoWord +sInIz: +Nor
Generate or Recognize: yorgunsunuz
Generate Recognize
Batch File: turkish.batch_test
Go!