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

Alaska to controller: Ready to taxi.
Boston to controller: Ready to taxi.
Cape to controller: Ready to taxi.
Alaska to controller: Clear to taxi to runway 5.
Roger.
Horizon to controller: Inbound.
United to controller: Inbound.
Alaska to controller: Inbound.
Roger, awaiting a new destination.
Alaska to controller: Taxi to taxiway via echo.
Roger.
Alaska to controller: Doing runup....
Alaska to controller: Runup completed. Ready for departure.
Alaska to controller: Hold short.
Roger.
Alaska to controller: Taxi into position and hold.
Roger.
Alaska to controller: Position and hold. Ready for departure.
Alaska to controller: Clear for takeoff.
Roger.
Alaska: \[z\]

Landing is always harder...

Alaska to controller: Ready to taxi.
Boston to controller: Ready to taxi.
Cape to controller: Ready to taxi.
Horizon to controller: Inbound.
United to controller: Inbound.
Alaska to controller: Horizon, you're number 2 for landing behind a United.
Roger.
Horizon's United, clear for landing runway twenty.
Roger, preparing to land on runway 25.
United's Horizon, clear for landing runway twenty.
Roger, preparing to land on runway 05.
Horizon's: \[z\]
And then there’s...

- Alaska to controller: Ready to taxi.
- Taxiing to controller: Ready to taxi.
- George to controller: Ready to taxi.
- Alaska: Clear to runway two-five, three.
- Roger.
- Horizon to controller: Inbound.
- United to controller: Inbound.
- Alaska: Taxi to runway ten via echo, I do not think there is such a runway.
- Alaska: Taxi to runway twenty via echo. Roger.
- Alaska to controller: Taxiing runway...
- Alaska to controller: Run up completed, ready for departure.
- Alaska: Position and hold.
- Roger.
- Alaska to controller: Position and hold, ready for departure.
- Alaska: Clear for takeoff.
- Roger.

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

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:

- foxes → black box → fox + s
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

Aristotle, e.g., only 2500 years old...

Language = Pairing sound & meaning

Only interfaces ‘remain’
Why is NLP hard?

- Human language is not 'wysiwyg'
- Human language is ambiguous

Why is NLP hard?

- Human language is not 'wysiwyg'
  - What you see 'on the surface' is not 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 pronunciation is it?  S or IZ ?

Bach’s many cantatas NOT BachIZ despiteAnalogy/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 *generative* system – i.e., a set of *rules* 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 + ngau + juma + miraq + lauq + sima + nngit + junga
Surface: Pari mu ngau juma miraq lauq sima nngit tunga

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

Figure 2: Inusitatu: Parinanganmumuninirinlanuqsimannngittunga = "I never said I wanted to go to Paris"
An ancient tradition

- Insight of Panini (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 what 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.

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?

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.

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

too expensive for what? what are we supposed to conclude about what John did? how do we connect “it” to “expensive”?
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, ‘to 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 \(\rightarrow\) cat-Noun-Plural
  cat-Noun-Plural \(\rightarrow\) cats
  the cat slept \(\rightarrow\) Noun phrase Verb
The central problem: parsing

- The problem of mapping from one representational level to another is called parsing.
- If there is > 1 possible outcome (the mapping is not a function) then the input expression is ambiguous.
  
  dogs → dog-Noun-plural or
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

- Keyword 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
  European → Europe
  generalization → generic
  noise → noisy
Morphology

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

What about other languages?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>amor</td>
<td>amas</td>
<td>ama</td>
<td>amaro</td>
<td>ame</td>
<td>amena</td>
<td>amas</td>
<td>amas</td>
<td>amare</td>
</tr>
<tr>
<td>amas</td>
<td>ama</td>
<td>amas</td>
<td>amas</td>
<td>ame</td>
<td>amaras</td>
<td>ames</td>
<td>amas</td>
<td>amares</td>
</tr>
<tr>
<td>ames</td>
<td></td>
<td>amas</td>
<td>ames</td>
<td>ame</td>
<td>amarias</td>
<td>amas</td>
<td>amas</td>
<td>amares</td>
</tr>
<tr>
<td>ama</td>
<td>amamba</td>
<td>amarai</td>
<td>amari</td>
<td>ame</td>
<td>amara</td>
<td>amas</td>
<td>amas</td>
<td>amareme</td>
</tr>
<tr>
<td>amamos</td>
<td></td>
<td>amas</td>
<td>amas</td>
<td>ame</td>
<td>amarais</td>
<td>amas</td>
<td>amas</td>
<td>amareis</td>
</tr>
<tr>
<td>amais</td>
<td>amad</td>
<td>amamais</td>
<td>amamos</td>
<td>amemos</td>
<td>amarianos</td>
<td>amrais</td>
<td>amaras</td>
<td>amareis</td>
</tr>
<tr>
<td>aman</td>
<td>amaban</td>
<td>amaran</td>
<td>amaron</td>
<td>amen</td>
<td>amarián</td>
<td>amarain</td>
<td>amarín</td>
<td>amaren</td>
</tr>
</tbody>
</table>

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: \textit{un-}, \textit{anti-}, etc.
  - Suffix: \textit{-ity}, \textit{-ation}, etc.
  - Infix:
    - Tagalog: \textit{um}+\textit{hinigi} \rightarrow \textit{humingi} (borrow)

Any infixes in 'nonexotic' language like English?

Here’s one: \textit{un-f*****-believable}

OK, now how do we deal with this computationally?

- \textit{What} knowledge do we need?
- \textit{How} is that knowledge put to use?

- \textit{What}:
  - \textit{duckling}; \textit{beer} (implies what K...?)
  - \textit{chase} + \textit{ed} \rightarrow \textit{chased} (implies what K?)
  - \textit{breakable} + \textit{un} \rightarrow \textit{unbreakable} (‘prefix’)
- \textit{How}: 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:

  Lebensversicherungsgesellschaftsangestellter
  Leben+s+versichergun+gesellschaft+s+angestellter
  life+CmpAug+insurance+CmpAug+company+CmpAug+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

Adjective → er → #

Pure model of linear concatenation

More states & arcs for more discrimination

Adjective → er → #

Verb → er → #
Linear concatenation of morphemes as fsa

What are we modeling – part 2

- Morphemes surface differently in different contexts
- Fox+s → foxes; fly+s → flies; quiz+s → quizzes; dog+s → 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 prefix tree automaton:
Representing possible spell-out

Wordlist

clear  clever  ear  ever  fat  father

/usr/dict/words  2 sec
25K words  37100 arcs

Network

FSM
17728 states, 37100 arcs

Representing spelling changes

- This gives the surface spelling
- Now we add a second pair on the arc labels to specify what the underlying or lexical ‘spelling’ is
- Example: dog s (surface)
  
dog PL (lexical)

This is called a finite-state transducer
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: “epenthesis”
- Statement of rule is actually quite complex:
  - Rewrite rule: $x \rightarrow y \mid \alpha \_\_\_\_\_\_\_\beta$ (Chomsky & Halle)
  - $0:e \Rightarrow [\text{Csib} (c\ h) (s\ h) y:i] +:0\ _\ s$ (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 each kind of change
  - Epenthesis (e insertion)
  - Consonant doubling (gemination)
Example from English

Rule A: \( s \rightarrow es \) after \( z \)

Rule B: \( z \) doubles before
Suffix beginning with vowel

How to handle > 1 rule?

- Ans: intersect FTNs...
- So we get this picture...
Two-level morphology

- Lexical form
- Rules
- Surface form

Lookup and Analysis in Tandem

- Diagram showing the process of lookup and analysis in tandem