Natural Language Processing
Introduction
Course materials and Acknowledgements

- **Book**: SPEECH and LANGUAGE PROCESSING

- **Other links**:
  - [http://www.cs.colorado.edu/~martin/slp2.html](http://www.cs.colorado.edu/~martin/slp2.html)
  - [http://www.stanford.edu/class/cs224s/](http://www.stanford.edu/class/cs224s/)

- **Course material on**:
  - [http://twiki.di.uniroma1.it/twiki/view/NLP/WebHome](http://twiki.di.uniroma1.it/twiki/view/NLP/WebHome)
Course organization

• Each lesson starts with a 15min questionnaire on one of previous lessons topics (assigned readings)
• Simple projects using Sphinx speech understanding and Stanford parser
Natural Language Processing

• “NLP is the branch of computer science focused on developing systems that allow computers to communicate with people using everyday language” (R. Mooney).

• Also called Computational Linguistics
  – Also concerns how computational methods can aid the understanding of human language

NLP is about COMMUNICATION
Course syllabus

• Introduction to NLP (1)
• Information Retrieval and Extraction (2)
• Question Answering (3)
• Speech recognition (4)
• Dialogue systems (5)
• Papers/projects (6)
• Use/experiment Sphynx CMU tool for speech recognition
Related Areas

- Artificial Intelligence
- Formal Language (Automata) Theory
- Machine Learning
- Linguistics
- Psycholinguistics
- Cognitive Science
- Philosophy of Language
Why NLP in your curriculum?

- Huge amounts of data
  - Internet = at least 20 billions pages
  - Intranet
- Applications for processing large amounts of texts require NLP expertise
- Classify text into categories
- Index and search large texts
- Automatic translation
- Speech understanding
  - Understand phone conversations
- Information extraction
  - Extract useful information from resumes
- Automatic summarization
  - Condense 1 book into 1 page
- Question answering
- Knowledge acquisition
- Text generations / dialogues
- The “latest”: micro-blog mining
Why NLP in your curriculum?

- Yahoo, Google, Microsoft → Information Retrieval
- Monster.com, HotJobs.com (Job finders) → Information Extraction + Information Retrieval
- Systran powers, Babelfish, Google Translate → Machine Translation
- Ask Jeeves, Wiki.answers → Question Answering
- Myspace, Facebook, Blogspot → Processing of User-Generated Content
- Alice, Eliza → Conversational agents
- Tools for “business intelligence”
- All “Big Guys” have (several) strong NLP research labs:
  - Google, IBM, Microsoft, AT&T, Xerox, Sun, etc.
- Academia: research in an university environment
NLP is difficult: Turing Test

- A test of a machine’s ability to demonstrate intelligence
- Introduced in 1950 by Alan Turing
- "I propose to consider the question, 'Can machines think?" Since "thinking" is difficult to define, Turing chooses to "replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. [...] Are there imaginable digital computers which would do well in the imitation game?"
  - Alan Turing, “Computing Machinery and Intelligence” (1950)
- Inspired by a party game, known as the “imitation game" (a man vs. a woman). It is a conversational task!!
Loebner Prize Gold Medal

- $100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's (= imitation game!)
- [http://www.loebner.net/Prizef/loebner-prize.html](http://www.loebner.net/Prizef/loebner-prize.html)
Loebner Prize Gold Medal

- A test of a machine’s ability to demonstrate intelligence
Loebner Prize Gold Medal

- A human judge engages in a (written) natural language conversation with one human and one machine
- The players try to appear human (not easy! Read “The most human human” by Brian Christian)
- All participants are separated from each another
- The judge tries to determine which player is a computer and which is a human
- Assumption: **NLP is AI-complete!**
- In other words, if we solve NLP, we are able to solve AI
Loebner Prize Gold Medal

- Winner is the program that manages to convince humans that he is a human.
- No one so far fully convinced all the judges, but the winner is the one that gets closer.
- In 2008 a program missed the target by one vote only!
- But they use (also) tricks..and psychology.
“Eccentric” answers

- By often changing the topic of conversation and saying funny things a computer can give the impression of a “human” personality
- Judge: Today I had hard time. I would like to relax a bit now. Do you have any joke to chase away the clouds??
- Computer: everyone talks about time, but no one does anything about. Why did you feel as saying ME that you would like to relax??
The first conversational “computer”

- Hal 9000
Science fiction..

….But in 2011…
• On February 2011 won the *Jeopardy!* contest against two humans

• Operating on a single CPU, it could take Watson two hours to answer a single question. A typical *Jeopardy!* contestant can accomplish this in less than three seconds.

• For Watson to rival the speed of its human competitors in delivering a single, precise answer to a question requires custom algorithms, terabytes of storage and thousands of POWER7 computing cores working in a massively parallel system.
Ibm Watson in Jeopardy!
Example

• “A recent best seller by Muriel Barbery is called ‘This of the Hedgehog,’ ”

• Watson: ”What" is "Elegance"?

• “You just need a nap. You don’t have this sleep disorder that can make sufferers nod off while standing up,”

• Watson “What" is "narcolepsy"?
Algorithms + Brute Force computing power

- The team fed numerous databases of knowledge into Watson's 16-terabyte memory, including multiple dictionaries, thesauri and even the entire text of Wikipedia, then armed it with the best available NLP algorithms to determine the correct response.
Brute force, but not so much..

• Watson's main innovation was not in the creation of a new algorithm but rather its ability to quickly execute thousands of language analysis algorithms simultaneously to find the correct answer.

• The more algorithms that find the same answer independently the more likely Watson is to be correct.

• Once Watson has a small number of potential solutions it is able to check against its database to ascertain if the solution makes sense.
So what about this course?

- To understand what the problems are
- To study (some) available solution
- Solution=algorithm
- So algorithms, after all!
..as any other field of computer science, NLP:

- Need to decompose the problem into sub-problems
- Find a reasonable solution for sub-problems
- Implement solution with an algorithm
- So, the standard problem solving methodology for ICT!
Sub-problems of NLP
The goal in the production and comprehension of natural language is communication.

Communication for the speaker:

- **Intention**: Decide when and what information should be transmitted (a.k.a. *strategic generation*). May require planning and reasoning about agents’ goals and beliefs.

- **Generation**: Translate the information to be communicated (in internal logical representation or “language of thought”) into string of words in desired natural language (a.k.a. *tactical generation*).

- **Synthesis**: Output the string in desired modality, text or speech.
NLP=Communication (cont)

• Communication for the hearer:
  – **Perception**: Map input modality to a string of words, e.g. *optical character recognition* (OCR) or *speech recognition*.
  – **Analysis**: Determine the **information content** of the string.
    • **Syntactic interpretation (parsing)**: Find the correct parse tree showing the phrase structure of the string.
    • **Semantic Interpretation**: Extract the (literal) meaning of the string (*logical form*).
    • **Pragmatic Interpretation**: Consider effect of the overall context on altering the literal meaning of a sentence.
  – **Incorporation**: Decide whether or not to believe the content of the string and add it to the KB.
Doesn’t always work....

What we say to dogs

Okay, Ginger! I've had it!
You stay out of the garbage!
Understand, Ginger? Stay out of the garbage, or else!

What they hear

blah blah GINGER blah
blah blah blah blah
blah blah GINGER blah
blah blah GIHNER blah
Another odd example
Computers are no better than your dog. But we can teach them “how-to” by coding our knowledge of the language comprehension process.
Aspects of NL processing

• **Analysis**
  
• From a natural language text to an unambiguous and formal (=computer processable) representation
Aspects of NL processing

- **Synthesis**
- Building programs able to generate a “correct” natural language text starting from some formal (=computer processable) content

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Cloud Sky Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td><strong>Percent (%)</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>09:00-12:00</td>
</tr>
<tr>
<td>Min</td>
<td>Mean</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind Speed</th>
<th>Cloud Sky Cover</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td><strong>Mode</strong></td>
</tr>
<tr>
<td>06:00-21:00</td>
<td>06:00-21:00</td>
</tr>
<tr>
<td>Min</td>
<td>Mean</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
</tr>
</tbody>
</table>

Cloudy, with a low around 10. South wind between 15 and 30 mph.
Architecture of a NLP system

INPUT

SPEECH

Phoneme recognition

Characer recognition

Morphological analysis

Part-of-speech tagging

LEXICAL ANALYSIS

Syntactic analysis

Semantic analysis

Pragmatics (discourse analysis)
Syntax, Semantic, Pragmatics

• Syntax concerns the proper ordering of words and its affect on meaning.
  – The dog bit the boy.
  – The boy bit the dog.
  – * Bit boy dog the the.
  – Colorless green ideas sleep furiously.
Syntax, Semantics, Pragmatics

- Semantics concerns the (literal) meaning of words, phrases, and sentences.
  - “plant” as a photosynthetic organism
  - “plant” as a manufacturing facility
  - “plant” as the act of sowing
Syntax, Semantic, Pragmatics

- Pragmatics concerns the overall communicative and social context and its effect on interpretation.

  - I went on holidays. It was my best time this year. (co-reference, ἀναφορά, "carrying back")

  - “the best course of action ... is doing nothing at all.”. (ἔλλειψις, ellipsis, "omission")
Ambiguity is the main issue in all NLP phases

• Natural language is highly ambiguous and must be disambiguated.
  – I saw the man on the hill with a telescope.
  – I saw the Grand Canyon flying to LA.
Ambiguity is Ubiquitous

• **Speech Recognition**
  - “recognize speech” vs. “wreck a nice beach”
  - “youth in Asia” vs. “euthanasia”

• **Morphology/POS**
  - Fly (noun, verb)

• **Syntactic Analysis**
  - “I ate spaghetti with chopsticks” vs. “I ate spaghetti with meatballs.”

• **Semantic Analysis**
  - “The lion is in the pen.” vs. “The ink is in the pen.”
  - “I put the plant in the window” vs. “Ford put the plant”

• **Pragmatic Analysis**
  - From “The Pink Panther Strikes Again”:
    - Clouseau: Does your dog bite?
    - Hotel Clerk: No.
    - Clouseau: [bowing down to pet the dog] Nice doggie.
    - [Dog barks and bites Clouseau in the hand]
    - Clouseau: I thought you said your dog did not bite!
    - Hotel Clerk: That is not my dog.
Ambiguity is Explosive

- Ambiguities generate an enormous numbers of possible interpretations.
- In English, a sentence ending in $n$ prepositional phrases has $\textit{over } 2^n$ syntactic interpretations (cf. Catalan numbers).
  - “Touch the man with the telescope”: 2 parses
  - “Touch the man on the hill with the telescope.”: 5 parses
  - “Touch the man on the hill in Texas with the telescope”: 14 parses
  - “Touch the man on the hill in Texas with the telescope at noon.”: 42 parses
  - “Touch the man on the hill in Texas with the telescope at noon on Monday” 132 parses
Why is Language Ambiguous?

- Having a unique linguistic expression for every possible conceptualization that could be conveyed would make language overly complex and linguistic expressions unnecessarily long.
- Allowing resolvable ambiguity permits shorter linguistic expressions, i.e. data compression.
- Language relies on people’s ability to use their knowledge and inference abilities to properly resolve ambiguities.
- Infrequently, disambiguation fails, i.e. the compression is lossy.
Time flies like an arrow
Natural Languages vs. Computer Languages

• Ambiguity is the primary difference between natural and computer languages.

• Formal programming languages are designed to be unambiguous, i.e. they can be defined by a grammar that produces a unique parse for each sentence in the language.

• Programming languages are also designed for efficient (deterministic) parsing, i.e. they are deterministic context-free languages (DCFLs).
  – A sentence in a DCFL can be parsed in $O(n)$ time where $n$ is the length of the string.
Natural Language Tasks

• Processing natural language text involves various syntactic, semantic and pragmatic tasks in addition to other problems.
Lexical and Syntactic Tasks
Word Segmentation

• The very first task is identifying the meaning units (words) = breaking a string of characters (graphemes) into a sequence of words.

• In some written languages (e.g. Chinese) words are not separated by spaces.

• Even in English, characters other than white-space can be used to separate words [e.g. , ; . - : ( ) ]

• Examples from English URLs:
  – jumptheshark.com ⇒ jump the shark .com
  – myspace.com/pluckerswingbar
    ⇒ myspace .com pluckers wing bar

• Examples from twitter hashtags
  – cold, congestion, low grade fevers, I hate #FeelingSick
    missing class today also, not good.
Morphological Analysis

- **Morphology** is the field of linguistics that studies the internal structure of words. (Wikipedia)
- **A morpheme** is the smallest linguistic unit that has semantic meaning (Wikipedia)
  - e.g. “carry”, “pre”, “ed”, “ly”, “s”
- Morphological analysis is the task of segmenting a word into its morphemes:
  - carried $\Rightarrow$ carry + ed (past tense)
  - independently $\Rightarrow$ in + (depend + ent) + ly
  - Googlers $\Rightarrow$ (Google + er) + s (plural)
  - unlockable $\Rightarrow$ un + (lock + able) ?
  - $\Rightarrow$ (un + lock) + able ?
Why is this necessary?

• Why do we need to know that “going” and “gone” are two forms of the same lemma “go”? 
Part Of Speech (POS) Tagging

• Annotate each word in a sentence with a part-of-speech.

I ate the spaghetti with meatballs.
Pro V Det N Prep N

John saw the saw and decided to take it to the table.
PN V Det N Con V Part V Pro Prep Det N

• Useful for subsequent syntactic parsing and word sense disambiguation.
Phrase Chunking

- Find all non-recursive **noun phrases** (NPs) and **verb phrases** (VPs) in a sentence.
  - [NP I] [VP ate] [NP the spaghetti] [PP with] [NP meatballs].
  - [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September]
Syntactic Parsing

- Produce the correct syntactic parse tree for a sentence.
Semantic Tasks
Word Sense Disambiguation (WSD)

• Words in natural language usually have a fair number of different possible meanings.
  – Ellen has a strong interest in computational linguistics.
  – Ellen pays a large amount of interest on her credit card.

• For most NLP tasks the proper sense of each ambiguous word in a sentence must be determined.
Semantic Role Labeling (SRL)

• For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.
  
  agent  patient  source  destination  instrument
  – John drove Mary from Austin to Dallas in his Toyota Prius.
  – The hammer broke the window.

• Also referred to as “case role analysis,” “thematic analysis,” and “shallow semantic parsing”
Labels vary according to domains

• “Pick up the pallet of boxes in the middle and place them on the trailer to the left”.
• Labels: EVENT, OBJECT; PLACE, PATH
• [Pick up the pallet of boxes in the middle]_{E1} and [place them on the trailer to the left]_{E2}.
• Pick up [the pallet of boxes]_{O} [in the middle]_{P} and place them [on the trailer to the left]_{P}.
Semantic Parsing

• A *semantic parser* maps a natural-language sentence to a complete, detailed semantic representation (*logical form*).

• For many applications, the desired output is immediately executable by another program.

• Example: Mapping an English database query to a logic expression:

   How many cities are there in the US?
   answer(A, count(B, (city(B), loc(B, C),
                       const(C, countryid(USA))),
                        A))
Textual Entailment

• Determine whether one natural language sentence entails (implies) another under an ordinary interpretation.
<table>
<thead>
<tr>
<th>TEXT</th>
<th>HYPOTHESIS</th>
<th>ENTAILMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeing the huge market potential, currently led by Google, Yahoo took over search company Overture Services Inc last year.</td>
<td>Yahoo bought Overture.</td>
<td>TRUE</td>
</tr>
<tr>
<td>Microsoft's rival Sun Microsystems Inc. bought Star Office last month and plans to boost its development as a Web-based device running over the Net on personal computers and Internet appliances.</td>
<td>Microsoft bought Star Office.</td>
<td>FALSE</td>
</tr>
<tr>
<td>The National Institute for Psychobiology in Israel was established in May 1971 as the Israel Center for Psychobiology by Prof. Joel.</td>
<td>Israel was established in May 1971.</td>
<td>FALSE</td>
</tr>
<tr>
<td>Since its formation in 1948, Israel fought many wars with neighboring Arab countries.</td>
<td>Israel was established in 1948.</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
Pragmatics/Discourse Tasks
Anaphora Resolution/Co-Reference

• Determine which phrases in a document refer to the same underlying entity.  
  – John put the carrot on the plate and ate it.
  
  – Bush started the war in Iraq. But the president needed the consent of Congress.

• Some cases require difficult reasoning.
  • Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a kite. "Don't do that," said Penny. "Jack has a kite. He will make you take it back."
Ellipsis Resolution

• Frequently words and phrases are omitted from sentences when they can be inferred from context.

"Wise men talk because they have something to say; fools, because they have to say something." (Plato)

"Wise men talk because they have something to say; fools talk because they have to say something." (Plato)
Putting all tasks together
Pipelining Problem

- Assuming separate independent components for speech recognition, syntax, semantics, pragmatics, etc. allows for more convenient modular software development.

- However, frequently constraints from “higher level” processes are needed to disambiguate “lower level” processes.
  - Example of syntactic disambiguation relying on semantic disambiguation:
    - At the zoo, several men were showing a group of students various types of flying animals. Suddenly, one of the students hit the man with a bat.
If a hard decision is made at each stage, cannot backtrack when a later stage indicates it is incorrect.

- If attach “with a bat” to the verb “hit” during syntactic analysis, then cannot reattach it to “man” after “bat” is disambiguated during later semantic or pragmatic processing.
Increasing Module Bandwidth

- If each component produces multiple scored interpretations, then later components can rerank these interpretations.

- **Problem:** Number of interpretations grows combinatorially.

- **Solution:** Efficiently encode combinations of interpretations.
  - Word lattices
  - Compact parse forests
Global Integration/ Joint Inference

• Integrated interpretation that combines phonetic/syntactic/semantic/pragmatic constraints.

- Integrated Interpretation
  - sound waves
  - meaning (contextualized)

• Difficult to design and implement.
• Potentially computationally complex.
So far we listed only problems..

• Forthcoming lessons will show *solutions* for specific applications (Information Extraction, Question Answering..)

• **Now**: An example of (not so “specific”) solution for the problem of POS tagging
An example: POS tagging algorithms

- Several algorithms for POS tagging have been defined in literature, based on algebraic, probabilistic or knowledge based methods.
- Why: a widely use algorithm for a variety of applications (cellular phones, human genoma, speech recognition and more).
Summary of HMM

• Hidden Markov Models are a stochastic model widely used in computer science, especially in telecommunications

• In NLP, HMM are used for:
  – Speech recognition
  – Part of Speech tagging
  – Syntactic analysis
Markov Models

- Set of states: \( \{s_1, s_2, \ldots, s_N\} \)
- Process moves from one state to another generating a sequence of states:
  \[ s_{i_1}, s_{i_2}, \ldots, s_{i_k}, \ldots \]
  (\( k_{th} \) state of sequence \( i \))
- Markov chain property: probability of each subsequent state depends only on the previous state:
  \[
  P(s_{ik} \mid s_{i_1}, s_{i_2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})
  \]
- To define a Markov model, the following probabilities have to be specified:
  - transition probabilities \( a_{ij} = P(s_i \mid s_j) \)
  - and initial probabilities \( \pi_i = P(s_i) \)
Example of Markov Model

- Two states: ‘Rain’ and ‘Dry’.
- Transition probabilities: $P(\text{‘Rain’}|\text{‘Rain’})=0.3$, $P(\text{‘Dry’}|\text{‘Rain’})=0.7$, $P(\text{‘Rain’}|\text{‘Dry’})=0.2$, $P(\text{‘Dry’}|\text{‘Dry’})=0.8$
- Initial probabilities: $P(\text{‘Rain’})=0.4$, $P(\text{‘Dry’})=0.6$. 
Calculation of sequence probability

• By Markov chain property, the probability of a state sequence can be found by the formula:

\[
P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1})P(s_{i1}, s_{i2}, \ldots, s_{ik-1})
\]

\[
= P(s_{ik} \mid s_{ik-1})P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots
\]

\[
= P(s_{ik} \mid s_{ik-1})P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1})P(s_{i1})
\]

• Suppose we want to calculate a probability of a sequence of states in our example, \{‘Dry’,’Dry’,’Rain’,Rain’\}.

\[
P(\{‘Dry’,’Dry’,’Rain’,Rain’\} ) = P(‘Rain’\mid’Rain’) P(‘Rain’\mid’Dry’) P(‘Dry’\mid’Dry’) P(‘Dry’)=
\]

\[
= 0.3*0.2*0.8*0.6
\]
Hidden Markov models.

- Set of states: \( \{s_1, s_2, \ldots, s_N\} \)
- Process moves from one state to another generating a sequence of states: \( s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots \)
- Markov chain property: probability of each subsequent state depends only on what was the previous state:
  \[
P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})
  \]
- States are not visible, but each state randomly generates one of \( M \) observations (or visible output)
  \( \{v_1, v_2, \ldots, v_M\} \)
- To define hidden Markov model, the following probabilities have to be specified: matrix of transition probabilities \( A=(a_{ij}) \), \( a_{ij}=P(s_i \mid s_j) \), matrix of observation probabilities \( B=(b_{mi})=P(v_m \mid s_i) \) and a vector of initial probabilities \( \pi=(\pi_i), \pi_i = P(s_i) \). Model is represented by \( M=(A, B, \pi) \).
Example of Hidden Markov Model

Weather conditions are VISIBLE, the states are HIDDEN (Low or High Pressure)
Example of Hidden Markov Model

• Two states: ‘Low’ and ‘High’ atmospheric pressure.

• Two observations: ‘Rain’ and ‘Dry’.

• Transition probabilities: \( P('Low'|'Low') = 0.3 \), \( P('High'|'Low') = 0.7 \), \( P('Low'|'High') = 0.2 \), \( P('High'|'High') = 0.8 \).

• Observation probabilities: \( P('Rain'|'Low') = 0.6 \) (“probability of seeing rain when the pressure is low”), \( P('Dry'|'Low') = 0.4 \), \( P('Rain'|'High') = 0.4 \), \( P('Dry'|'High') = 0.3 \).

• Initial probabilities: \( P('Low') = 0.4 \), \( P('High') = 0.6 \).
Calculation of observation sequence probability

• Suppose we want to calculate a probability of a sequence of observations in our example, {‘Dry’, ’Rain’}.

• Consider all possible hidden state sequences:


(a visible sequence can be generated by any of the possible hidden state sequences)

\[ P(A \& B) = P(A / B)P(B) \]

• Joint probabilities are calculated in the following way:

\[ P(\{’Dry’, ’Rain’\} & \{’Low’, ’Low’\}) = P(\{’Dry’, ’Rain’\} | \{’Low’, ’Low’\}) \cdot P(\{’Low’, ’Low’\}) = P(’Dry’|’Low’)P(’Rain’|’Low’)P(’Low’) = 0.4*0.4*0.6*0.4*0.3 \]

\[ P(seq) = \sum_i P(seq \land output_{seq_i}) = \sum_i P(seq / output_{seq_i})P(output_{seq_i}) \]
Main issues using HMMs:

**Evaluation problem.** Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1 o_2 ... o_K$, calculate the probability that model $M$ has generated sequence $O$.

**Decoding problem.** Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1 o_2 ... o_K$, calculate the most likely sequence of hidden states $s_i$ that produced this observation sequence $O$.

**Learning problem.** Given some training observation sequences $O=o_1 o_2 ... o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

$O=o_1...o_K$ denotes a sequence of observations $o_k \in \{v_1, ..., v_M\}$. 
POS tagging is an example of decoding problem

Outside pets are often hit by cars

The Markov model is not visible. But we assume to know that 1) each state generate a subset of all possible words; 2) from a given state, certain state transitions have zero probability (e.g. from PREP to PREP)
Which state sequence is the most likely?

- Which state sequence more likely generated “Outside pets are often hit by cars”?
  - Adj → Noun → Verb → Adv → Verb → Prep → Noun
  - Adv → Noun → Verb → Adv → Verb → Prep → Noun
  - Prep → Noun → Verb → Adv → Verb → Prep → Noun
  - Noun → Noun → Verb → Adv → Verb → Prep → Noun
  - Adj → Verb → Verb → Adv → Verb → Prep → Noun
  - 4x2x2x2 sequences = 64!!
  - Target: find an efficient algorithm to compute the most likely sequence
Trellies show the temporal evolution of a sequence

Outside pets are hit
Example 2

In this example all the $P(x_i, x_k)$ are non-zero

For observed sequences of length $k$ there are $|S|^k$ possible state sequences
We must estimate the max probability sequence of states

- Since this is a Markov process, for every $i$ we have:

$$P(s_1, \ldots, s_i, s_{i+1}, \ldots, s_k, y_1, \ldots, y_i, y_{i+1}, \ldots, y_k | s_0) = P(s_1, \ldots, s_i, y_1, \ldots, y_i | s_0)P(s_{i+1}, \ldots, s_k, y_{i+1}, \ldots, y_k | s_i)$$

$$\gamma(s_i) = \max_{s_{1..i-1}} P(s_1, \ldots, s_i, y_1, \ldots, y_i | s_0)$$

$$\max_{s_{1..k}} P(s_1, \ldots, s_i, s_{i+1}, \ldots, s_k, y_1, \ldots, y_i, y_{i+1}, \ldots, y_k | s_0) =$$

$$\max_s \left\{ \max_{s_{i+1..k}} P(s_i + 1, \ldots, s_k, y_{i+1}, \ldots, y_k | s)\gamma_i(s_i) \right\}$$

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_{i-1})$$

We can consider an internal state of the Markov chain and compute the sub-sequence that maximizes the probability of reaching this state.
\( \gamma_i(s) \)

- \( \gamma_i(s_i) \) is a function that determines the max-prob sequence of \((i-1)\) states that will bring to state \(s_i\) in step \(i\), given that \(s_0\) is the initial state, and given the observation of a sequence \(y_1..y_i\) of symbols.
Example

Let’s consider one of the possible sequences generating outside pets are hit:

\[
P(\text{adj, noun, ver, ver, outside, pets, are, hit} \mid s0) =
\]

\[
P(\text{adj, noun, ver, outside, pets, are} \mid s0)P(\text{ver, hit} \mid \text{ver})
\]

iterating:

\[
P(\text{adj, noun, ver, outside, pets, are} \mid s0) =
\]

\[
P(\text{adj, noun, outside, pet} \mid s0)P(\text{ver, are} \mid \text{noun})
\]

And finally:

\[
P(\text{adj, outside} \mid s0)P(\text{noun, pets} \mid \text{adj})P(\text{ver, are} \mid \text{noun})P(\text{ver, hit} \mid \text{ver})
\]

Probability of sequences are easily calculated, but what when there are millions of sequences?
Max_prob sequence

\[
(1) \quad \max_s \left\{ \max_{si+1..sk} P(si + 1, ..sk, yi + 1 ..yk \mid s) \gamma_i(s) \right\}
\]

Therefore:

1) For any level \( i \) of the trellis, and for any state \( s \) of \( i \), find the sequence that maximizes the probability of reaching \( s \):

\[ \gamma_i(s) \]

2) Then, find the most likely sequence that, from state \( s \) of level \( i \) of trellis brings to \( s_k \):

\[ \max_{si+1..sk} P(si + 1, ..sk, yi + 1 ..yk \mid s) \]

3) Finally, by considering all the \( s \) in \( i \), find the complete most likely sequence (formula (1))
Max_prob sequence

\[
\gamma(si) = \max_{s1..si-1} P(s1,..si, y_i,..y_i | s0)
\]

In a Markov chain we have

\[
p(s, y_i | s') = q(y_i | s, s') p(s | s')
\]

And therefore:

\[
\gamma_i(s_i) = \max_{s1...si-1} P(s1,s2..si, y1, y2..y_i | s0) = \max_{si-1} P(y_i, si / si-1) \max_{s1...si-2} P(s1,..si-1, y1...yi-1 | s0) = \max_{si-1} P(y_i, si / si-1) \gamma_{i-1}(s_{i-1}) = \max_{s} \gamma_k(s)
\]
And then..

$$\gamma_1(s) = \max_{s'} p(y_1, s | s') \gamma_0(s') = p(y_1, s | s0)$$

$$\gamma_2(s) = \max_{s'} p(y_2, s | s') \gamma_1(s')$$

$$\gamma_3(s) = \max_{s'} p(y_3, s | s') \gamma_2(s')$$

Etc etc
Viterbi algorithm

1. Set $\gamma_0(s_0) = 1$
2. Use previous formula (2) to compute the gamma function for the first column of the trellis, that is:
   $$\gamma_1(s) = \max_{s'} p(y_1, s | s') \gamma_0(s') = p(y_1, s | s_0)$$
   Note that $\gamma_0$ is zero for $s \neq s_0$!
3. Compute $\gamma_2$ for all $s$ of level 2 of trellis
   $$\gamma_2(s) = \max_{s'} p(y_2, s | s') \gamma_1(s')$$
   delete transitions $s' \to s$ for which
   $$p(y_2, s | s') \gamma_1(s') < \gamma_2(s)$$
4. Repeat for all states of column $i$ ($i=3,..k$) , and backwards, generate all possible sequences from $s$ that maximize
   $$\gamma_k(s)$$
Example

Outside/adj, noun, adv, ver pets/noun, ver are/verb hit/noun, ver lby/prep cars/noun

The problem is the the estimate of \( p(w_k, pos_i | pos_j) \)

\[
\begin{align*}
\gamma_1(\text{adj}) &= p(\text{outside}, \text{adj}|s0) \times 1 \\
\gamma_1(\text{adv}) &= p(\text{outside}, \text{adv}|s0) \times 1 \\
\gamma_1(\text{noun}) &= p(\text{outside}, \text{noun}|s0) \times 1 \\
\gamma_1(\text{ver}) &= p(\text{outside}, \text{ver}|s0) \times 1 \\
\gamma_1(s, s\neq \text{adj}, \text{noun}, \text{adv}, \text{ver}) &= 0
\end{align*}
\]

For now we suppose that the Markov model \( M=(A, B, \pi) \) is known.
i=2

$$p(\text{pets} \mid \text{adj})g_{1}(\text{adj}) = 0.4 \times 0.4 = 0.16$$

$$p(\text{pets} \mid \text{adj})g_{1}(\text{adj}) = 0.2 \times 0.4 = 0.08$$

$$p(\text{pets} \mid \text{noun})g_{1}(\text{noun}) = 0.2 \times 0.3 = 0.06$$

$$p(\text{pets} \mid \text{noun})g_{1}(\text{noun}) = 0.5 \times 0.3 = 0.15$$

$$p(\text{pets} \mid \text{adv})g_{1}(\text{adv}) = 0.1 \times 0.2 = 0.02$$

$$p(\text{pets} \mid \text{adv})g_{1}(\text{adv}) = 0.2 \times 0.2 = 0.04$$

$$p(\text{pets} \mid \text{ver})g_{1}(\text{ver}) = 0.3 \times 0.1 = 0.03$$

$$p(\text{pets} \mid \text{ver})g_{1}(\text{ver}) = 0.1 \times 0.1 = 0.01$$

$$\gamma_{2}(\text{noun}) = 0.16$$

Less likely sequences are eliminated
i=3

\[ p(\text{are}, \text{verb} | \text{noun}) \gamma_2(\text{ag}) = 0.5 \times 0.18 = 0.82 \]

\[ \gamma_3(\text{verb}) = 0.82 \]
..finally

In the last step “ver” is chosen since it is the most probable

Therefore, the “hidden” most likely string is ADJ NOUN VERB VERB
HMM+Viterbi is also used for speech recognition (later in this course)

- Observed input signal (voice input)
- Spectral vectors
- Estimate of phoneme sequences
- HMM+Viterbi
- Word sequences
Parameter estimation $M=(A, B, \pi)$

- The Viterbi algorithm is based on an estimate of probabilities $p(y_k, s|s')$ where $y_k$ is the observed output and $s, s'$ are the model states (words, parts of speech, etc.).
  - Model parameters can be estimated on a training set, if available.
  - For POS tagging, corpora manually tagged with the appropriate POS tags have been prepared, e.g. *Wall Street Journal corpus*, for speech understanding, several decoded speech corpora are also available, like PRONELEX, CMUdict.)
  - A well-known algorithm for estimating parameters in a HMM is the **Baum-Welch algorithm** [http://labrosa.ee.columbia.edu/doc/HTKBook21/node7.html](http://labrosa.ee.columbia.edu/doc/HTKBook21/node7.html)
Several hundreds sentences Annotated with POS tags allows it To estimate \( p(w_i, \text{POS}_i | \text{POS}_{i-1}) \) \( E \ p(\text{POS}_i, S_i | \text{POS}_{i-1}, S_{i-1}) \)
This brings us to another BIG problem in NLP (and AI in general)

The knowledge bottleneck
Manual Knowledge Acquisition

• Traditional, “rationalist,” approaches to language processing require human specialists to specify and formalize the required knowledge.
• Manual knowledge engineering, is difficult, time-consuming, and error prone.
• “Rules” in language have numerous exceptions and irregularities.
  – “All grammars leak.”: Edward Sapir (1921)
• Manually developed systems were expensive to develop and their abilities were limited and “brittle” (not robust).
Automatic Learning Approach

• Use machine learning methods to automatically acquire the required knowledge from appropriately annotated text corpora.

• Variously referred to as the “corpus based,” “statistical,” or “empirical” approach.

• Statistical learning methods were first applied to speech recognition in the late 1970’s and became the dominant approach in the 1980’s.

• During the 1990’s, the statistical training approach expanded and came to dominate almost all areas of NLP.
Learning Approach

Manually Annotated Training Corpora

Machine Learning

Linguistic Knowledge

NLP System

Raw Text

Automatically Annotated Text
Advantages of the Learning Approach

• Large amounts of electronic text are now available.

• Annotating corpora is easier and requires less expertise than manual knowledge engineering.

• Learning algorithms have progressed to be able to handle large amounts of data and produce accurate probabilistic knowledge.

• The probabilistic knowledge acquired allows robust processing that handles linguistic regularities as well as exceptions.
Next lesson

Information Extraction
To conclude

A brief history of NLP
Early History: 1950’s

• Shannon (the father of information theory) explored probabilistic models of natural language (1951).

• Chomsky (the extremely influential linguist) developed formal models of syntax, i.e. finite state and context-free grammars (1956).

• First computational parser developed at U Penn as a cascade of finite-state transducers (Joshi, 1961; Harris, 1962).

• Bayesian methods developed for *optical character recognition* (OCR) (Bledsoe & Browning, 1959).
History: 1960’s

- Work at MIT AI lab on question answering (BASEBALL) and dialog (ELIZA).
- Semantic network models of language for question answering (Simmons, 1965).
- First electronic corpus collected, Brown corpus, 1 million words (Kucera and Francis, 1967).
- Bayesian methods used to identify document authorship (The Federalist papers) (Mosteller & Wallace, 1964).
“Natural language understanding” systems developed that tried to support deeper semantic interpretation.

- SHRDLU (Winograd, 1972) performs tasks in the “blocks world” based on NL instruction.
- Schank et al. (1972, 1977) developed systems for conceptual representation of language and for understanding short stories using hand-coded knowledge of scripts, plans, and goals.

Prolog programming language developed to support logic-based parsing (Colmeraufer, 1975).

Initial development of hidden Markov models (HMMs) for statistical speech recognition (Baker, 1975; Jelinek, 1976).
History: 1980’s

- Development of more complex (mildly context sensitive) grammatical formalisms, e.g. unification grammar, HPSG, tree-adjoining grammar.
- Symbolic work on discourse processing and NL generation.
- Initial use of statistical (HMM) methods for syntactic analysis (POS tagging) (Church, 1988).
History: 1990’s

- Rise of statistical methods and empirical evaluation causes a “scientific revolution” in the field.
- Initial annotated corpora developed for training and testing systems for POS tagging, parsing, WSD, information extraction, MT, etc.
- First statistical machine translation systems developed at IBM for Canadian Hansards corpus (Brown et al., 1990).
- First robust statistical parsers developed (Magerman, 1995; Collins, 1996; Charniak, 1997).
- First systems for robust information extraction developed (e.g. MUC competitions).
History: 2000’s

- Increased use of a variety of ML methods, SVMs, logistic regression (i.e. max-ent), CRF’s, etc.
- Continued development of corpora and competitions on shared data.
  - TREC Q/A
  - SENSEVAL/SEMEVAL
  - CONLL Shared Tasks (NER, SRL…)
- Increased emphasis on unsupervised, semi-supervised, and active learning as alternatives to purely supervised learning.
- Shifting focus to semantic tasks such as WSD and SRL.
Relevant Scientific Conferences

• Association for Computational Linguistics (ACL)
• North American Association for Computational Linguistics (NAACL)
• International Conference on Computational Linguistics (COLING)
• Empirical Methods in Natural Language Processing (EMNLP)
• Conference on Computational Natural Language Learning (CoNLL)
• International Association for Machine Translation (IMTA)
Homework for next lesson

• Viterbi algorithm and HMM
• 5 VERY simple questions on today’s presentation + 2 questions on Viterbi algorithm (download tutorial from course web site)