Introduction to Information Sciences

Introduction to the Course

M. Cuturi, X. Liang
Summary of the Course

- Provide a wide overview of
  - natural language processing & linguistics
  - information theory
  - computer science
  - artificial intelligence
  - pattern recognition, speech processing *etc.*

- This course is designed to introduce the activities of the *Intelligence Science and Technology* department to the rest of the university.
Grading

- Two lecturers → overall grade split into two respective parts
- each part is equally weighted.

- Attendance is important.
  - Total of 13 classes.
  - You should attend at least \( \geq 10 \) classes. you have 3 jokers.
  - Below that, -10% per missed class.
  - Below \( \leq 7 \), no credit.

- Grading will be based on reports. 2 reports in my course.
Before we proceed...

Questions?···
Let’s start with a little Quizz

Question 1

Consider the following algorithm

- \( x \leftarrow 0; \)
- \textbf{For} \( i = 1, \cdots, 3 \)
  - \( x \leftarrow x + i; \)
- \textbf{endFor}

What is the value of \( x \) once this algorithm has been executed?

1. 3  
2. 0  
3. 6  
4. 1  
5. 12
Let’s start with a little Quizz

Question 1

Consider the following algorithm

- $x \leftarrow 0$
- **For** $i = 1, \ldots, 3$
  - $x \leftarrow x + i$
- **endFor

What is the value of $x$ once this algorithm has been executed?

1. 3  
2. 0  
3. 6  
4. 1  
5. 12
Let’s start with a little Quizz

Question 2

Consider the following algorithm

- $x \leftarrow 2$
- For $i = 1, \cdots, 3$
  - If $i \leq x$
    - $x \leftarrow x - i$
  - Else
    - $x \leftarrow x + i$
- endIf
- endFor

What is the value of $x$ once this algorithm has been executed?

1. 2  2. 0  3. 4  4. -1  5. 3
Let's start with a little Quizz

Question 2

Consider the following algorithm

- $x \leftarrow 2$;
- **For** $i = 1, \ldots, 3$
  - **If** $i \leq x$,
    - $x \leftarrow x - i$;
  - **Else**
    - $x \leftarrow x + i$;
- **endIf**
- **endFor**

What is the value of $x$ once this algorithm has been executed?

1. 2  2. 0  3. 4  4. -1  5. 3
Introduction to Information Sciences

Natural Language Processing
Formal Languages

mcuturi@i.kyoto-u.ac.jp
Summary

- Illustrate the difficulties tackled by computational linguistics
  - Define a few of the problems commonly studied
- Introduce formal language theory & Automata
  - formal languages
  - formal grammars
  - Chomsky hierarchy

Sources for these slides: A. McCallum’s (UMass) online lectures, Wikipedia, Jurafsky/Martin
We start with an example: HAL

- An example taken from a famous movie and book:

![2001: A Space Odyssey](image)

- Let’s check a few scenes:
2001 was shot in 1968

A few years after 2001, what sounds familiar, if not outdated about HAL?

- Graphic capabilities?.. We have much better. The future rather looks like this...

- Chess? 2006, Deep Fritz and before, late 90’s, Deep Blue
What still sounds difficult to achieve is HAL’s *articulated syntax*...

David Bowman:

*Open the pod bay doors, Hal.*

HAL:

*I’m sorry, Dave, I’m afraid I can’t do that.*

David Bowman:

*What are you talking about, Hal?*

...HAL:

*I know that you and Frank were planning to disconnect me, and I’m afraid that’s something I cannot allow to happen.*

The machine is also displaying *intelligence*. See Turing’s test.

Yet, *why does language seem more difficult to reach than chess?*
THIS WINE. WOOT MEMBER WILL SOON BE ABLE TO AFFORD MORE EXPENSIVE WINE

http://www.says-it.com/jeopardy/
Layers of Computational Linguistics

Complex and multilayered system, each layer a different study field

- Phonetics
- Phonology
- Morphology
- Syntax
- Semantics
- Pragmatics
- Discourse
**Phonetics**

Study of language sounds, physical aspects.

### THE INTERNATIONAL PHONETIC ALPHABET (revised to 1993)

<table>
<thead>
<tr>
<th>Consonants (Pulmonic)</th>
<th>Bilabial</th>
<th>Labiodental</th>
<th>Dental</th>
<th>Alveolar</th>
<th>Postalveolar</th>
<th>Palatoalveolar</th>
<th>Palatal</th>
<th>Velar</th>
<th>Uvular</th>
<th>Velopharyngeal</th>
<th>Glottal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fricative</td>
<td>p b</td>
<td>t d</td>
<td>t d</td>
<td>c j</td>
<td>k g q g</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasal</td>
<td>m m</td>
<td>n n</td>
<td>n n j</td>
<td>n j n j</td>
<td>n n n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trill</td>
<td>B R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tap or flap</td>
<td>r t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fricative</td>
<td>ð f v</td>
<td>Ø ð s z j 3</td>
<td>ð s z</td>
<td>ç i x y x</td>
<td>h n h f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lateral Fricative</td>
<td>l b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximant</td>
<td>o j a j</td>
<td>i u j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lateral Approximant</td>
<td>l l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where symbols appear in pairs, the one to the right represents a voiceless consonant. Shaded areas denote articulations judged impossible.

### Consonants (Non-Pulmonic)

**VOYCELS**

<table>
<thead>
<tr>
<th>Front</th>
<th>Central</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>i y</td>
<td>ø u o</td>
<td>e r e</td>
</tr>
<tr>
<td>Close</td>
<td></td>
<td>Open</td>
</tr>
<tr>
<td>Close-mid</td>
<td>e ø e</td>
<td>e ø e</td>
</tr>
<tr>
<td>Open-mid</td>
<td>e ø e</td>
<td>e ø e</td>
</tr>
<tr>
<td>Open</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Stress**

- Primary stress
- Secondary stress

**Tones & Word Accents**

<table>
<thead>
<tr>
<th>Level</th>
<th>Contour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra High</td>
<td>Rising</td>
</tr>
<tr>
<td>High</td>
<td>Falling</td>
</tr>
<tr>
<td>Mid</td>
<td>High Rising</td>
</tr>
<tr>
<td>Low</td>
<td>Low Rising</td>
</tr>
<tr>
<td>Extra Low</td>
<td>Rising-falling</td>
</tr>
</tbody>
</table>

**Diacritics**

- Voiced
- Voiceless
- Labialized
- Palatized
- Rhotacized
- Voiced labio-dental fricative
- Voiced labialized fricative
- Voiced palato-alveolar fricative
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental fricative
- Voiced palatal affricate
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal fricative
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labialized fricative
- Voiced labio-dental approximant
- Voiced palatal approximant
- Voiced labialized approximant
- Voiced labIALIZED fricative
- Voiced labializedapproximant
Phonology

Study of sound **structure** of language.

- Identify units of sounds, in different human languages.
  - phonemes,
  - syllables,
- Phonemes are a major difference between animal language and human language.
- Useful for instance in animations. Phonemes in English:
Morphology

Study of morphemes, the minimal units of linguistic form and meaning

**disconnect**

“not”  “to attach”

- Important for compounded languages *e.g.* Turkish:

  **uygarlastiramadiklarimizdanmissinizcasina**

  uygur las tir ama dik lar imiz dan mis siniz casina

  (behaving) as if you are among those whom we could not civilize

- In Chinese, Chinese characters = morphems = basic semantic unit
Syntax

• From words to sentences:

I know that you and Frank were planning to disconnect me.

• Of course, the structure (the syntax) of the following sentence is also correct

You know me—Frank and I were planning to disconnect that.
Semantics

Study of meaning, the minimal units of linguistic form and meaning

- The meaning of

  I know that you and Frank were planning to disconnect me.

  can be summarized as

  - an *action*, **disconnect**,  
  - performed by an *actor*, **you and Franck**,  
  - on an *object*, **me**

- In computer science, different syntaxes for the same operation:
  
  - `x += y` (C, Java, Perl, Python, Ruby, etc.)
  - `x := x + y` (Pascal)
  - `LET X = X + Y` (early BASIC)
  - `x = x + y` (MATLAB, most BASIC dialects, Fortran)
  - `(incf x y)` (Common Lisp)
Pragmatics

The study of how language is used to accomplish goals within a given context

• What is the practical outcome of a sentence as

   *Im sorry, Dave, Im afraid I cant do that.*

   given the context?

• The sentence ”You have a green light” can have different meanings:
  
  ○ It could mean you are holding a green light bulb.
  ○ Or that you have a green light to drive your car.
  ○ Or it could be indicating that you can go ahead with the project.
  ○ Or that your body has a green glow
Discourse

Study of linguistic units which are larger than single utterances

- Capture the different turns, threads, changes in the conversation

David Bowman:

Open the pod bay doors, Hal.

HAL:

I'm sorry, Dave, I'm afraid I can't do that.

David Bowman:

What are you talking about, Hal?

...HAL:

I know that you and Frank were planning to disconnect me, and I'm afraid that's something I cannot allow to happen.
Languages contain all possible utterances

- Here are sentences in the english language,
  - The man took the book.
  - This sentence is not true.
  - The horse was galloping in the prairie

- Here are sentences which are not part of it
  - The true the eat lot looks bird.
  - sense any make not does sentence this
  - dfdlkfh lKjer lEREQ ARlkjdf

- A few different kinds of language:
  - Natural languages language that arises in an unpremeditated manner as the product of the human innate facility to communicate. Can be spoken, signed, written etc..
  - Constructed languages constructed languages as auxiliary languages such as esperanto or artistic languages (e.g. in fiction)
  - Formal languages: languages that computers can parse and understand.

- The latter is the family of languages we will study the most in these 2 lectures.
Seen from a computer, a language is a set

- We start with the formal idea of alphabets, a set of tokens

\[ \Sigma = \{a, b, c, d, e, f, g, \cdots, z, \cdots\} \text{ or,} \]
\[ \Sigma = \{0, 1\} \text{ or,} \]
\[ \Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, \ln, \exp, \cdots\} \].

- and use the Kleene operator as a shortcut for

\[ \Sigma^* = \{x \in \Sigma^n, n \in \mathbb{N}\}. \]

- A formal language \( L \) is a **subset** of \( \Sigma^* \).
Example of a language

Rules can describe a formal language L

- Consider the language L defined as
  - The alphabet = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, =:
  - Every nonempty string that does not contain + or = and does not start with 0 is in L.
  - The string 0 is in L.
  - A string containing = is in L if and only if there is exactly one =, and it separates two valid strings in L.
  - A string containing + but not = is in L if and only if every + in the string separates two valid strings in L.
  - No string is in L other than those implied by the previous rules.

- With such rules,
  - "23+4=55" is in L,
  - "d433+2e2" is not,
  - "=234=+" is not.

- no meaning yet though. Only notion of belonging or not to a language.
Formal languages = typology of such rules

- Other ways to define a language from an alphabet:
  - For instance, a language can be given as
    - all strings generated by a **formal grammar**;
    - all strings accepted by some **automaton**, in the example the automaton can generate the language of all words containing at least "aba" once
    - all strings described or matched by a particular **regular expression**;
    - all strings for which some decision procedure (an algorithm that asks a sequence of related YES/NO questions) produces the answer YES.
Typical questions asked about such formalisms

- What is their expressive power? (Can formalism X describe every language that formalism Y can describe? Can it describe other languages?)

- What is their recognizability? (How difficult is it to decide whether a given word belongs to a language described by formalism X?)

- What is their comparability? (How difficult is it to decide whether two languages, one described in formalism X and one in formalism Y, or in X again, are actually the same language?).
A **formal grammar** is a set of rules which generate **formal languages**, defined by:

- a finite set of terminal symbols,
- a finite set of nonterminal symbols,
- a start symbol which is a nonterminal symbol,
- a finite set of production rules:

  \[ \text{Rule} : \cdots \rightarrow \cdots \]

  where the dots are arbitrary symbols.
Formal grammar

- How?
  - Start with the start symbol.
  - Apply any rule by replacing an occurrence of the symbols on the left-hand side of the rule with those that appear on the right-hand side.

- A sequence of rule applications is called a derivation.

Such a grammar defines the formal language: all words consisting solely of terminal symbols which can be reached by a derivation from the start symbol.

- Usually, NONTERMINALS are represented by uppercase letters,
- terminals by lowercase letters,
- the start symbol by S.
Formal grammar Example

• For example, the grammar with

  ○ terminals \( \{a, b\} \),
  ○ nonterminals \( \{S, A, B\} \), starting \( S \),
  ○ production rules
    ▶ \( S \rightarrow ABS \)
    ▶ \( S \rightarrow \varepsilon \) (where \( \varepsilon \) is the empty string)
    ▶ \( BA \rightarrow AB \)
    ▶ \( BS \rightarrow b \)
    ▶ \( Bb \rightarrow bb \)
    ▶ \( Ab \rightarrow ab \)
    ▶ \( Aa \rightarrow aa \)

defines the language of all words of the form \( a^n b^n \).

• simpler grammar that defines the same language:

  ○ Terminals \( \{a, b\} \),
  ○ Nonterminals \( \{S\} \), Start symbol \( S \), Production rules
    ▶ \( S \rightarrow aSb \)
    ▶ \( S\varepsilon \)
Chomsky Hierarchy of Formal Languages

- **Type-0**: all grammars.
- **Type-1**: \(\alpha A\beta \rightarrow \alpha\gamma\beta\) where \(\gamma\) cannot be empty. \(S \rightarrow \varepsilon\) is allowed iff \(S\) does not appear on the right side of a rule.
- **Type-2**: \(A \rightarrow \gamma\) where \(\gamma\) a string of terminals and nonterminals.
- **Type-3**: Nonterminals can only appear on one side, \(S \rightarrow \varepsilon\) is allowed iff \(S\) does not appear on the right side of a rule.

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Languages</th>
<th>Automaton</th>
<th>Production rules (constraints)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-0</td>
<td>Recursively enumerable</td>
<td>Turing machine</td>
<td>(\alpha \rightarrow \beta) (no restrictions)</td>
</tr>
<tr>
<td>Type-1</td>
<td>Context-sensitive</td>
<td>Linear-bounded non-deterministic Turing machine</td>
<td>(\alpha A\beta \rightarrow \alpha\gamma\beta)</td>
</tr>
<tr>
<td>Type-2</td>
<td><strong>Context-free</strong></td>
<td>Non-deterministic pushdown automaton</td>
<td>(A \rightarrow \gamma)</td>
</tr>
<tr>
<td>Type-3</td>
<td>Regular</td>
<td>Finite state automaton</td>
<td>(A \rightarrow a) and (A \rightarrow aB)</td>
</tr>
</tbody>
</table>

- Most programming languages are generated by Type-2 rules.
- Trade-off between size of language & capacity to parse it.