# Introduction to Computational Linguistics

Pavlina Ivanova
University of Plovdiv, Bulgaria

## Lecture 1: Overview of the Field, Quick Review of Regular Expressions

# Linguistics

**Linguistics** – the study of language (in general and of particular languages)

**Language** – one of the fundamental aspects of human behavior

It serves as:

- knowledge representation (in written form) - a long-term record of knowledge from one generation to the next
- communication between people (in spoken form) - our primary means of coordinating our day-to-day behavior with others

Language is studied in several different academic disciplines. Each discipline defines its own set of problems and has its own methods for addressing them.

- **Linguistics** - studies the structure of language itself, considering questions such as why certain combinations of words form sentences but others do not, and why a sentence can have some meanings but not others.
- **Psycholinguistics** - studies the processes of human language production and comprehension, considering questions such as how people identify the appropriate structure of a sentence and when they decide on the appropriate meaning for words.
- **Philosophy** - considers how words can mean anything at all and how they identify objects in the world. Philosophers also consider what it means to have beliefs, goals, and intentions, and how these cognitive capabilities relate to language.

# Definition

**Computational linguistics (CL)** is an interdisciplinary field concerned with the **processing** of a natural language (NLP) by computers. It deals with the **modeling** of NL from a computational perspective.

The goal of the CL is to develop a computational theory of language, using the notions of algorithms and data structures from computer science.

Of course, to build a computational model, you must take advantage of what is known from all the other disciplines.

# Different approaches to studying language

| Discipline | Typical Problems | Tools |
|---|---|---|
| Linguistics | How do words form phrases and sentences? What constrains the possible meanings for a sentence? | Intuitions about well-formedness and meaning; mathematical models of structure (for example, formal language theory, model theoretic semantics) |
| Psycholinguistics | How do people identify the structure of sentences? How are word meanings identified? When does understanding take place? | Experimental techniques based on measuring human performance; statistical analysis of observations |
| Philosophy | What is meaning, and how do words and sentences acquire it? How do words identify objects in the world? | Natural language argumentation using intuition about counter-examples; mathematical models (for example, logic and model theory) |
| Computational Linguistics | How is the structure of sentences identified? How can knowledge and reasoning be modeled? How can language be used to accomplish specific tasks? | Algorithms, data structures; formal models of representation and reasoning; AI techniques (search and representation methods) |

# Goals

- to create computational models of language in enough detail that you could write computer programs to perform various tasks involving natural language; to develop methods and algorithms for (automatic) NLP usable in the real systems
- (the ultimate goal) to be able to specify models that approach human performance in the linguistic tasks of reading, writing, hearing, and speaking

# Computational models are useful for:

- scientific purposes - for exploring the nature of linguistic communication (exploring the language phenomena)
- practical purposes - for enabling effective human-machine communication (to build the effective computational programs for modeling different aspects of the human language)

# Motivations for developing computational models

- (scientific) to obtain a better understanding of how language works (how language comprehension and production work)

  We may be able to realize complex theories as computer programs and then test them by observing how well they perform. By seeing where they fail, we can incrementally improve them.

  Computational models may provide very specific predictions about human behavior that can then be explored by the psycholinguists. By continuing in this process, we may eventually acquire a deep understanding of how human language processing occurs.

- (practical, or technological) natural language processing capabilities would revolutionize the way computers are used

  Since most of human knowledge is recorded in linguistic form, computers that could understand natural language could access all this information.

  Natural language interfaces to computers would allow complex systems to be accessible to everyone.

  For technological purposes it does not matter if the model used reflects the way humans process language. It only matters that it works.

# Subareas of NLP (CL)

- speech (spoken language) processing
- text (written language) processing

The main goal of NLP is to make machines "understand" NL. "To understand" means to recognize and use information expressed in NL (i.e. to model human ability to understand NL by computer program). It concerns the acquisition of internal representation of the world that corresponds to the text.

# Applications of NLP

- text-based applications - involve the processing of written text, such as books, newspapers, reports, manuals, e-mail messages, and so on. These are all reading-based tasks.
- dialogue-based applications - involve human-machine communication. Most naturally this involves spoken language, but it also includes interaction using keyboards.

# Text-based natural language applications

- **information retrieval** – finding appropriate documents on certain topics from a database of texts (for example, finding relevant books in a library)
- **information extraction** – extracting information from messages or articles on certain topics (for example, building a database of all stock transactions described in the news on a given day)
- **machine translation** – translating documents from one language to another (for example, producing automobile repair manuals in many different languages)
- **text summarizing** – summarizing texts for certain purposes (for example, producing a 3-page summary of a 1000-page government report)

# Dialogue-based applications

- **question-answering** systems, where natural language is used to query a database (for example, a query system to a personnel database)
- **automated customer service** over the telephone (for example, to perform banking transactions or order items from a catalogue)
- **tutoring systems**, where the machine interacts with a student (for example, an automated mathematics tutoring system)
- **spoken language control of a machine** (for example, voice control of a computer)
- **general cooperative problem-solving systems** (for example, a system that helps a person plan and schedule freight shipments)

Not all systems that perform such tasks must be using natural language understanding techniques. They use simply a (string or pattern) matching technique – what the computers do well.

A crucial characteristic of an understanding system: it must compute some representation of the information that can be used for later inference.

# Knowledge needed to build NL-system

A natural language-system must use considerable knowledge about the structure of the language itself, including what the words are, how words combine to form sentences, what the words mean, how word meanings contribute to sentence meanings, and so on.

# Kinds of knowledge needed

- **Phonetic and phonological knowledge -** concerns how words are related to the sounds that realize them. Such knowledge is crucial for speech-based systems.
- **Morphological knowledge** - concerns how words are constructed from more basic meaning units called morphemes.
- **Syntactic knowledge** - concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of what other phrases.
- **Semantic knowledge** - concerns what words mean and how these meanings combine in sentences to form sentence meanings. This is the study of context-independent meaning - the meaning a sentence has regardless of the context in which it is used.

# Kinds of knowledge needed

- **Pragmatic knowledge** - concerns how sentences are used in different situations and how use affects the interpretation of the sentence.
- **Discourse knowledge** - concerns how the immediately preceding sentences affect the interpretation of the next sentence. This information is especially important for interpreting pronouns and for interpreting the temporal aspects of the information conveyed.
- **World knowledge** - includes the general knowledge about the structure of the world that language users must have in order to, for example, maintain a conversation. It includes what each language user must know about the other user's beliefs and goals.

# Example (Syntax, Semantics, and Pragmatics)

Consider each of the following sentences as a candidate for the initial sentence of a book about CL:

2. Language is one of the fundamental aspects of human behavior and is a crucial component of our lives.
3. Green frogs have large noses.
4. Green ideas have large noses.
5. Large have green ideas noses.

# Ambiguity

- Computational linguists are obsessed with ambiguity
- Ambiguity is a fundamental problem of computational linguistics
- Resolving ambiguity is a crucial goal

# Ambiguity

- lexical – a single word can have more than one meaning

  ball (noun) – a spherical object or a dancing event

  round – can be a noun, or a verb, or an adjective, or an adverb, or a preposition

- structural

  The man saw the girl with a telescope.

  Visiting relatives can be boring.

# Different levels of ambiguity

- Structural ambiguities can be further classified according to the "range" of the ambiguity they represent
  - local ambiguities – lies not in the words themselves but in their juxtaposition

    Did you read the story about the aircrash in the jungle?

    John mentioned the book I sent to Susan.
  - global ambiguities – caused by combinations of category ambiguities

    He noticed her shaking hands.

# Different levels of ambiguity

- Subcategorization of ambiguity according the "depth" at which it the arises

  John is eager to please.

  John is easy to please.

  The rabbit is ready to eat.

- The problem of anaphora resolution (identification of antecedents of pronouns)

  The monkey ate the banana because it was hungry.

  The monkey ate the banana because it was ripe.

  The monkey ate the banana because it was tea-time.

# Other features of the language

- Implicit meaning

  I have two children.

  Have you got a watch?

- Concept and denoter

  The president gave a party.

  Venus, evening star, vesper, morning star, Hesperus

# Evaluating NLP Systems

- run the program and see how well it performs the task it was designed to do (black box evaluation)

  It is problematic in the early stages of research because early evaluation results can be misleading. Sometimes the techniques that produce the best results in the short term will not lead to the best results in the long term.

- identify various subcomponents of a system and then evaluate each one with appropriate tests (glass box evaluation)

  The problem with glass box evaluation is that it requires some consensus on what the various components of a natural language system should be. Achieving such a consensus is an area of considerable activity at the present.

# Turing Test

- Question: „can machine think“ is not operational
- Operational version:
  - 2 people and a computer
  - Interrogator talks to contestant and computer via teletype
  - Task of machine is to convince interrogator it is human
  - Task of contestant is to convince interrogator he and not machine is human.

# Eliza: Weizenbaum (1966)

- The system plays the role of a **therapist** and, to obtain the best results, the user should correspondingly play the role of a **patient**.
- It serves as an excellent example as its behavior initially seems impressive.

# A dialogue with ELIZA

| USER | ELIZA |
|---|---|
| Men are all alike. | In what way? |
| They're always bugging us about something or other. | Can you think of a specific example? |
| Well, my boyfriend made me come here. | Your boyfriend made you come here. |
| He says I'm depressed much of the time. | I am sorry to hear you are depressed. |
| It's true, I am unhappy. | Do you think coming here will help you not to be unhappy? |
| I need some help; that much seems certain. | What would it mean to you if you got some help? |
| Perhaps I could learn to get along with my mother. | Tell me more about your family. |
| My mother takes care of me. | Who else in your family takes care of you? |
| My father. | Your father. |
| You are like my father in some ways. | What resemblance do you see? |

# Simple description of how ELIZA works

- database of particular words (keywords)
- for each keyword the system stores:
  - an integer
  - a pattern to match against the input
  - a specification of the output

The algorithm is as follows:

Given a sentence S. Find a keyword in S whose pattern matches S. If there is more than one keyword, pick the one with the highest integer value. Use the output specification that is associated with this keyword to generate the next sentence. If there are no keywords, generate an innocuous continuation statement, such as "*Tell me more"* or "*Go on"*.

# Sample data from ELIZA

| Word | Rank | Pattern | Outputs |
|---|---|---|---|
| alike | 10 | ?X | In what way?<br>What resemblance do you see? |
| are | 3 | ?X are you ?<br>Y | Would you prefer it if I weren't ?<br>Y? |
| | 3 | ?X are ?Y | What if they were not ?Y? |
| always | 5 | ?X | Can you think of a specific example?<br>When?<br>Really, always? |
| what | 2 | ?X | Why do you ask?<br>Does that interest you? |

# Approaches in CL

- Rule-Based
- Data-Driven

# Rule-Based Approach

- Explicit encoding of linguistic knowledge
- Usually consisting of a set of hand-crafted, grammatical rules
- Easy to test and debug
- Require considerable human effort
- Often based on limited inspection of the data with an emphasis on prototypical examples
- Often fail to reach sufficient domain coverage
- Often lack sufficient robustness when input data are noisy

# Data-Driven Approach

- Implicit encoding of linguistic knowledge
- Often using statistical methods or machine learning methods
- Require less human effort
- Are data-driven and require large-scale data sources
- Achieve coverage directly proportional to the richness of the data source
- Are more adaptive to noisy data

# Models and Algorithms

- **Models**: formalisms used to capture the various kinds of linguistic structure.
  - State machines (fsa, transducers, markov models)
  - Formal rule systems (context-free grammars, feature systems)
  - Logic (predicate calculus, inference)
  - Probabilistic versions of all of these + others (gaussian mixture models, probabilistic relational models, etc etc)
- **Algorithms** used to manipulate representations to create structure.
  - Search (dept-first, best-first, A*, dynamic programming)
  - Supervised learning, etc etc

# History: foundational insights 1940s-1950s

- Automaton:
  - Turing (1936) - model of algorithmic computation
  - McCulloch-Pitts neuron (1943) – a simplified model of the neuron as a kind of computing element that could be described in terms of propositional logic
  - Kleene (1951/1956) finite automata and regular expressions
  - Shannon (1948) link between automata and Markov models
  - Chomsky (1956)/Backus (1959)/Naur(1960): CFG
- Probabilistic/Information-theoretic models
  - Shannon (1948) – the metaphor of the noisy channel and decoding -> development of probabilistic algorithms for speech and language processing
  - Bell Labs speech recognition (1952) – statistical system for speech recognition

# History: the two camps 1957-1970

- Symbolic – two lines of research: formal language theory, parsing algorithms and artificial intelligence
  - Zellig Harris 1958 TDAP first parser?
    - Cascade of finite-state transducers
  - Chomsky
  - AI workshop at Dartmouth 1956 (McCarthy, Minsky, Shannon, Rochester)
  - Newell and Simon: Logic Theorist, General Problem Solver
- Statistical
  - Bledsoe and Browning (1959): Bayesian OCR
  - Mosteller and Wallace (1964): Bayesian authorship attribution
  - Brown corpus of American English (1960) – the first on-line collection (1 million words, 500 written texts)

# History: Four paradigms 1970-1983

- Stochastic
  - Hidden Markov Model 1972
    - Independent application of Baker (CMU) and Jelinek/Bahl/Mercer lab (IBM) following work of Baum and colleagues at IDA
- Logic-based
  - Colmerauer (1970,1975) Q-systems
  - Definite Clause Grammars (Pereira and Warren 1980)
  - Kay (1979) functional grammar, Bresnan and Kaplan (1982) unification
- Natural language understanding
  - Winograd (1972) Shrdlu which simulated a robot embedded in a world of toy bloks
  - Schank and Abelson (1977) scripts, story understanding (conceptual knowledge)
  - Influence of case-role work of Fillmore (1968) via Simmons (1973), Schank.
- Discourse Modeling
  - Grosz and colleagues: discourse structure and focus
  - Perrault and Allen (1980) Brief-Desire-Intention model (speech acts)

# History: Empiricism and Finite State Redux:1983-1993

- Finite State Models
  - Kaplan and Kay (1981): Phonology/Morphology
  - Church (1980): Syntax
- Return of Probabilistic Models:
  - Corpora created for language tasks
  - Early statistical versions of NLP applications (parsing, tagging, machine translation)
  - Increased focus on methodological rigor:
    - Can't test your hypothesis on the data you used to build it!
    - Training sets and test sets

# History: The field comes together: 1994-2005

- Statistical models standard
  - ACL conference:
    - 1990: 39 articles 1 statistical
    - 2003 62 articles 48 statistical
  - Machine learning techniques key
- Comercial exploration – speech recognition, spelling and grammar checking
- Information retrieval meets NLP
- Unified field:
  - NLP, MT, ASR, TTS, Dialog, IR

- Machine Translation:

http://translate.google.com/translate_t

- Text-to-Speech:

http://public.research.att.com/~ttsweb/tts/demo.php

- Question Answering:

http://www.languagecomputer.com/demos/question_answering/internet_demo/more_examples.html

- Anaphora resolution:

http://clg.wlv.ac.uk/MARS/index.php

- ELIZA:

http://www-ai.ijs.si/eliza/eliza.html

http://www.manifestation.com/neurotoys/eliza.php3

# Regular expressions

- A formal language for specifying text strings
- How can we search for any of these?
  - woodchuck
  - woodchucks
  - Woodchuck
  - Woodchucks

Figure from Dorr/Monz slides

# Regular Expressions

- Basic regular expression patterns
- Perl-based syntax (slightly different from other notations for regular expressions)
- Disjunctions `/[wW]oodchuck/`

| RE | Match | Example Patterns |
|---|---|---|
| `/[wW]oodchuck/` | Woodchuck or woodchuck | "Woodchuck" |
| `/[abc]/` | 'a', 'b', *or* 'c' | "In uomini, in soldati" |
| `/[1234567890]/` | any digit | "plenty of 7 to 5" |

Slide from Dorr/Monz

# Regular Expressions

- Ranges `[A-Z]`

| RE | Match | Example Patterns Matched |
|---|---|---|
| `/[A-Z]/` | an uppercase letter | "we should call it 'Drenched Blossoms'" |
| `/[a-z]/` | a lowercase letter | "my beans were impatient to be hoed!" |
| `/[0-9]/` | a single digit | "Chapter 1: Down the Rabbit Hole" |

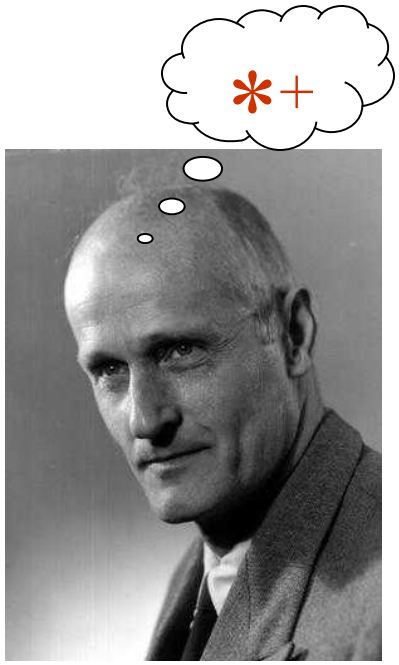
- **Negations `[^Ss]`**

| RE | Match (single characters) | Example Patterns Matched |
|---|---|---|
| `[^A-Z]` | not an uppercase letter | "Oyfn pripetchik" |
| `[^Ss]` | neither 'S' nor 's' | "I have no exquisite reason for't" |
| `[^\.]` | not a period | "our resident Djinn" |
| `[e^]` | either 'e' or '^' | "look up ^ now" |
| `a^b` | the pattern 'a^b' | "look up a^ b now" |

Slide from Dorr/Monz

# Regular Expressions

- Optional characters `?` ,`*` and `+`
  - `?` (0 or 1)
    - /colou?r/ ➔ color *or* colour
  - `*` (0 or more)
    - /oo*h!/ ➔ oh! *or* Ooh! *or* Ooooh!
  - **`+` (1 or more)**
    - **/o+h!/ ➔ oh! *or* Ooh! *or* Ooooh!**



*Stephen Cole Kleene*

★ Wild cards `.`
  - **/beg.n/ ➔ begin *or* began *or* begun**

Slide from Dorr/Monz

# Regular Expressions

- Anchors ^ and $
  - /^[A-Z]/ ➔ "Ramallah, Palestine"
  - /^[^A-Z]/ ➔ "¿verdad?" "really?"
  - /\.$/ ➔ "It is over."
  - /.$/ ➔ ?
- Boundaries \b and \B
  - /\bon\b/ ➔ "on my way" "Monday"
  - /\Bon\b/ ➔ "automaton"
- Disjunction |
  - /yours|mine/ ➔ "it is either yours or mine"

Slide from Dorr/Monz

# Disjunction, Grouping, Precedence

- Column 1 Column 2 Column 3 …
  How do we express this?

  `/Column [0-9]+ */`

  `/(Column [0-9]+ +)*/`
- Precedence
  - Parenthesis `()`
  - Counters `* + ? {}`
  - Sequences and anchors `the ^my end$`
  - Disjunction `|`
- REs are greedy!

Slide from Dorr/Monz

# Example

- Find me all instances of the word “the” in a text.
  - `/the/`
    - `Misses capitalized examples`
  - `/[tT]he/`
    - `Returns other or theology`
  - `/\b[tT]he\b/`
  - `/[^a-zA-Z][tT]he[^a-zA-Z]/`
  - `/(^|[^a-zA-Z])[tT]he[^a-zA-Z]/`

Slide from Dorr/Monz

# Errors

- The process we just went through was based on two fixing kinds of errors
  - Matching strings that we should not have matched (there, then, other)
    - False positives
  - Not matching things that we should have matched (The)
    - False negatives

# Errors cont.

- We'll be telling the same story for many tasks, all quarter. Reducing the error rate for an application often involves two antagonistic efforts:
  - Increasing accuracy (minimizing false positives)
  - Increasing coverage (minimizing false negatives).

# More complex RE example

- Regular expressions for prices
- /$[0-9]+/
  - Doesn’t deal with fractions of dollars
- /$[0-9]+\.[0-9][0-9]/
  - Doesn’t allow $199, not word-aligned
- \b$[0-9]+(\.[0-9][0-9])?\b)

# Advanced operators

should be _

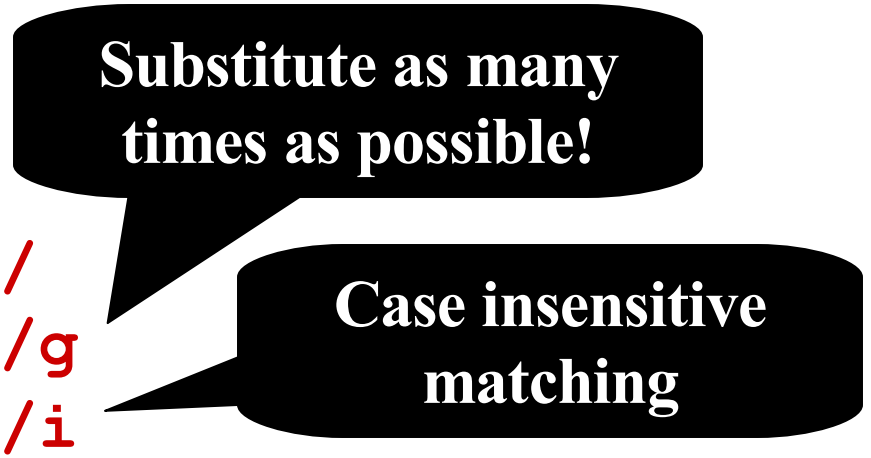
| RE | Expansion | Match | Example Patterns |
|---|---|---|---|
| \d | [0-9] | any digit | Party␣of␣5 |
| \D | [^0-9] | any non-digit | Blue␣moon |
| \w | [a-zA-Z0-9␣] | any alphanumeric or space | Daiyu |
| \W | [^\w] | a non-alphanumeric | !!!! |
| \s | [␣\r\t\n\f] | whitespace (space, tab) | |
| \S | [^\s] | Non-whitespace | in␣Concord |

Slide from Dorr/Monz

# Substitutions and Memory

- Substitutions

```
s/colour/color/
s/colour/color/g
s/colour/color/i
```




★Memory (**$1**, **$2**, etc. refer back to matches)

```
/the (.*)er they were, the $1er they will be/

/the (.*)er they (.*), the $1er they $2/
```

Slide from Dorr/Monz

# Eliza [Weizenbaum, 1966]

User: Men are all alike

ELIZA: IN WHAT WAY

User: They're always bugging us about something or other

ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE?

User: Well, my boyfriend made me come here

ELIZA: YOUR BOYFRIEND MADE YOU COME HERE

User: He says I'm depressed much of the time

ELIZA: I AM SORRY TO HEAR THAT YOU ARE DEPRESSED

# Eliza-style regular expressions

Step 1: replace first person with second person references

```
s/\bI('m| am)\b /YOU ARE/g
s/\bmy\b /YOUR/g
S/\bmine\b /YOURS/g
```

Step 2: use additional regular expressions to generate replies

```
s/.* YOU ARE (depressed|sad) .*/I AM SORRY TO HEAR
  YOU ARE \1/
s/.* YOU ARE (depressed|sad) .*/WHY DO YOU THINK YOU
  ARE \1/
s/.* all .*/IN WHAT WAY/
s/.* always .*/CAN YOU THINK OF A SPECIFIC EXAMPLE/
```

Step 3: use scores to rank possible transformations

Slide from Dorr/Monz

# Summary on REs so far

- Regular expressions are perhaps the single most useful tool for text manipulation
  - Dumb but ubiquitous
- Eliza: you can do a lot with simple regular-expression substitutions