Introduction to Computational Linguistics

Miriam Butt
5.6.02
Introduction to the Study of Language

The Beginnings of Computational Linguistics

- During WWII computational methods were very successful in cracking codes.

- (Turing was a key contributor to the war effort in England.)

- In the Cold War era, attention was focused on automatic translation:
  - It would have been nice to have a computational system that automatically parsed and translated Russian documents.

The Turing Test

al of the emerging field of Artificial Intelligence: build a computational system that could pass the **Turing Test**.

A computational system is successful if

1. A person sits in one room.

2. A computer and another person sit in two different rooms where they cannot be seen by the first person.

3. The first person converses with both the computer and the person and cannot tell the difference.

computational system has passed this test to date.

Machine Translation

- The Machine Translation Problem turned out to be harder than anticipated.

- **Classic Example:**
  - English Input: *The spirit is willing but the flesh is weak.*
  - Roughly, the English equivalent of the Russian output was: *The alcohol is good, but the steak is bad.*

Machine Translation (2)

veral problems by now are recognized as standard in MT.

me examples

The **Head-Switching** Problem:

John **likes** *to swim.* ⟶*John* **schwimmt** *gerne.*

The Problem of Reversability (Generation vs. Parsing)

ne of the famous problems has received a solution as yet.

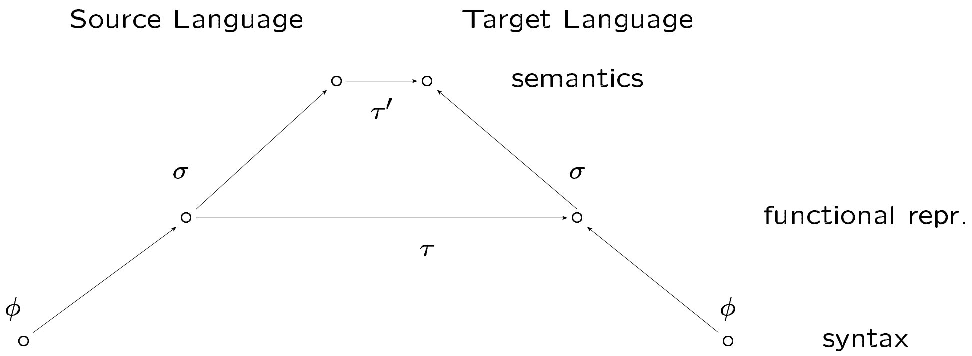
e best and most useful Machine Translation systems continue be the old ones.

What is so Difficult?

- Word-for-word analyses and translations are inadequate — they do not take the **structure** of the language into account.
- The computer has to know almost as much about language as humans do.
- The complexity of language was vastly underestimated in the early days of computational linguistics.
- We do not understand language well enough as yet.
- This makes it difficult to explain it to a computer.

What is so Difficult? (2)

as long been acknowledged that the most successful machine nslation would be on a basis that is as close to a **semantic** alysis as possible.



What is so Difficult? (3)

- We have very little knowledge about how to do computational semantics (though some work is done in this field).
- We have a better idea about how to model the syntactic structure of a language computationally.
- But even there we are far away from a realistic solution.
- Statistical approaches are taking over in applications:

  Deep Analysis vs. **Shallow Analysis**

The Parsing Problem (Deep Analysis)

e rule system of a language allows for the generation of an inite number of strings.

Simple Example: Adjectives

The small dog barked.

The small, grey dog barked.

The small, grey, bad dog barked.

. . .

The Parsing Problem (2)

- **Another Simple Example:** Prepositional Phrases

  1. The dog barked in the garden.
  2. The dog barked in the garden under the tree.
  3. The dog barked in the garden under the tree behind the wall.
  4. . . .

- Therefore: **no finite enumeration** of parses is possible.

The Parsing Problem (3)

other problem: language is **not context free**.

We need to know which nouns are subjects, which are objects.

I like beans. *Beans, I like.*

We need to know about the lexical semantics of verbs.

#The dog barked the apple.

We need to know the scope of modification (PP-attachment):

Shankar saw the monkey with the telescope.

is makes it an NP complete problem (i.e., a hard one).

How Linguistic Generalizations can Help

The	small	dog	barked.
Det	Adj	Noun	Verb

Possible Rule: S →Det Adj N V

The	small	dog	saw	the	grey	cat.
Det	Adj	Noun	Verb	Det	Adj	Noun

Possible Rule: S →Det Adj N V Det Adj N

The	small	dog	in	the	red	house	saw	the	grey	cat.
Det	Adj	Noun	P	Det	Adj	Noun	Verb	Det	Adj	Noun

Possible Rule: S →Det Adj N V P Det Adj N Det Adj N

... ad infinitum. (since language is not finite)

- **Note:** The same patterns appear over and over in the sentence.

How Linguistic Generalizations can Help (2)

e can use the information about recurring patterns.

r example, we can define a category Noun Phrase (NP), which l always contain the same basic things in the same basic order.

 ⟶ Det Adj N

is is called a constituent.

Prepositional Phrase then could consist of a Noun Phrase plus preposition.

 ⟶P NP

Rewrite Rules

- These types of rules are usually called **rewrite rules**.
- They are formulated as regular expressions.
- The rules can make use of the powerful syntax of regular expressions.
  - **Kleene star** *: none or infinitely many
  - +: one or infinitely many
  - **round brackets** ( ): optional item

Rewrite Rules (2)

basic sentence can now be characterized with just a handful finite rules.

r example:

S ⟶ NP VP

NP ⟶ Det Adj* N

VP ⟶ V (NP) (PP)

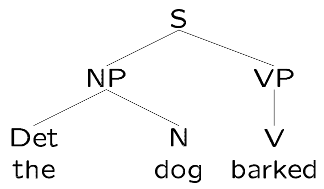
Rewrite Rules (3)

- Rewrite Rules within the linguistic context have the following properties:

1. They are **context-free**
2. They express **Linear Precedence** (LP)
3. They model **Immediate Dominance** (ID)

Rewrite Rules (4)

e result of the rules can thus be represented in tree format.



write rules can be used to write computational **parsers**.

Context-Free Parsing

- Differing parsing strategies have been developed in the literature: **bottom-up**, **top-down**.
- Differing parsing algorithms have been described and implemented.
- **A concrete example:** An LFG grammar as implemented in XLE (Xerox Linguistic Environment).
  - bottom-up
  - active chart parsing algorithm
  - complexity is polynomial for the length of the string

Adding Context Via Constraints

t: parsing without context is low-level and unintelligent.

e need to know that *Shankar* is the subject and *monkey* the ject.

ankar sees the monkey.

is can be encoded via constraints on the context-free backne.

$$\begin{array}{lll} \longrightarrow & \text{NP} & \text{VP} \\ & (\uparrow\text{SUBJ}) = \downarrow & \\ \text{P} \longrightarrow & \text{V} & \text{NP} \\ & & (\uparrow\text{OBJ}) = \downarrow \end{array}$$

World Knowledge

- We also need to know about what is likely to happen in our world.
- PP-Attachment: Shankar is unlikely to have the banana.

  Shankar sees the monkey with the banana.
- This kind of world knowledge is extremely difficult to model.

Grammar Components

typical (linguistically oriented) parser will contain

lexicons

a morphological component

preferences for parsing options

(a semantic interpretation module)

(information about world knowledge)

Grammar Components — Lexikon

- There are several ways to code up lexical items like *dog*, *barked*, etc.

1. **Hand-coding**: painful, slow, inefficient, but usually necessary for a small subset of words.
2. Using a **morphological analyzer** (will come back to this).
3. **Semi-automatic information extraction** from large corpora and other resources.

Semi-automatic Information Extraction

nding out about **Subcategorization Frames**:

barked: only one argument:
The dog barked. **not** *The dog barked the apple.*

saw: must have 2 arguments, etc.

ormation can be culled from

existing databases (e.g., Celex, Sadaw)

tagged texts/corpora

tree banks (e.g., the Penn Tree bank)

Part-of-Speech Tagging

- Information Extraction Techniques rely on low-level linguistic knowledge such as Part-of-Speech Information.
- Parts-of-Speech are types of words: adjective vs. noun vs. verb vs. adverb.
- Huge corpora can be tagged quickly with the help of POS (Part-of-Speech) Taggers.

Part-of-Speech Tagging (2)

ry useful information can be extracted based on this extra bit linguistic knowledge: is this word a verb or a noun?

Time flies like an arrow.

Fruit flies like a banana.

Time	flies	like	an	arrow.
Noun	Verb	Comp	Det	Noun

Fruit	flies	like	a	banana.
Adj	Noun	Verb	Det	Noun

Part-of-Speech Tagging (3)

- POS tagging is stochastic
- The best results are around 97% correct.
- Tagged texts must be hand-checked for 100% accuracy.
- **Example:** Feldweg's LIKELY System at Tübingen

Grammar Components — Morphology

xical Items can be listed either as *Lemma* or *Full Forms*

ll Forms: *bark*, *barks*, *barked*.

mma: *bark*

Grammar Components — Morphology

- A Morphological Analyzer is able to

1. analyze each Full Form
2. return the Lemma and the abstract morphological information

   barks

   1. `bark+Verb+Pres+3P+Sg`
   2. `bark+Noun+Pl`

Morphological Analyzers via Finite-State Technology

ry efficient morphological analyzers are built with finite-state chnology.

rox: `http://www.rxrc.xerox.com/research/mltt/fst/`

mplexity is linear for the length of the input string.

rphological analyzers come with a huge lexicon.

ugging a morphological analyzer into the grammar automatily yields the use of a huge lexicon.

Speech

- Efforts at Speech Recognition/Production in the past were pursued within a separate field (mostly from an engineering perspective).
  - This is now changing, but linguistically informed speech recognition is rare.
- Most speech recognition is based on statistic methods (HMM).
- One exception: Lahiri and Reetz (Konstanz): the FUL Model.
- Applications: Text-to-Speech, Intelligent Question-Answer Systems.

Outlook

e products on the market involving Parsing, Machine Transion and Generation work, but not well.

w-level solutions have contributed to

creation of large, tagged corpora

semi-automatic generation of lexicons

Web-based applications: information mining, key-word spotting, knowledge extraction.