CPE 372/641 Natural Language Processing

From Morphology to Syntax

Asst. Prof. Dr. Nuttanart Muansuwan

contents from Mark Granroth-Wilding slides, University of Helsinki

MORPHOLOGY: some concepts

- **Morpheme**: smallest grammatical unit in language
- **Affix**: morpheme that occurs only together with others
- Word = **stem** [+ **affixes**]

  radios → radio + s

- **Compound**: word with multiple stems

  thunderstorms → thunder + storm + s

Types of affix:

- **prefix**: un+help+ful
- **suffix**: taste+ful, taste+ful+ness
- **infix**: internal affix (e.g. Arabic)
- **circumfix**: prefix & suffix
  E.g. German: ge+kauf+t

Two types of morphology

Two types of morphology:

1. **Inflectional**: regular patterns for word classes
   - Changes *grammatical* roles
   - E.g. noun cases: *kauppa*, *kauppa+a*, *kaupa+n*, ...
2. **Derivational**: creates new words
   - Changes *meaning*
   - E.g. diminutive suffix: *tuuli* $\rightarrow$ *tuulo+nen*

Uses in NLP

A few uses of morphology in NLP:

- Morphosyntactic categorization (rough POS tagging)
- Morphological features
- Stemming/lemmatization
- Generation: apply syntax, features, agreement to base forms

the	*loyal*	*princes*	*occupied*	*the*	*throne*	*in*	*his*	*absence*
det	adj	noun	verb	det	noun	prep adv	det adj prn	noun

A few uses of morphology in NLP:

- Morphosyntactic categorization (rough POS tagging)
- Morphological features
- Stemming/lemmatization
- Generation: apply syntax, features, agreement to base forms

the	*loyal*	*princes*	*occupied*	*the*	*throne*	*in*	*his*	*absence*
def-sg		pl	past	def-sg	sg		m-sg	sg
def-pl			pst-prt	def-pl				

A few uses of morphology in NLP:

- Morphosyntactic categorization (rough POS tagging)
- Morphological features
- Stemming/lemmatization
- Generation: apply syntax, features, agreement to base forms

the	*loyal*	*princes*	*occupied*	*the*	*throne*	*in*	*his*	*absence*
the	loyal	prince	occupy	the	throne	in	his	absence

Theoretical limitations of Finite State Machines

- No memory, just current states
- No non-linear structure like recursion or embedding, which is important for languages

Formal Grammars

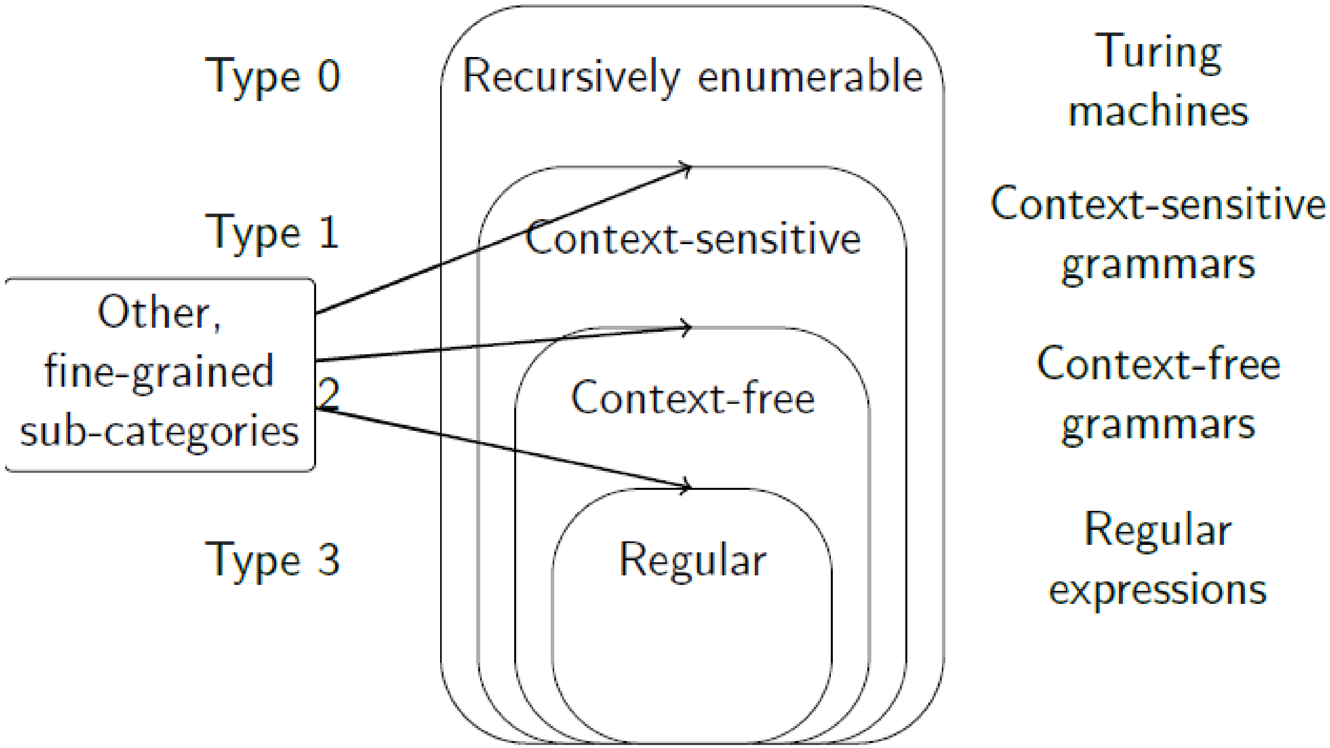
- Formal grammars categorized by expressive power(expressivity)
- FSAs < Context-Free Grammars (CFGs) <Turing machine E.g. Programming languages

•Greater expressivity →more languages as well as higher processing requirements

•FSA stores only current state

- CFG additionally requires stack memory

The Chomsky Hierarchy



Context-Free Grammar

A context-free grammar (CFG) is a list of rules that define the set of all well-formed sentences in a language.

Each rule has a left-hand side, which identifies a syntactic category, and a right-hand side, which defines its alternative component parts, reading from left to right.

E.g., the rule s --> np vp means that "a sentence is defined as a noun phrase followed by a verb phrase."

Figure 1 shows a simple CFG that describes the sentences from a small subset of English.

Figure 1. A grammar and a parse tree for "the giraffe dreams".

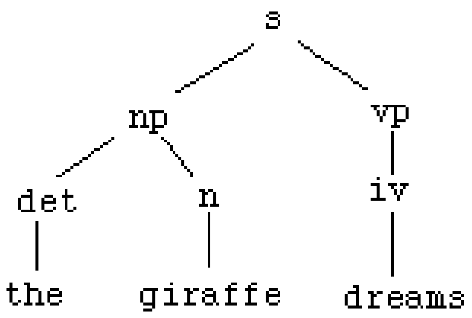
```
s    --> np vp

np   --> det n
vp   --> tv np
     --> iv

det  --> the
     --> a
     --> an

n    --> giraffe
     --> apple

iv   --> dreams
tv   --> eats
     --> dreams
```



Language Modeling

- **Language model** (LM): probability distribution over sentences of a language
- *Probability* of sentence appearing in language's text

$$p(w_0, \ldots, w_N)$$

But he may make trouble with your father

Highly plausible

Each night Arkilu departed, leaving a furry man-creature on guard

Less probable

(But both real, taken from same text)

Possible uses of language modeling

Uses of LMs:

- Speech recognition: plausibility judgement of possible outputs
- Word prediction for input device / communication aid
- Spelling correction
- Much more...

- One example of **statistics** in NLP
- Estimate probabilities from texts seen before
- Use **linguistic corpora**

Markov LM

Simple statistical LM: Markov chain

Markov assumption: probability of word depends only on previous word

- Far from true: *long-range dependencies*
- But, how well does it work in practice?

Simply model: $p(w_i | w_{i-1})$
Probability of sentence:

$$p(W_0^{n-1}) = \prod_{i=0}^{n-1} p(w_i | w_{i-1})$$

Bigram model

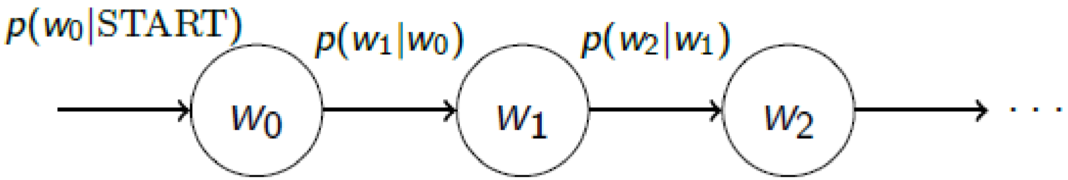
AKA *bigram* model

- bigram: pair of consecutive words ($w_i \; w_{i+1}$)
- model only looks at bigram statistics

Estimate probabilities from corpus counts

$$p(w_i|w_{i-1}) \simeq \frac{C(w_{i-1} \; w_i)}{C(w_{i-1})}$$

Can represent structure of model as **plate diagram**:



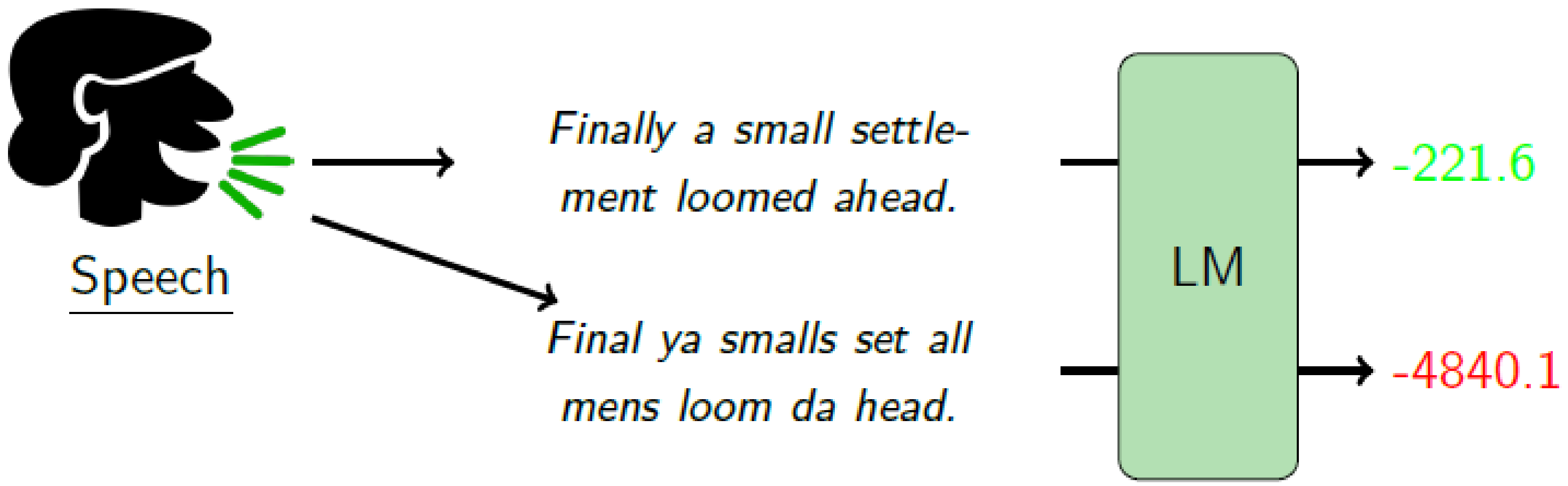
N-gram models

- Bigram model: predictions have limited **context** – Markov assumption
- Condition on longer context: *n-gram*
  - $p(w_i|w_{i-2}, w_{i-1})$ – *trigram*
  - $p(w_i|w_{i-3}, w_{i-2}, w_{i-1})$ – *4-gram*
- Estimate as before:

$$p(w_i|w_{i-3}, w_{i-2}, w_{i-1}) \simeq \frac{C(w_{i-3}\ w_{i-2}\ w_{i-1}\ w_i)}{C(w_{i-3}\ w_{i-2}\ w_{i-1})}$$

- Better predictions **if** enough examples of $w_{i-3}\ w_{i-2}\ w_{i-1}$ in training data

Language Modeling applications: speech recognition



Part-of-speech tagging

Noun Adjective

- Distinguish syntactic *function* of words in broad classes

For the present, we are. . . vs. *The present situation. . .*

- NLP subtask: **part-of-speech (POS) tagging**
- **Shallow** syntactic analysis: no explicit **structure**
- Includes *some* disambiguation important to meaning
- Useful practical first analysis step

POS tagging ambiguity

POS	Example
Noun	*The dog ate the bone*
Verb	*The dog ate the bone*
Adjective	*The big dog ate the bone*
Adverb	*He ate the bone quickly*
Pronoun	*He ate my bone*
Determiner	*The dog ate that bone*
Preposition	*He chewed with his teeth*
Coordinating conjunction	*He chewed and he growled*

Example

Return now to your quarters and I will send you word of the outcome

- POS tag this sentence, using the POSs above
- What other POSs could each word take (in other contexts)?

SOME AMBIGUITIES

Same POS tag
Different meanings

Different
POS tags

Easily misinterp

- *Ban on Nude Dancing on Governor's Desk*
- *Juvenile Court to Try Shooting Defendant*
- *Kids Make Nutritious Snacks*

2 interpretations

Several types of ambiguity – what?

Can **POS-tagging ambiguity** explain these?

[1] Thanks to Dan Klein & Roger Levy

Ambiguity

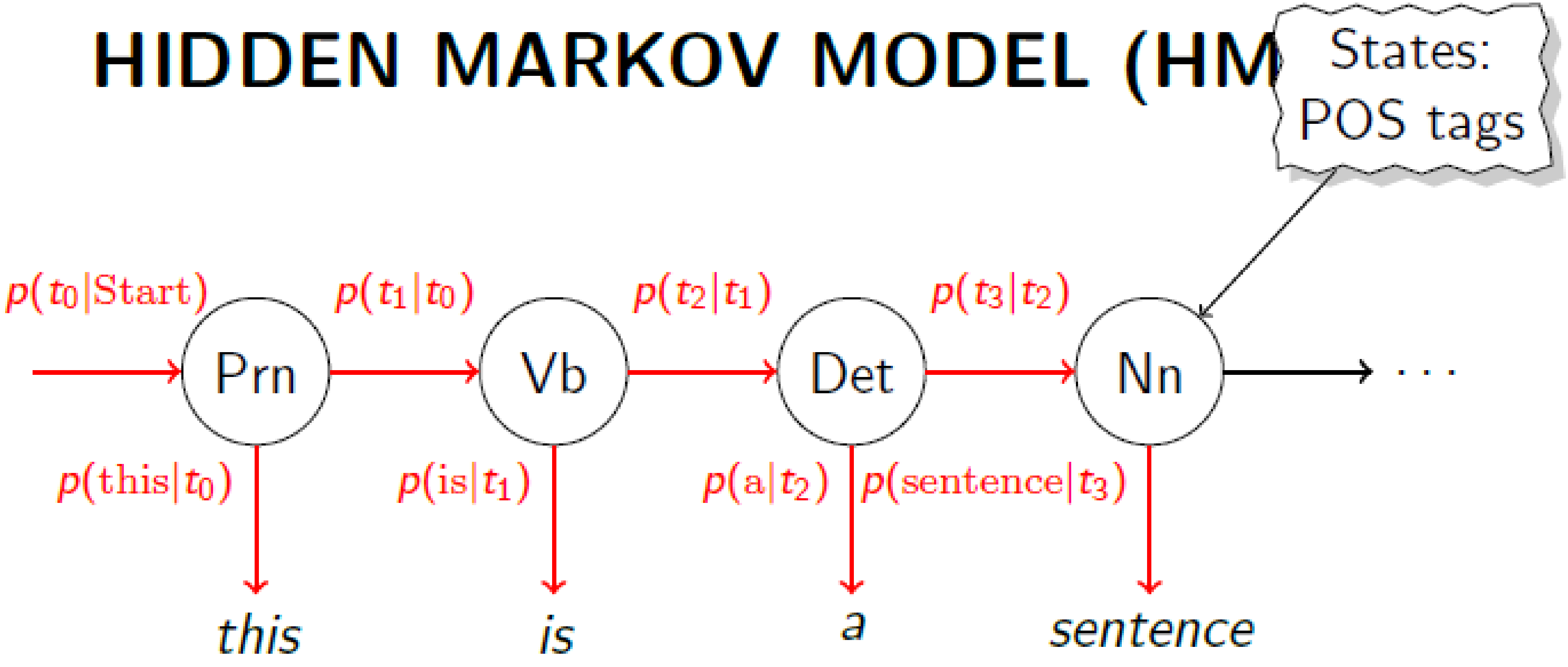
Juvenile Court to Try Shooting Defendant

- Two interpretations by human reader
- NLP system should output both
- Most syntactic parsers will produce dozens of structures!
- Majority have no intelligible interpretation

HIDDEN MARKOV MODEL

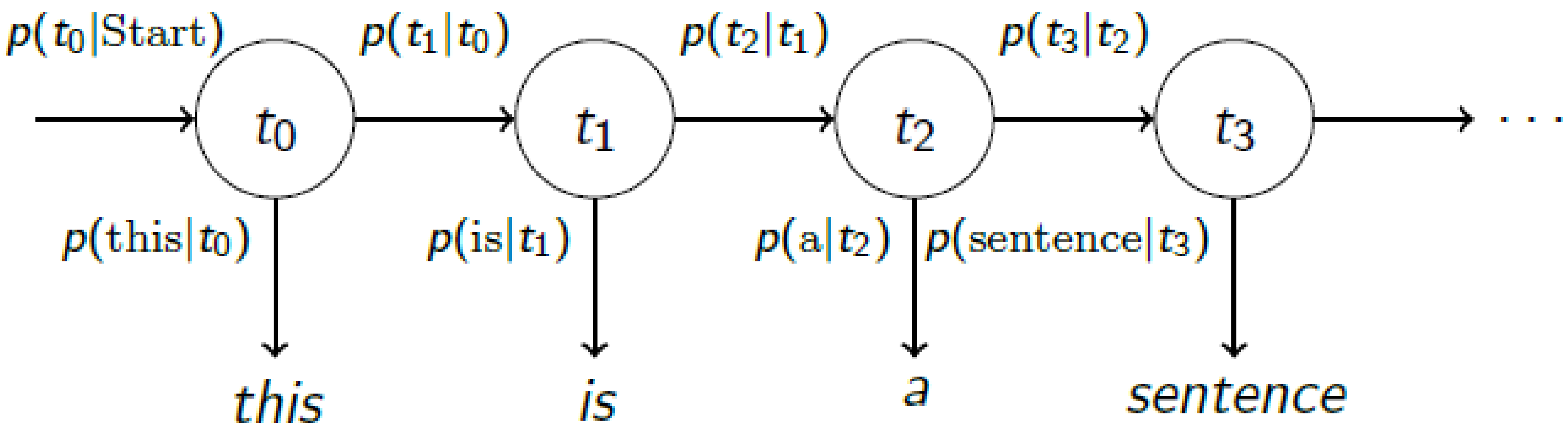
- **Probabilistic model**, widely used for POS tagging
- Maybe seen before: refresher here
- Model probability of (word,tag) sequence using
  - tag-word statistics
  - tag sequence statistics
- Makes some simplifying assumptions

HIDDEN MARKOV MODEL (HM



- Plate diagram: conditional prob dists
- **Markov assumption** on tag sequence: $p(t_i|t_{i-1})$
- **Conditional independence** of words: $p(w_i|\ t_i)$
- Independence assumptions pretty bad
- But works quite well in practice

HIDDEN MARKOV MODEL (HMM)



- Know $p(t_i|t_{i-1})$ and $p(w_i|t_i)$
  - Input sentence, POS tags unknown
  - Choose t_0, t_1, ... that **maximizes sentence probability**
  - Efficient inference with **Viterbi algorithm** (dynamic programming)

N-GRAM TAGGING MODEL

- Still an HMM: many more states
- Markov assumption over **n-grams**
- Each **state** represents (n-1)-gram
- Parameter estimation and inference as before