
Natural Language Processing

Philipp Koehn

23 April 2020



Overview



- Applications and advances
- Language as data
- Language models
- Part of speech
- Morphology
- Sentences and parsing
- Semantics

What is Language?



- **Nouns** — to describe things in the world
- **Verbs** — to describe actions
- **Adjectives** — to describe properties

+ glue to tie all this together

Why is Language Hard?



- **Ambiguity** on many levels
- **Sparse data** — many words are rare
- No clear understand how humans process language

Words



This is a simple sentence **WORDS**

Morphology



This is a simple sentence

be
3sg
present

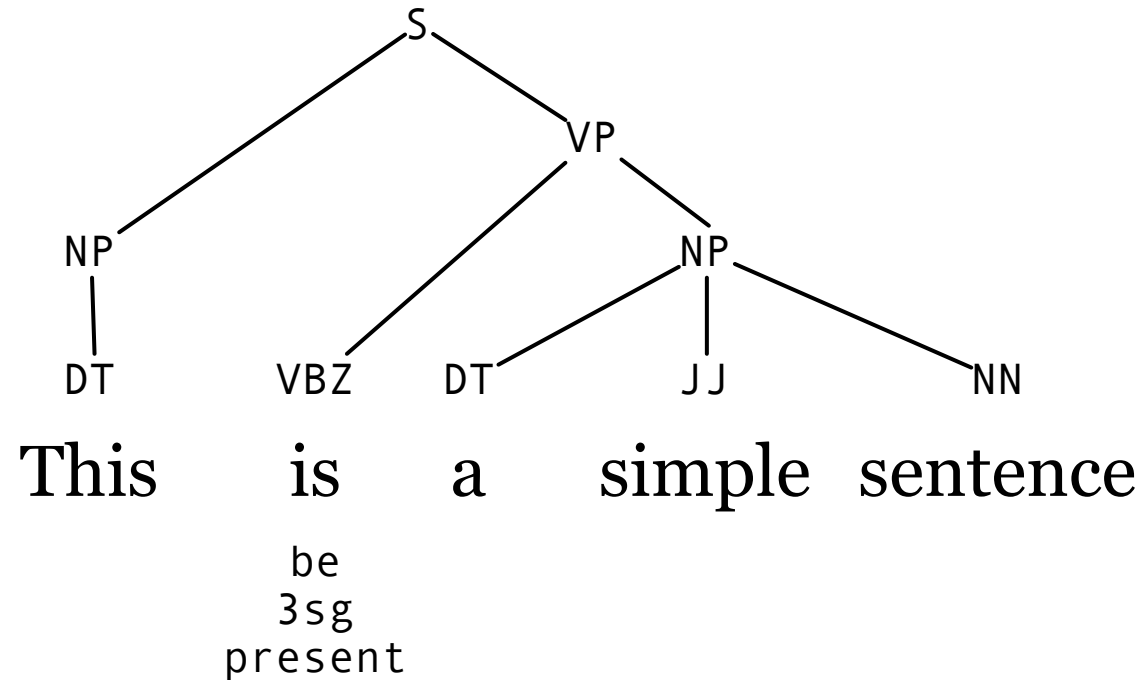
WORDS
MORPHOLOGY

Parts of Speech



DT	VBZ	DT	JJ	NN	PART OF SPEECH
This	is	a	simple	sentence	WORDS
	be 3sg present				MORPHOLOGY

Syntax



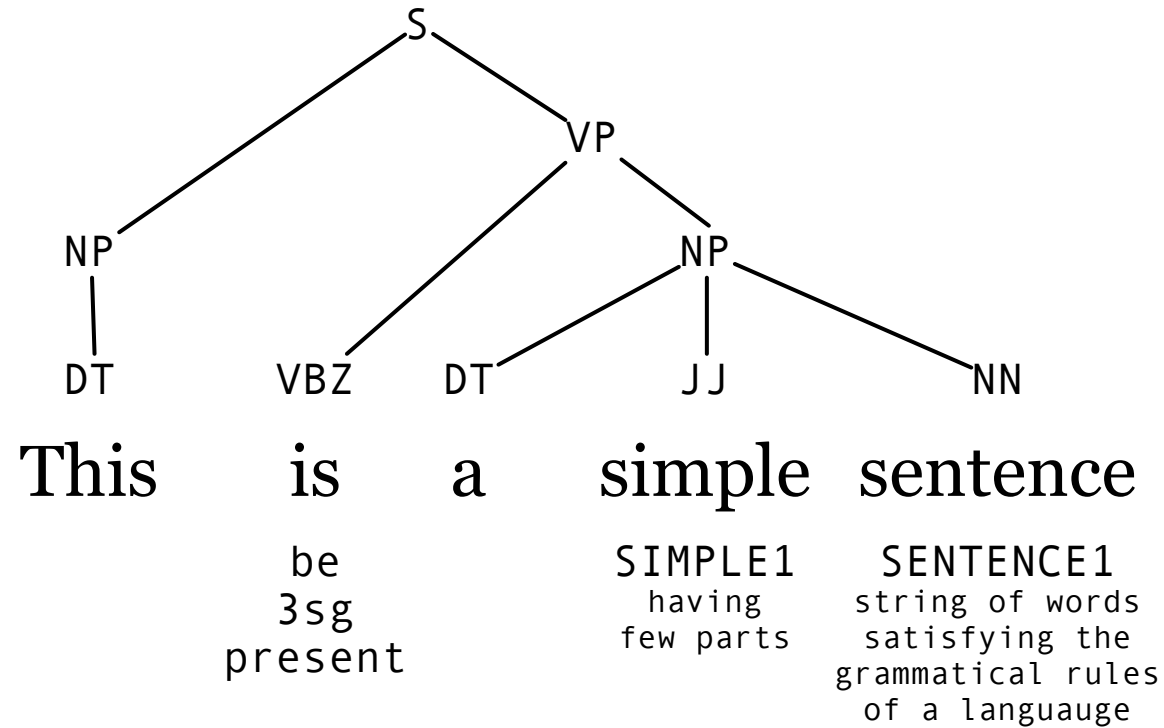
SYNTAX

PART OF SPEECH

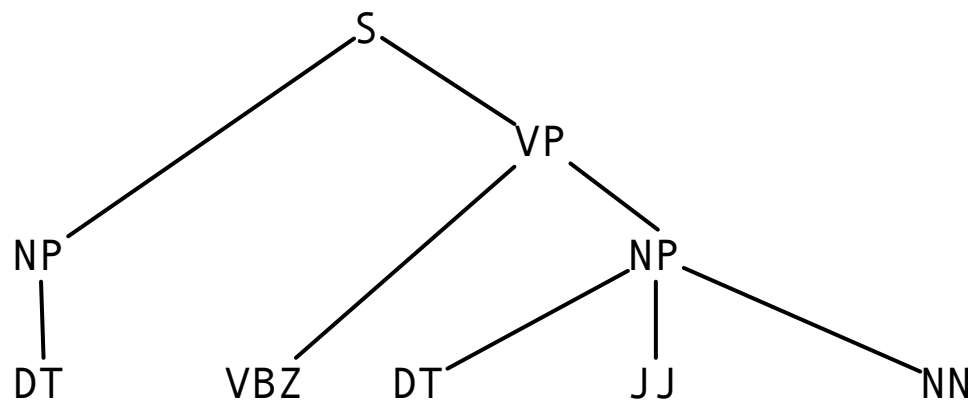
WORDS

MORPHOLOGY

Semantics



Discourse



SYNTAX

PART OF SPEECH

This is a simple sentence

WORDS

be
3sg
present

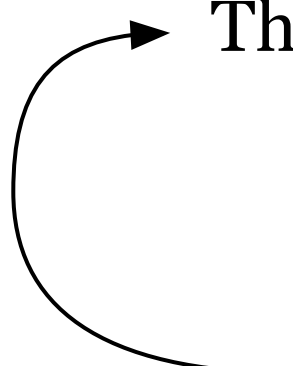
SIMPLE1
having
few parts

SENTENCE1
string of words
satisfying the
grammatical rules
of a language

MORPHOLOGY

SEMANTICS

CONTRAST



But it is an instructive one.

DISCOURSE

Recent Advances



Spoken dialogue devices
(Siri, Google Now, Echo, ...)

IBM Watson wins Jeopardy

Google machine translation

Web-scale question answering

IBM Watson



- IBM built a computer that won Jeopardy in 2011
- Question answering technology built on 200 million text pages, encyclopedias, dictionaries, thesauri, taxonomies, ontologies, and other databases

海军南海舰队战备巡逻远海训练编队 2 1 日在南海某海域开展立体抢滩登陆训练。参加这次训练的是来自井冈山舰兵力群的陆战队、航空兵等百余名官兵。围绕抢滩登陆 D 岛，部队进行了两栖破障、掠海突击、垂直突击等课目的训练。

South China Sea Fleet combat patrol far-sea training formation on the 21st stereoscopic beach landing training carried out in the South China Sea waters. To participate in the training, Marines, Air Force, more than 100 officers and men from Jinggangshan ship troops group. Around the beach landing D Island, forces amphibious barrier-breaking, the sea skimming assault, vertical assault subjects training.

La recette était simple. L'austérité budgétaire allait ramener la confiance, et celle-ci la croissance. On aurait l'équilibre des comptes publics et le retour de l'activité. Bon vieux bon sens britannique ! Ainsi parlait George Osborne, le péremptoire et sémillant chancelier de l'Echiquier. Il n'en n'a rien été. Depuis l'arrivée au pouvoir de David Cameron en 2010, l'impitoyable politique d'austérité budgétaire conduite à Londres n'a pas plus rétabli la croissance qu'elle n'a assaini les finances de l'Etat.

The recipe was simple. The austerity budget would restore confidence, and this growth. It would balance the public accounts and the return of the activity. Good old British common sense! Thus spoke George Osborne, the peremptory and spirited Chancellor of the Exchequer. It does not happen. Since coming to power in 2010 David Cameron, the relentless austerity policy London to conduct no more restored the growth it has cleaned up the finances of the State.

Eliza



Joseph Weizenbaum wrote the computer program **Eliza** in 1969 to demonstrate how easily people can be fooled into thinking that machines have some deep understanding.

young woman: Men are all alike.

eliza: In what way?

young woman: They're always bugging us about something specific or other.

eliza: Can you think of a specific example?

young woman: Well, my boyfriend made me come here.

eliza: Your boyfriend made you come here?

young woman: He says I'm depressed much of the time.

eliza: I am sorry to hear you are depressed.

young woman: It's true. I'm unhappy.

eliza: Do you think coming here will help you not to be unhappy?

Online demo: http://www-ai.ijs.si/eliza-cgi-bin/eliza_script

A Concrete Goal

- We would like to build
 - a machine that answers questions in natural language.
 - may have access to knowledge bases
 - may have access to vast quantities of English text
- Basically, a smarter Google
- This is typically called **Question Answering**

Example Question

- Question

When was Barack Obama born?

- Text available to the machine

Barack Obama was born on August 4, 1961

- This is easy.

- just phrase a Google query properly:

"Barack Obama was born on *"

- syntactic rules that convert questions into statements are straight-forward

Example Question (2)



- Question
 - What kind of plants grow in Maryland?
- Text available to the machine
 - A new chemical plant was opened in Maryland.
- What is hard?
 - words may have different meanings
 - we need to be able to disambiguate between them

Example Question (3)

- Question
 - Does the police use dogs to sniff for drugs?
- Text available to the machine
 - The police use canines to sniff for drugs.
- What is hard?
 - words may have the same meaning (synonyms)
 - we need to be able to match them

Example Question (4)

- Question

What is the name of George Bush's poodle?

- Text available to the machine

President George Bush has a terrier called Barnie.

- What is hard?

- we need to know that **poodle** and **terrier** are related, so we can give a proper response
- words need to be group together into semantically related classes

Example Question (5)



- Question
 - Which animals love to swim?
- Text available to the machine
 - Ice bears love to swim in the freezing waters of the Arctic.
- What is hard?
 - some words belong to groups which are referred to by other words
 - we need to have database of such **A is-a B** relationships, so-called ontologies

Example Question (6)



- Question

Did Poland reduce its carbon emissions since 1989?

- Text available to the machine

Due to the collapse of the industrial sector after the end of communism in 1989, all countries in Central Europe saw a fall in carbon emissions.

Poland is a country in Central Europe.

- What is hard?

- we need more complex semantic database
- we need to do inference

language as data

- Definition: strings of letters separated by spaces
- But how about:
 - punctuation: commas, periods, etc. typically separated (tokenization)
 - hyphens: **high-risk**
 - clitics: **Joe's**
 - compounds: **website, Computerlinguistikvorlesung**
- And what if there are no spaces:

伦敦每日快报指出,两台记载黛安娜王妃一九九七年巴黎死亡车祸调查资料的手提电脑,被从前大都会警察总长的办公室里偷走.

Word Counts

Most frequent words in the English Europarl corpus

any word		nouns	
Frequency in text	Token	Frequency in text	Content word
1,929,379	the	129,851	European
1,297,736	,	110,072	Mr
956,902	.	98,073	commission
901,174	of	71,111	president
841,661	to	67,518	parliament
684,869	and	64,620	union
582,592	in	58,506	report
452,491	that	57,490	council
424,895	is	54,079	states
424,552	a	49,965	member

Word Counts



But also:

There is a large tail of words that occur only once.

33,447 words occur once, for instance

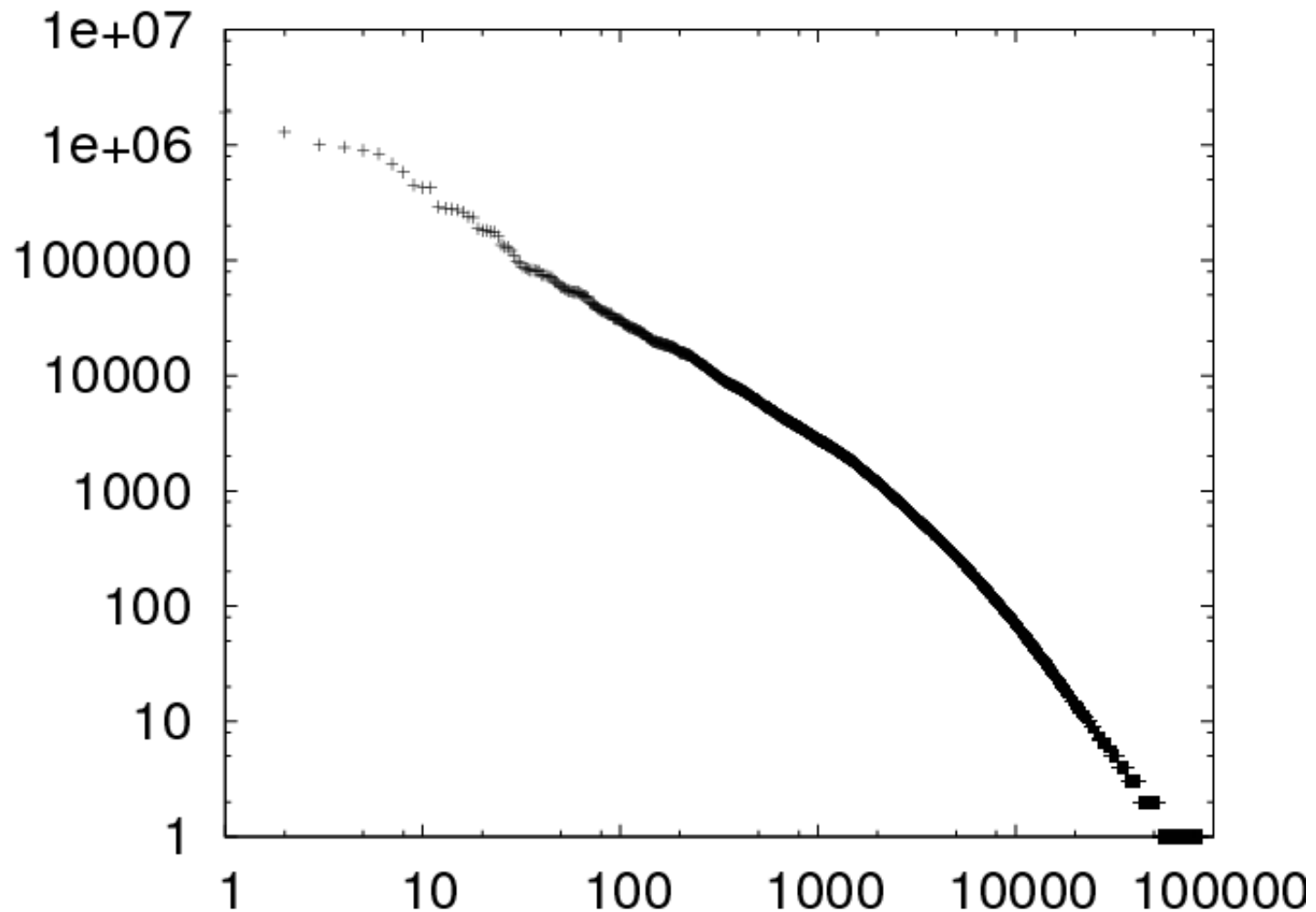
- cornflakes
- mathematicians
- Tazhikhistan

Zipf's Law

$$f \times r = k$$

f = frequency of a word
 r = rank of a word (if sorted by frequency)
 k = a constant

Zipf's Law as a Graph



why a line in log-scales? $fr = k \Rightarrow f = \frac{k}{r} \Rightarrow \log f = \log k - \log r$

language models

Language models

- **Language models** answer the question:

How likely is a string of English words good English?

- Help with ordering

$$p_{\text{LM}}(\text{the house is small}) > p_{\text{LM}}(\text{small the is house})$$

- Help with word choice

$$p_{\text{LM}}(\text{I am going home}) > p_{\text{LM}}(\text{I am going house})$$

N-Gram Language Models

- Given: a string of English words $W = w_1, w_2, w_3, \dots, w_n$
- Question: what is $p(W)$?
- Sparse data: Many good English sentences will not have been seen before

→ Decomposing $p(W)$ using the chain rule:

$$p(w_1, w_2, w_3, \dots, w_n) = p(w_1) p(w_2|w_1) p(w_3|w_1, w_2) \dots p(w_n|w_1, w_2, \dots, w_{n-1})$$

(not much gained yet, $p(w_n|w_1, w_2, \dots, w_{n-1})$ is equally sparse)

- **Markov assumption:**

- only previous history matters
- limited memory: only last k words are included in history (older words less relevant)
- **k th order Markov model**

- For instance 2-gram language model:

$$p(w_1, w_2, w_3, \dots, w_n) \simeq p(w_1) p(w_2|w_1) p(w_3|w_2) \dots p(w_n|w_{n-1})$$

- What is conditioned on, here w_{i-1} is called the **history**

Estimating N-Gram Probabilities

- Maximum likelihood estimation

$$p(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)}$$

- Collect counts over a large text corpus
- Millions to billions of words are easy to get
(trillions of English words available on the web)

Example: 3-Gram

- Counts for trigrams and estimated word probabilities

the green (total: 1748)			the red (total: 225)			the blue (total: 54)		
word	c.	prob.	word	c.	prob.	word	c.	prob.
paper	801	0.458	cross	123	0.547	box	16	0.296
group	640	0.367	tape	31	0.138	.	6	0.111
light	110	0.063	army	9	0.040	flag	6	0.111
party	27	0.015	card	7	0.031	,	3	0.056
ecu	21	0.012	,	5	0.022	angel	3	0.056

- 225 trigrams in the Europarl corpus start with **the red**
- 123 of them end with **cross**
- maximum likelihood probability is $\frac{123}{225} = 0.547$.

How good is the LM?

- A good model assigns a text of real English W a high probability
- This can be also measured with cross entropy:

$$H(W) = \frac{1}{n} \log p(W_1^n)$$

- Or, **perplexity**

$$\text{perplexity}(W) = 2^{H(W)}$$

Example: 3-Gram

prediction	p_{LM}	$-\log_2 p_{LM}$
$p_{LM}(i </s><s>)$	0.109	3.197
$p_{LM}(\text{would} <s>i)$	0.144	2.791
$p_{LM}(\text{like} i \text{ would})$	0.489	1.031
$p_{LM}(\text{to} would \text{ like})$	0.905	0.144
$p_{LM}(\text{commend} like \text{ to})$	0.002	8.794
$p_{LM}(\text{the} to \text{ commend})$	0.472	1.084
$p_{LM}(\text{rapporteur} commend \text{ the})$	0.147	2.763
$p_{LM}(\text{on} the \text{ rapporteur})$	0.056	4.150
$p_{LM}(\text{his} rapporteur \text{ on})$	0.194	2.367
$p_{LM}(\text{work} on \text{ his})$	0.089	3.498
$p_{LM}(. his \text{ work})$	0.290	1.785
$p_{LM}(</s> work \text{ .})$	0.99999	0.000014
average		2.634

Comparison 1-4-Gram

word	unigram	bigram	trigram	4-gram
i	6.684	3.197	3.197	3.197
would	8.342	2.884	2.791	2.791
like	9.129	2.026	1.031	1.290
to	5.081	0.402	0.144	0.113
commend	15.487	12.335	8.794	8.633
the	3.885	1.402	1.084	0.880
rapporteur	10.840	7.319	2.763	2.350
on	6.765	4.140	4.150	1.862
his	10.678	7.316	2.367	1.978
work	9.993	4.816	3.498	2.394
.	4.896	3.020	1.785	1.510
</s>	4.828	0.005	0.000	0.000
average	8.051	4.072	2.634	2.251
perplexity	265.136	16.817	6.206	4.758

Core Challenge



- How to handle low counts and unknown n-grams?
- Smoothing
 - adjust counts for seen n-grams
 - use probability mass for unseen n-grams
 - many discount schemes developed
- Backoff
 - if 5-gram unseen → use 4-gram instead
- Neural network models promise to handle this better

parts of speech

- **Open class words** (or content words)
 - nouns, verbs, adjectives, adverbs
 - refer to objects, actions, and features in the world
 - *open* class, new ones are added all the time ([email](#), [website](#)).
- **Close class words** (or function words)
 - pronouns, determiners, prepositions, connectives, ...
 - there is a limited number of these
 - mostly functional: to tie the concepts of a sentence together

Parts of Speech



- There are about 30-100 parts of speech
 - distinguish between names and abstract nouns?
 - distinguish between plural noun and singular noun?
 - distinguish between past tense verb and present tense word?
- Identifying the parts of speech is a first step towards syntactic analysis

Ambiguous Words

- For instance: *like*
 - verb: *I like the class.*
 - preposition: *He is like me.*
- Another famous example: *Time flies like an arrow*
- Most of the time, the local context disambiguated the part of speech

Part-of-Speech Tagging

- Task: Given a text of English, identify the parts of speech of each word
- Example
 - Input: Word sequence
Time flies like an arrow
 - Output: Tag sequence
Time/NN flies/VB like/P an/DET arrow/NN
- What will help us to tag words with their parts-of-speech?

Relevant Knowledge for POS Tagging



- The word itself
 - Some words may only be nouns, e.g. **arrow**
 - Some words are ambiguous, e.g. **like, flies**
 - Probabilities may help, if one tag is more likely than another
- Local context
 - two determiners rarely follow each other
 - two base form verbs rarely follow each other
 - determiner is almost always followed by adjective or noun

Bayes Rule

- We want to find the best part-of-speech tag sequence T for a sentence S :

$$\operatorname{argmax}_T p(T|S)$$

- Bayes rule gives us:

$$p(T|S) = \frac{p(S|T) p(T)}{p(S)}$$

- We can drop $p(S)$ if we are only interested in argmax_T :

$$\operatorname{argmax}_T p(T|S) = \operatorname{argmax}_T p(S|T) p(T)$$

Decomposing the Model

- The mapping $p(S|T)$ can be decomposed into

$$p(S|T) = \prod_i p(w_i|t_i)$$

- $p(T)$ could be called a *part-of-speech language model*, for which we can use an n-gram model (bigram):

$$p(T) = p(t_1) p(t_2|t_1) p(t_3|t_2) \dots p(t_n|t_{n-1})$$

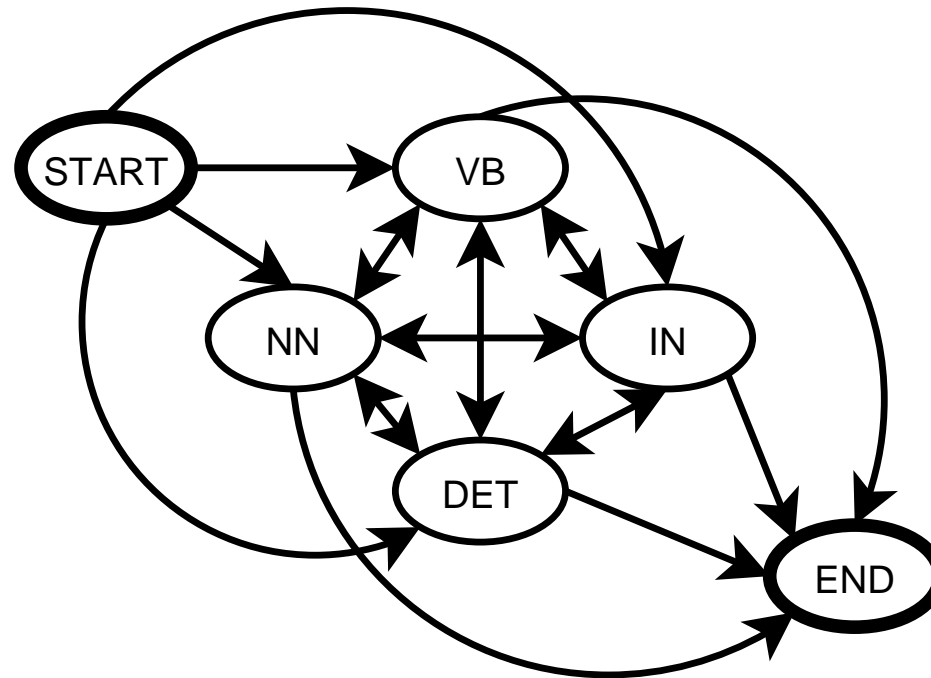
- We can estimate $p(S|T)$ and $p(T)$ with maximum likelihood estimation (and maybe some smoothing)

Hidden Markov Model (HMM)

- The model we just developed is a **Hidden Markov Model**
- Elements of an HMM model:
 - a set of states (here: the tags)
 - an output alphabet (here: words)
 - initial state (here: beginning of sentence)
 - state transition probabilities (here: $p(t_n|t_{n-1})$)
 - symbol emission probabilities (here: $p(w_i|t_i)$)

Graphical Representation

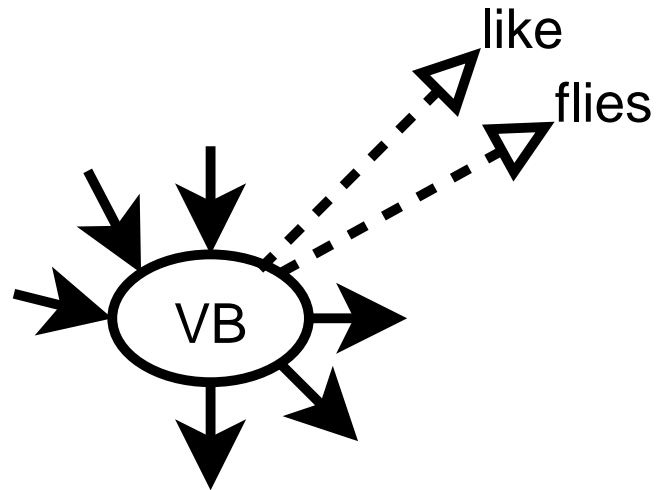
- When tagging a sentence, we are walking through the state graph:



- State transition probabilities: $p(t_n|t_{n-1})$

Graphical Representation

- At each state we emit a word:



- Symbol emission probabilities: $p(w_i|t_i)$

Search for the Best Tag Sequence

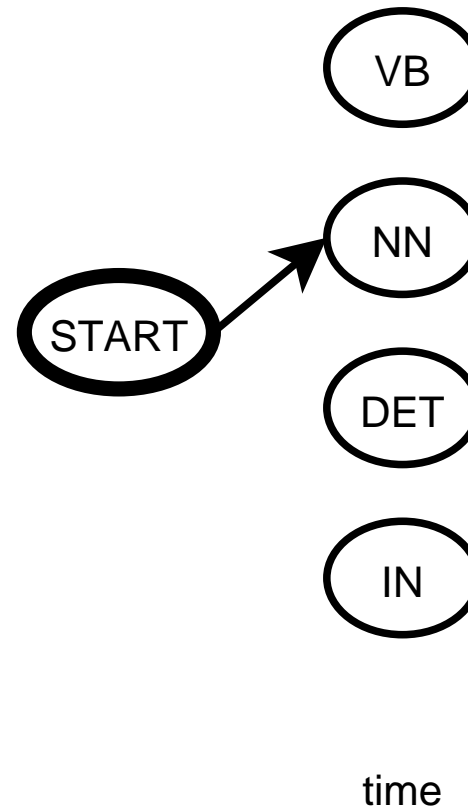
- We have defined a model, but how do we use it?
 - given: word sequence
 - wanted: tag sequence
- If we consider a specific tag sequence, it is straight-forward to compute its probability

$$p(S|T) p(T) = \prod_i p(w_i|t_i) p(t_i|t_{i-1})$$

- Problem: if we have on average c choices for each of the n words, there are c^n possible tag sequences, maybe too many to efficiently evaluate

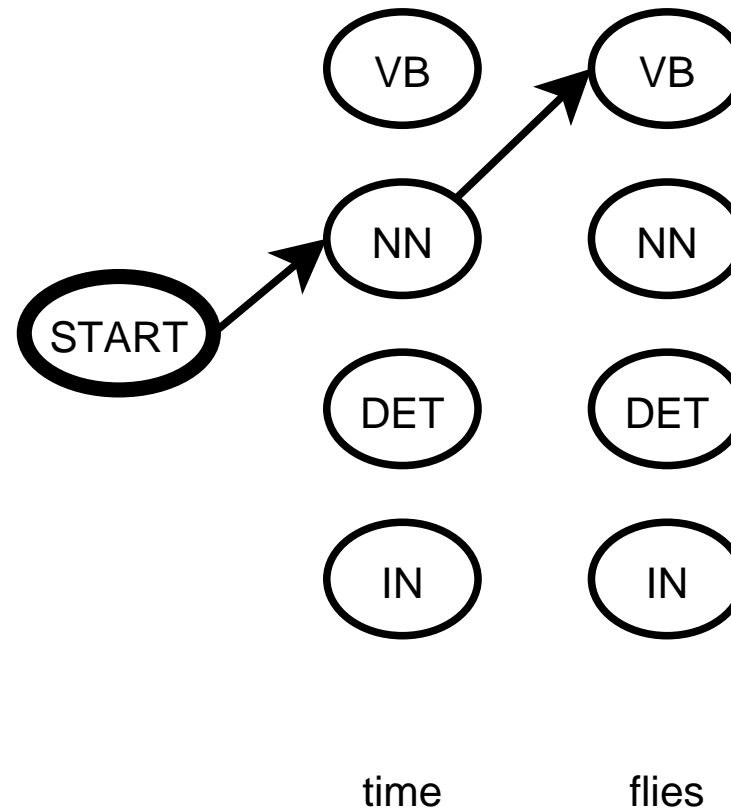
Walking Through the States

- First, we go to state NN to emit **time**:



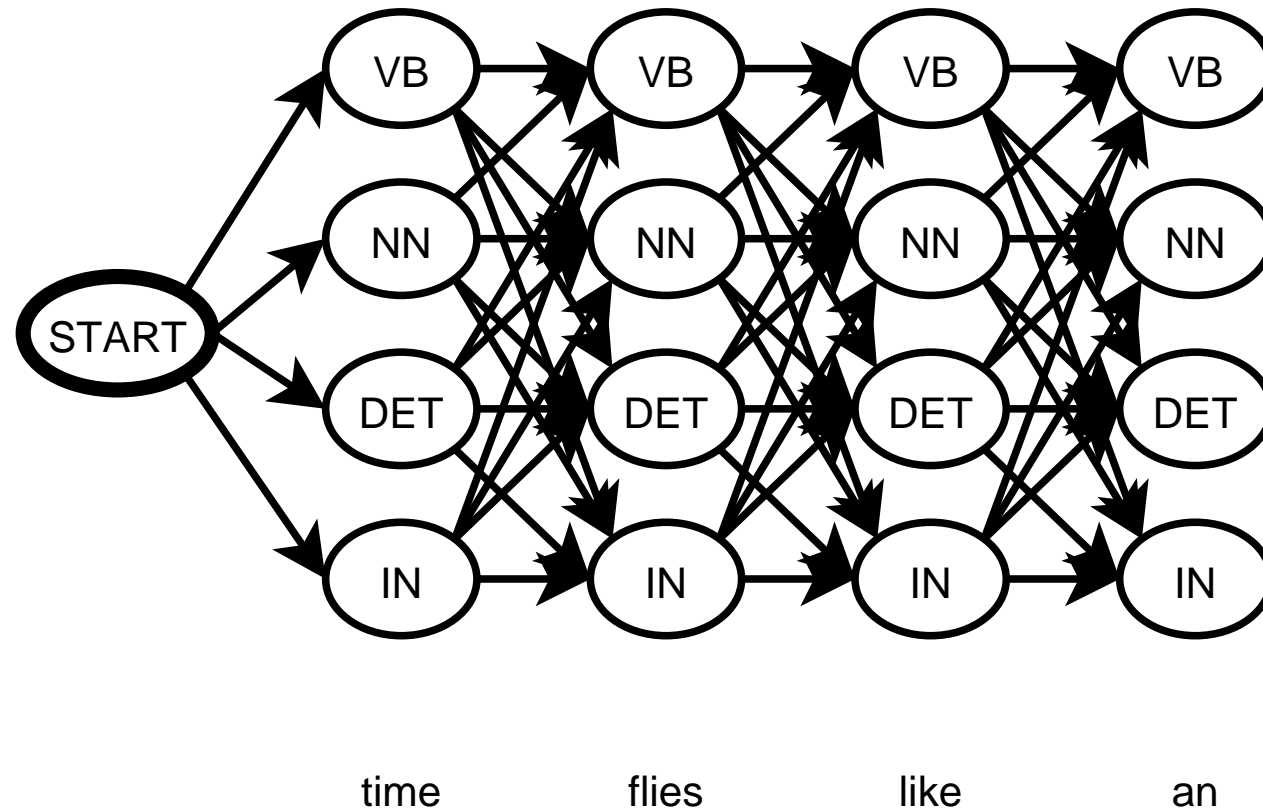
Walking Through the States

- Then, we go to state VB to emit *flies*:



Walking Through the States

- Of course, there are many possible paths:



Viterbi Algorithm

- Intuition: Since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path
- We record:
 - cheapest cost to state j at step s in $\delta_j(s)$
 - backtrace from that state to best predecessor $\psi_j(s)$
- Stepping through all states at each time steps allows us to compute
 - $\delta_j(s+1) = \max_{1 \leq i \leq N} \delta_i(s) p(t_j|t_i) p(w_{s+1}|t_j)$
 - $\psi_j(s+1) = \operatorname{argmax}_{1 \leq i \leq N} \delta_i(s) p(t_j|t_i) p(w_{s+1}|t_j)$
- Best final state is $\operatorname{argmax}_{1 \leq i \leq N} \delta_i(|S|)$, we can backtrack from there

morphology

How Many Different Words?

10,000 sentences from the Europarl corpus

Language	Different words
English	16k
French	22k
Dutch	24k
Italian	25k
Portuguese	26k
Spanish	26k
Danish	29k
Swedish	30k
German	32k
Greek	33k
Finnish	55k

Why the difference? Morphology.

Morphemes: Stems and Affixes



- Two types of morphemes
 - stems: **small, cat, walk**
 - affixes: **+ed, un+**

- Four types of affixes
 - suffix
 - prefix
 - infix
 - circumfix

Suffix



- Plural of nouns

cat+s

- Comparative and superlative of adjectives

small+er

- Formation of adverbs

great+ly

- Verb tenses

walk+ed

- All inflectional morphology in English uses suffixes

- In English: meaning changing particles

- Adjectives

un+friendly
dis+interested

- Verbs

re+consider

- German verb pre-fix **zer** implies destruction

Infix

- In English: inserting profanity for emphasis

abso+bloody+lutely
unbe+bloody+lievable

- Why not:

ab+bloody+solutely

- No example in English
- German past participle of verb:

ge+sag+t (German)

Not that Easy...

- Affixes are not always simply attached
- Some consonants of the lemma may be changed or removed
 - walk+ed
 - frame+d
 - emit+ted
 - eas(-y)+ier
- Typically due to phonetic reasons

Irregular Forms



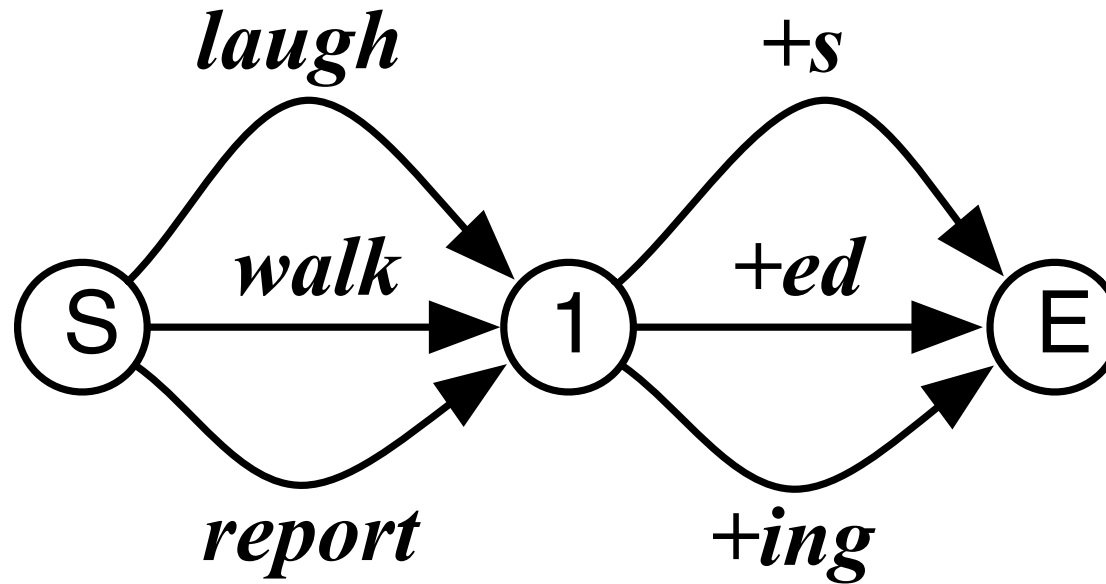
- Some words have irregular forms:
 - is, was, been
 - eat, ate, eaten
 - go, went, gone
- Only most frequent words have irregular forms
- A failure of morphology:
morphology reduces the need to create completely new words

Why Morphology?



- Alternatives
 - Some languages have no verb tenses
 - use explicit time references ([yesterday](#))
 - Case inflection determines roles of noun phrase
 - use fixed word order instead
 - Cased noun phrases often play the same role as prepositional phrases
- There is value in redundancy and subtly added information...

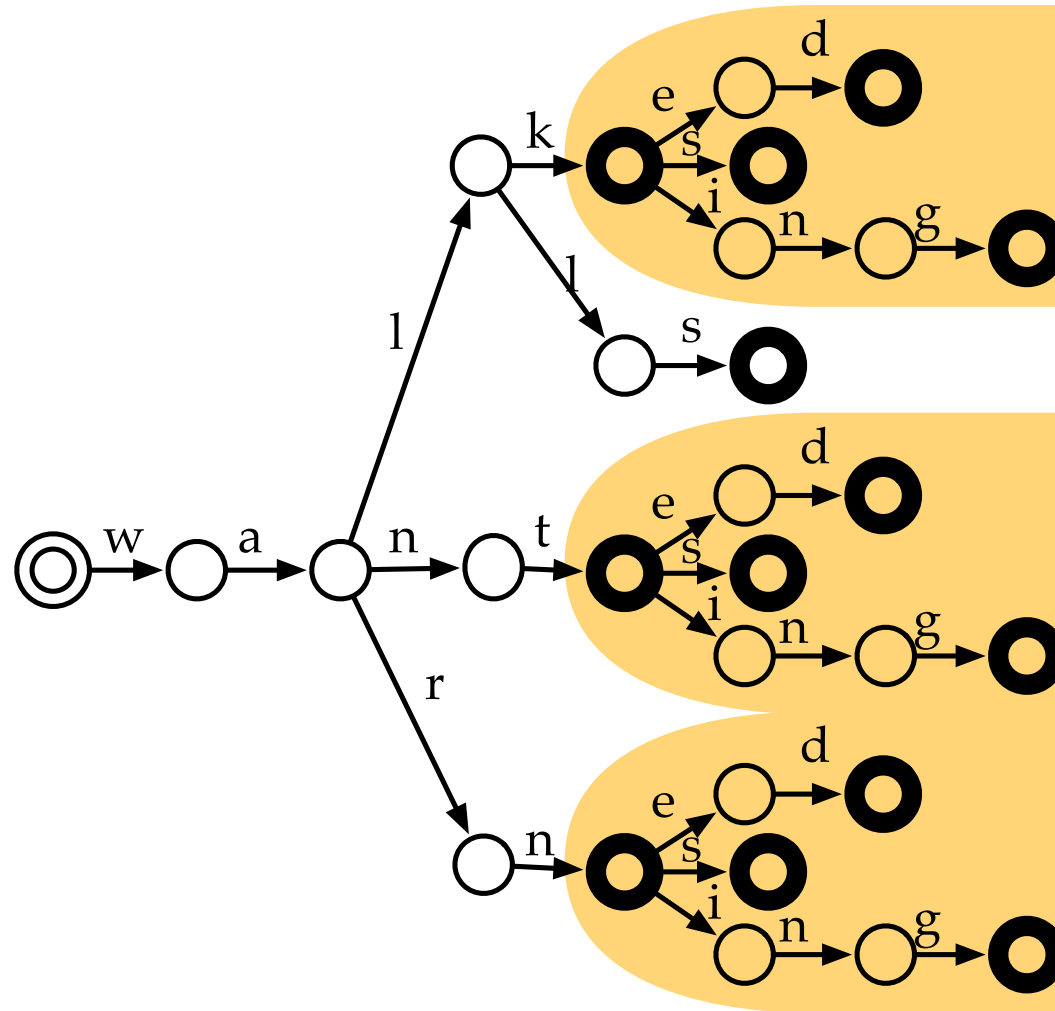
Finite State Machines



Multiple stems

- implements regular verb morphology
- *laughs, laughed, laughing*
walks, walked, walking
reports, reported, reporting

Automatic Discovery of Morphology



syntax

The Path So Far



- Originally, we treated language as a *sequence of words*
→ n-gram language models
- Then, we introduced the notion of *syntactic properties of words*
→ part-of-speech tags
- Now, we look at *syntactic relations* between words
→ syntax trees

A Simple Sentence



I like the interesting lecture

Part-of-Speech Tags

I	like	the	interesting	lecture
PRO	VB	DET	JJ	NN

Syntactic Relations

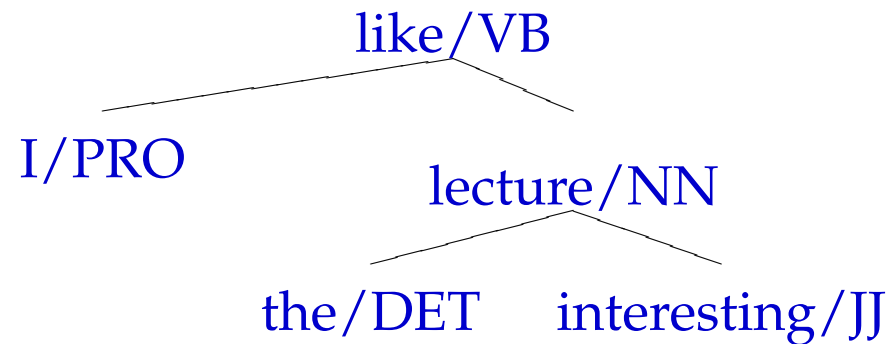
I	like	the	interesting	lecture
PRO	VB	DET	JJ	NN

- The adjective **interesting** gives more information about the noun **lecture**
- The determiner **the** says something about the noun **lecture**
- The noun **lecture** is the object of the verb **like**, specifying **what** is being liked
- The pronoun **I** is the subject of the verb **like**, specifying **who** is doing the liking

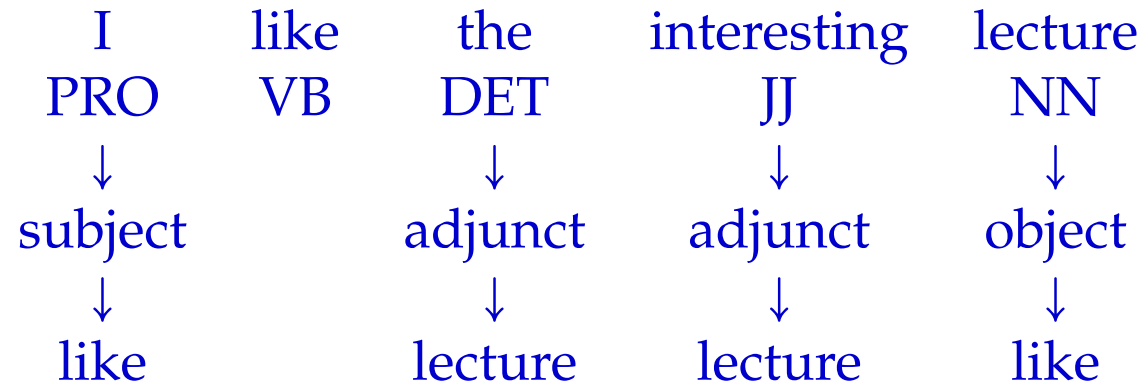
Dependency Structure

I	like	the	interesting	lecture
PRO	VB	DET	JJ	NN
↓		↓	↓	↓
like		lecture	lecture	like

This can also be visualized as a **dependency tree**:



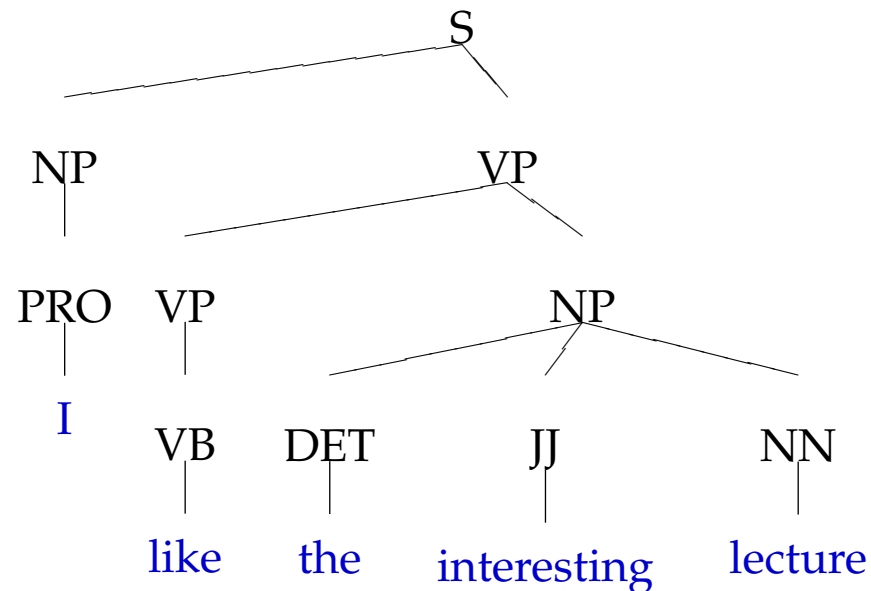
Dependency Structure



The dependencies may also be **labeled** with the type of dependency

Phrase Structure Tree

- A popular grammar formalism is **phrase structure grammar**
- Internal nodes combine leaf nodes into phrases, such as **noun phrases (NP)**



Building Phrase Structure Trees

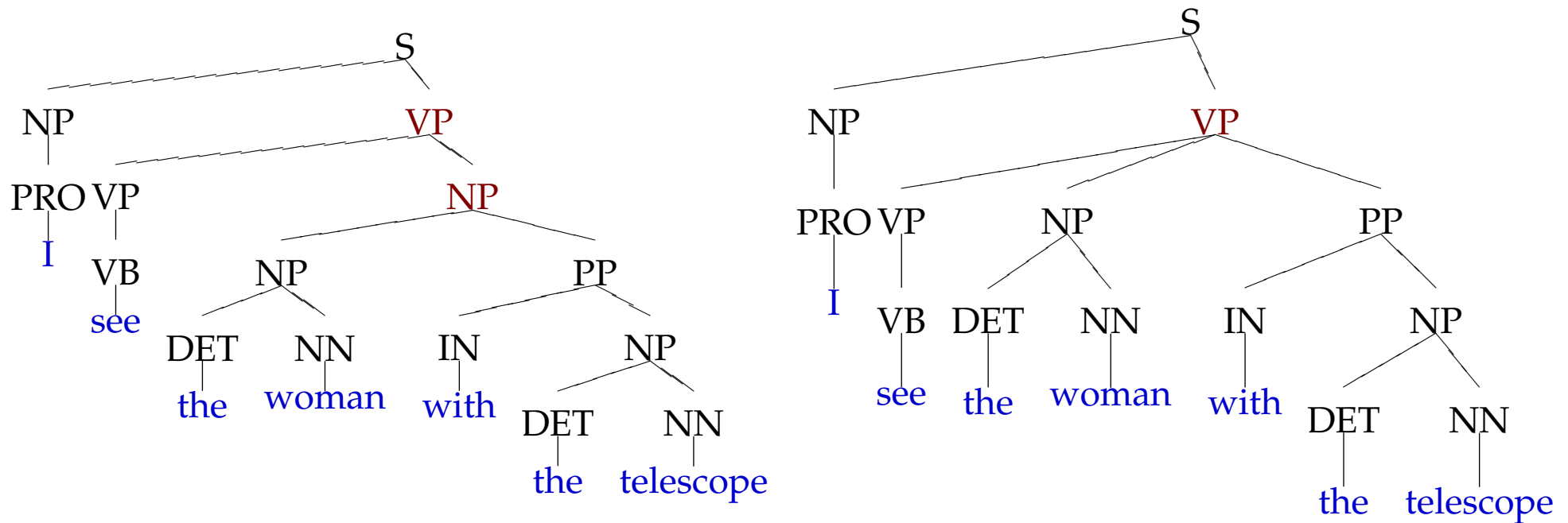


- Task: **parsing**
 - *given*: an input sentence with part-of-speech tags
 - *wanted*: the right syntax tree for it
- Formalism: **context free grammars**
 - **non-terminal nodes** such as **NP**, **S** appear inside the tree
 - **terminal nodes** such as **like**, **lecture** appear at the leafs of the tree
 - **rules** such as **NP** → **DET JJ NN**

- **Chomsky hierarchy of formal languages**
(terminals in caps, non-terminal lowercase)
 - **regular:** only rules of the form $A \rightarrow a, A \rightarrow B, A \rightarrow Ba$ (or $A \rightarrow aB$)
Cannot generate languages such as $a^n b^n$
 - **context-free:** left-hand side of rule has to be single non-terminal, anything goes on right hand-side. Cannot generate $a^n b^n c^n$
 - **context-sensitive:** rules can be restricted to a particular context, e.g. $\alpha A \beta \rightarrow \alpha a B c \beta$, where α and β are strings of terminal and non-terminals
- Moving up the hierarchy, languages are more expressive and parsing becomes computationally more expensive
- Is natural language context-free?

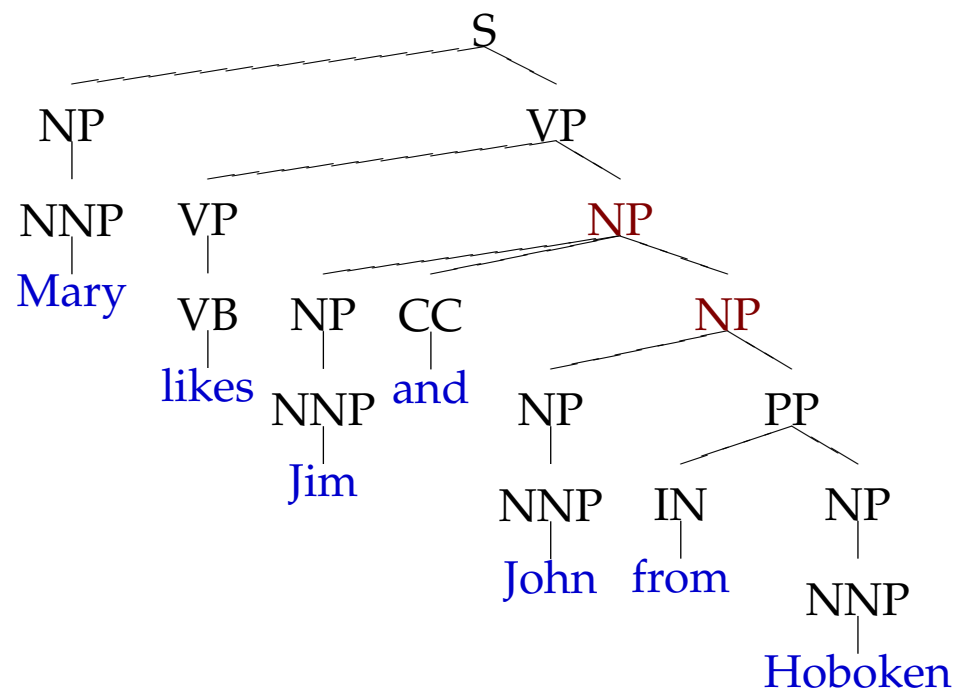
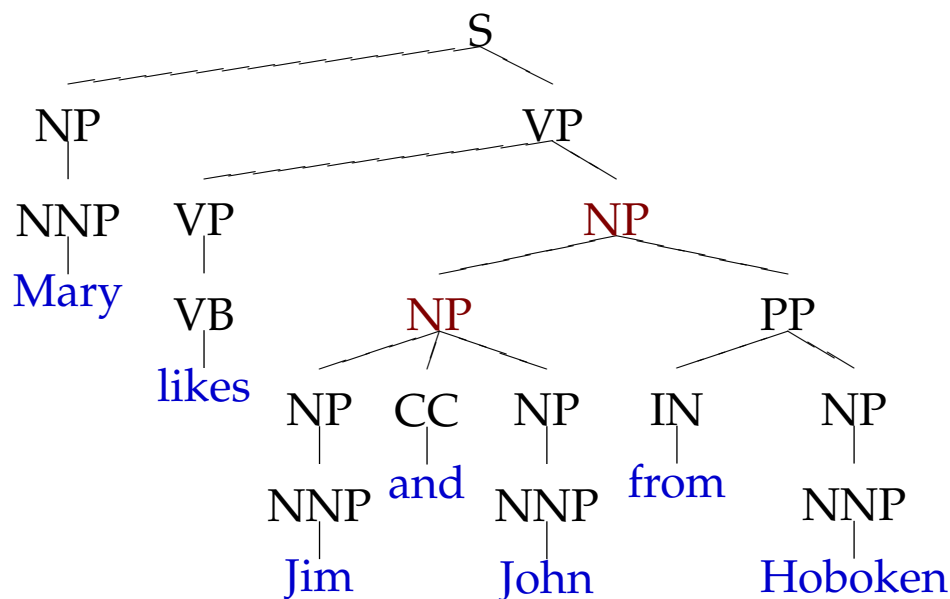
Why is Parsing Hard?

Prepositional phrase attachment: Who has the telescope?



Why is Parsing Hard?

Scope: Is **Jim** also from **Hoboken**?

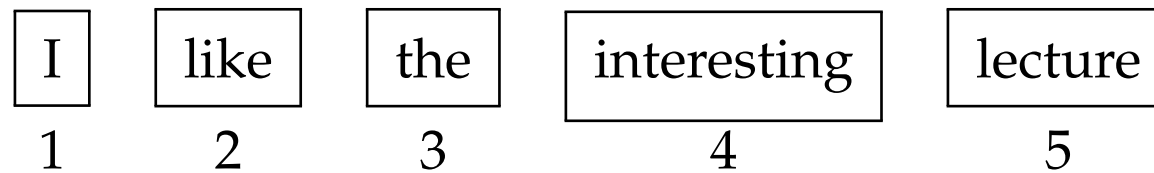


CYK Parsing

- We have input sentence:
I like the interesting lecture
- We have a set of context-free rules:
 $S \rightarrow NP VP, NP \rightarrow PRO, PRO \rightarrow I, VP \rightarrow VP NP, VP \rightarrow VB$
 $VB \rightarrow \text{like}, NP \rightarrow DET JJ NN, DET \rightarrow \text{the}, JJ \rightarrow, NN \rightarrow \text{lecture}$
- **Cocke-Younger-Kasami (CYK)** parsing
 - a **bottom-up** parsing algorithm
 - uses a **chart** to store intermediate result

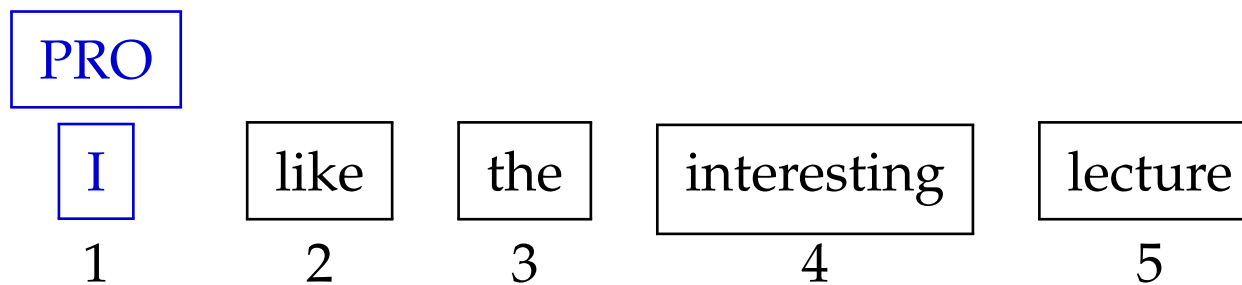
Example

Initialize chart with the words



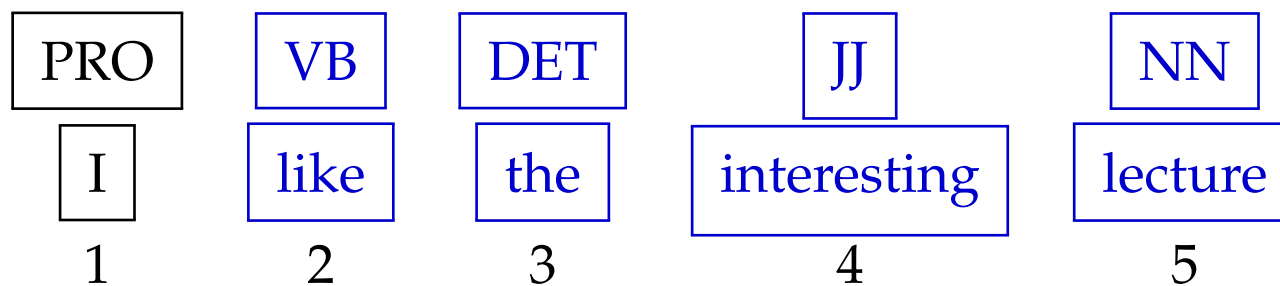
Example

Apply first terminal rule $PRO \rightarrow I$



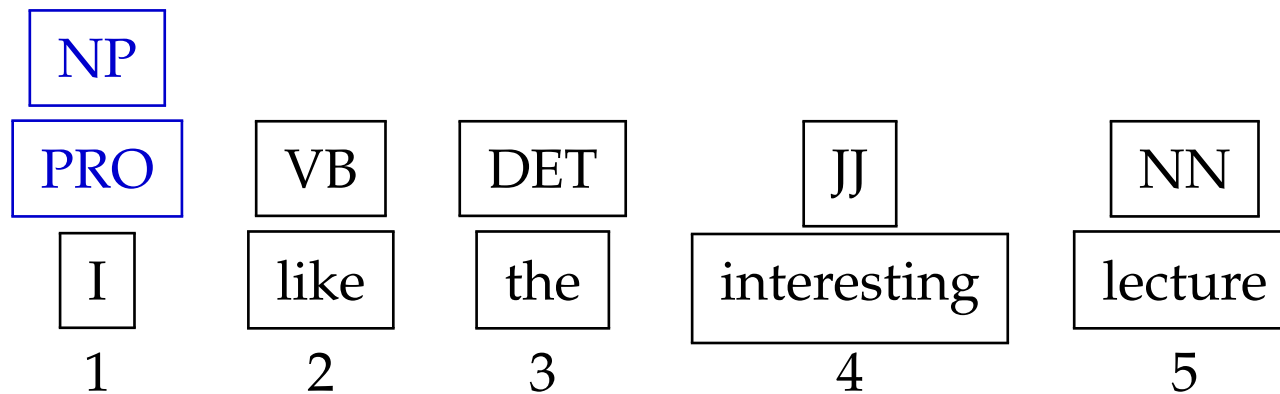
Example

... and so on ...



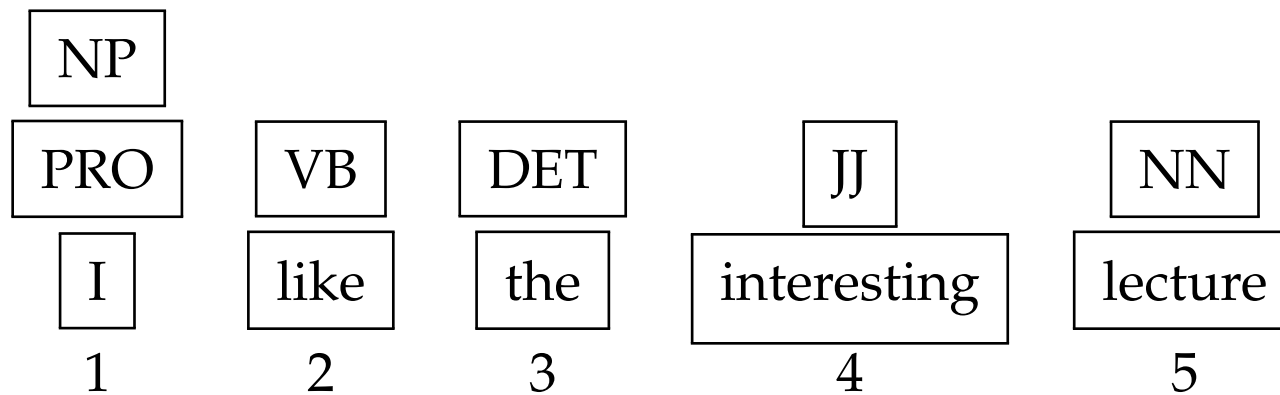
Example

Try to apply a non-terminal rule to the first word
The only matching rule is $NP \rightarrow PRO$



Example

Recurse: try to apply a non-terminal rule to the first word
No rule matches

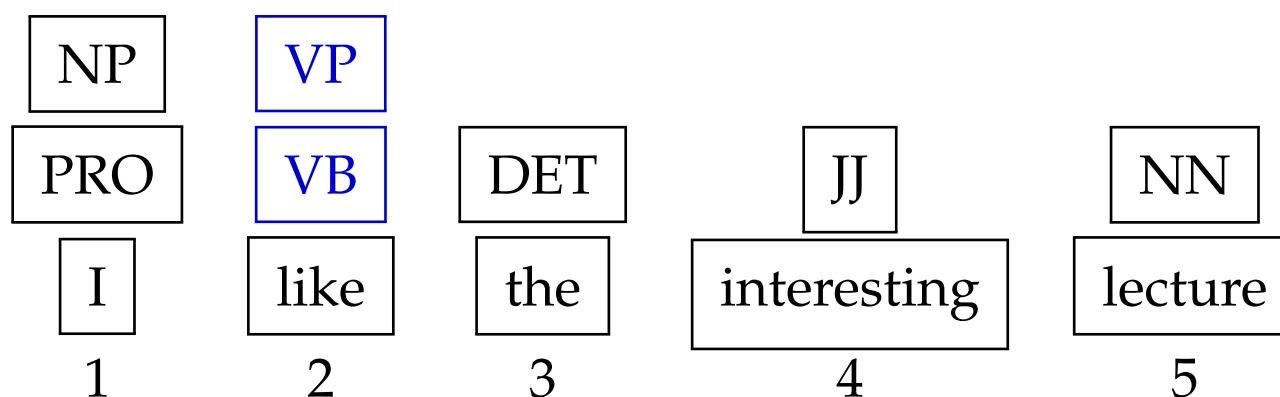


Example

Try to apply a non-terminal rule to the second word

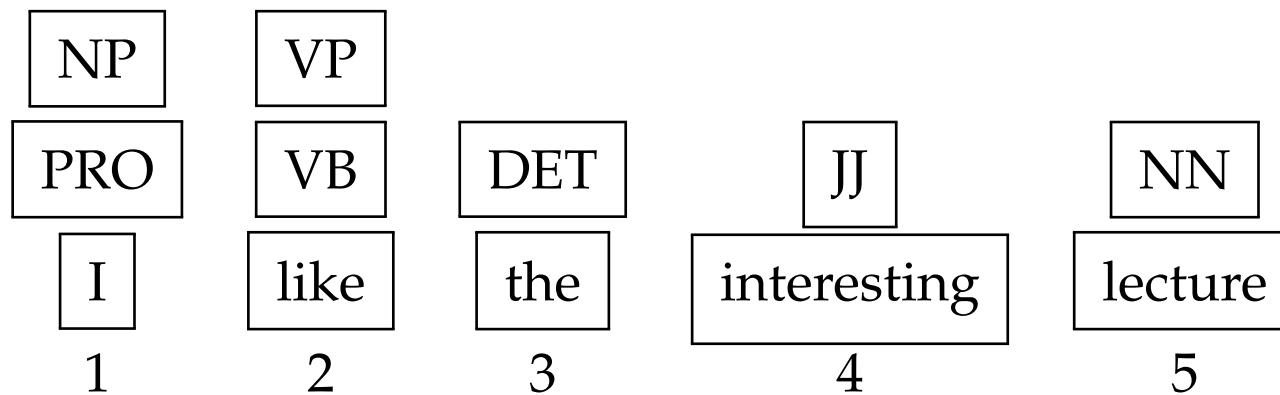
The only matching rule is $VP \rightarrow VB$

No recursion possible, no additional rules match



Example

Try to apply a non-terminal rule to the third word
No rule matches

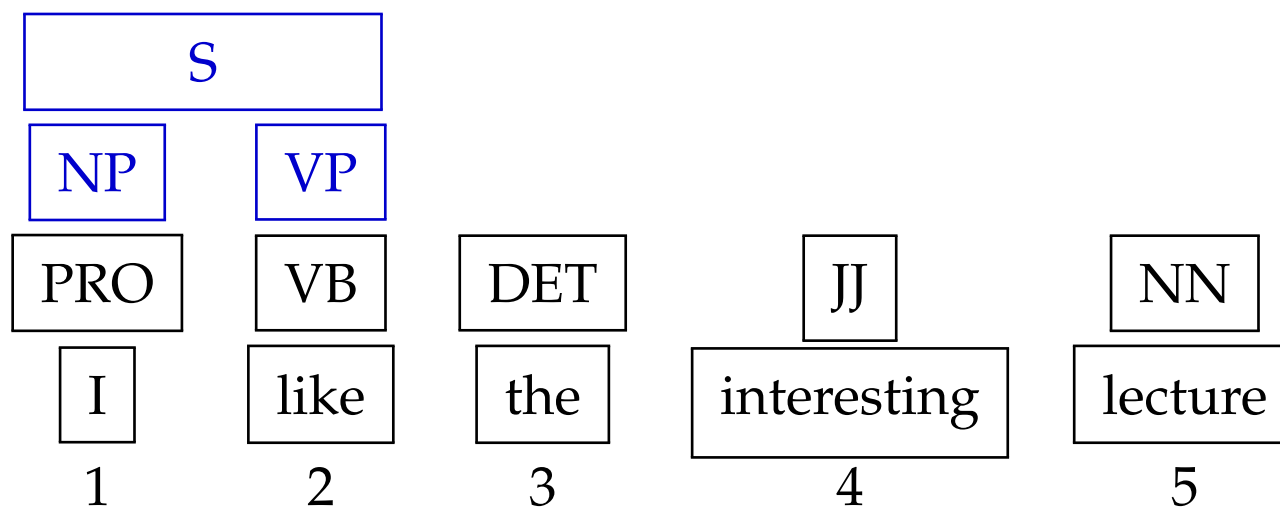


Example

Try to apply a non-terminal rule to the first two words

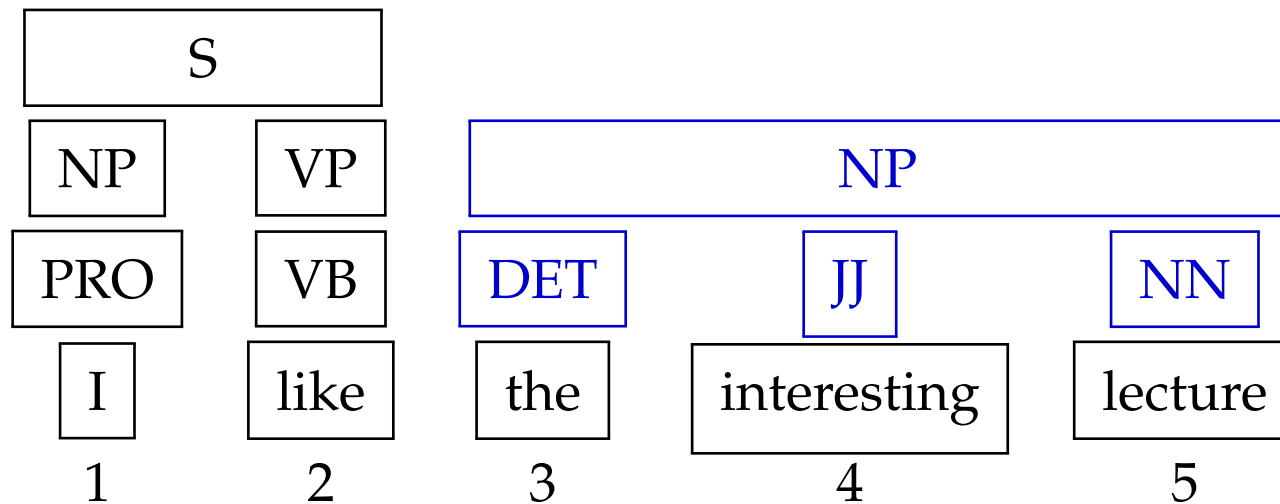
The only matching rule is $S \rightarrow NP VP$

No other rules match for **spans** of two words



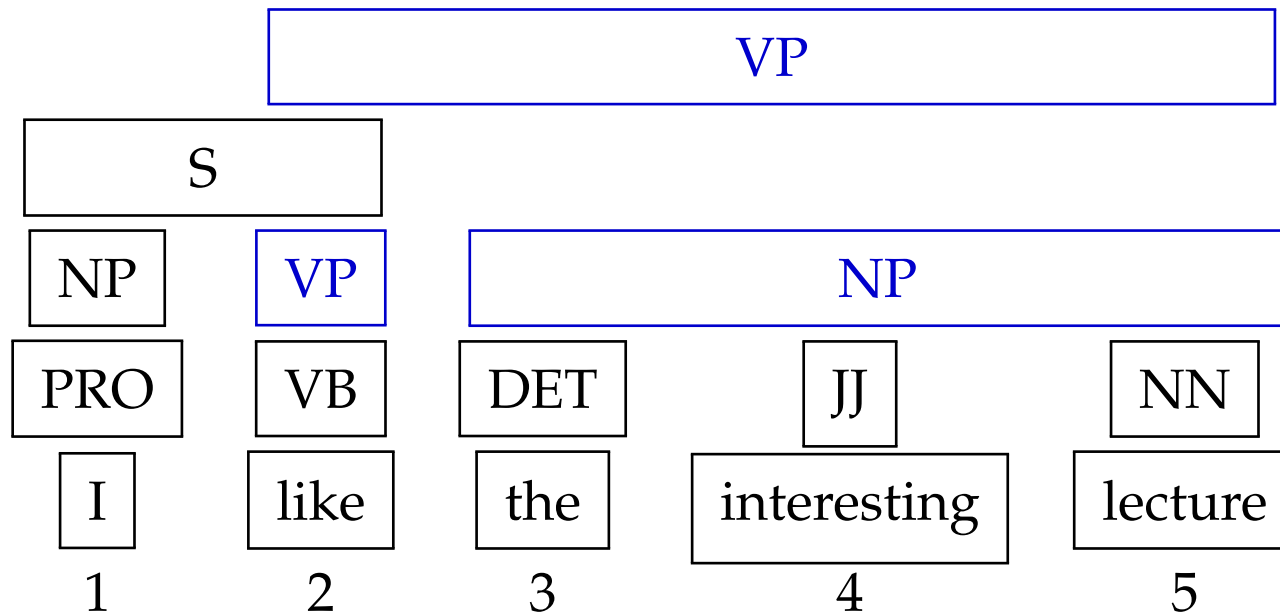
Example

One rule matches for a span of three words: $NP \rightarrow DET JJ NN$



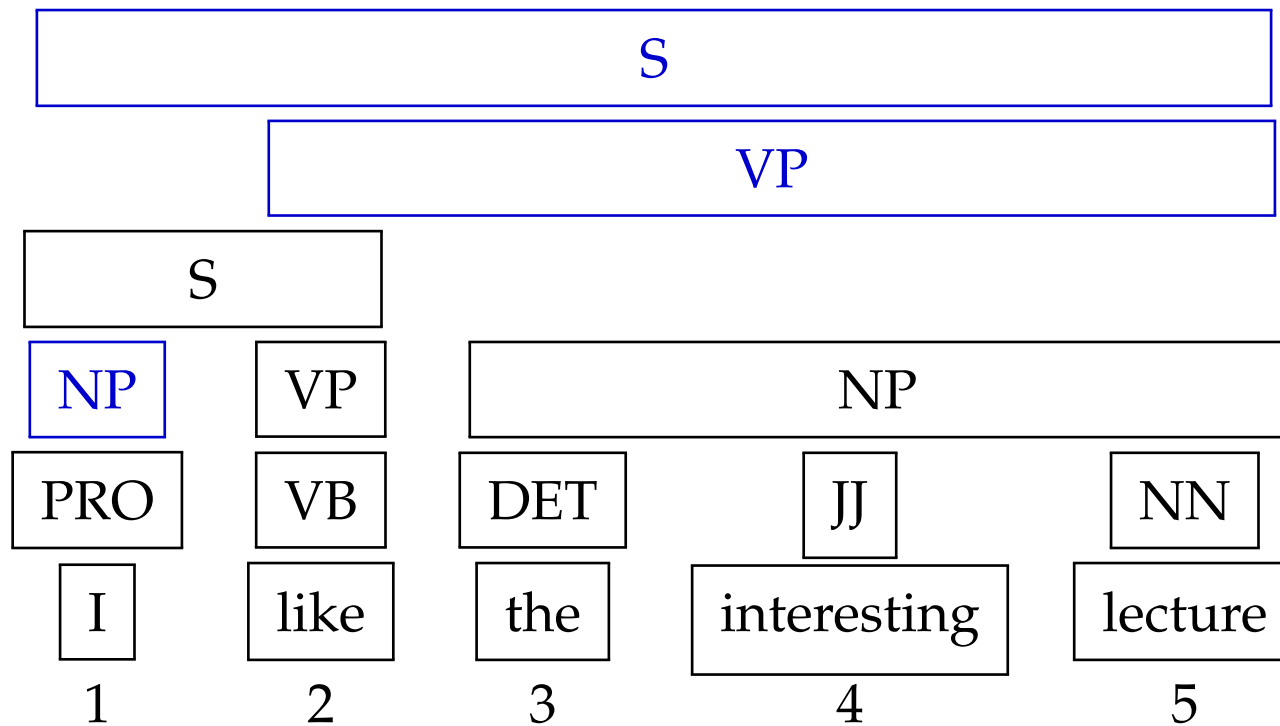
Example

One rule matches for a span of four words: $VP \rightarrow VP NP$



Example

One rule matches for a span of five words: $S \rightarrow NP VP$



Statistical Parsing Models



- Currently best-performing syntactic parsers are statistical
- Assign each rule a probability

$$p(\text{tree}) = \prod_i p(\text{rule}_i)$$

- Probability distributions are learned from manually crafted treebanks

semantics

- Some words have multiple **meanings**
- This is called **Polysemy**
- Example: **bank**
 - financial institution: **I put my money in the bank.**
 - river shore: **He rested at the bank of the river.**
- How could a computer tell these senses apart?

How Many Senses?



- How many senses does the word **interest** have?
 - She pays 3% **interest** on the loan.
 - He showed a lot of **interest** in the painting.
 - Microsoft purchased a controlling **interest** in Google.
 - It is in the national **interest** to invade the Bahamas.
 - I only have your best **interest** in mind.
 - Playing chess is one of my **interests**.
 - Business **interests** lobbied for the legislation.
- Are these seven different senses? Four? Three?

- According to Wordnet, **interest** has 7 senses:
 - Sense 1: a sense of concern with and curiosity about someone or something, Synonym: involvement
 - Sense 2: the power of attracting or holding one's interest (because it is unusual or exciting etc.), Synonym: interestingness
 - Sense 3: a reason for wanting something done, Synonym: sake
 - Sense 4: a fixed charge for borrowing money; usually a percentage of the amount borrowed
 - Sense 5: a diversion that occupies one's time and thoughts (usually pleasantly), Synonyms: pastime, pursuit
 - Sense 6: a right or legal share of something; a financial involvement with something, Synonym: stake
 - Sense 7: (usually plural) a social group whose members control some field of activity and who have common aims, Synonym: interest group

Word Sense Disambiguation (WSD)



- For many applications, we would like to disambiguate senses
 - we may be only interested in one sense
 - searching for **chemical plant** on the web, we do not want to know about chemicals in bananas
- Task: Given a polysemous word, find the sense in a given *context*
- Popular topic, data driven methods perform well

WSD as Supervised Learning Problem

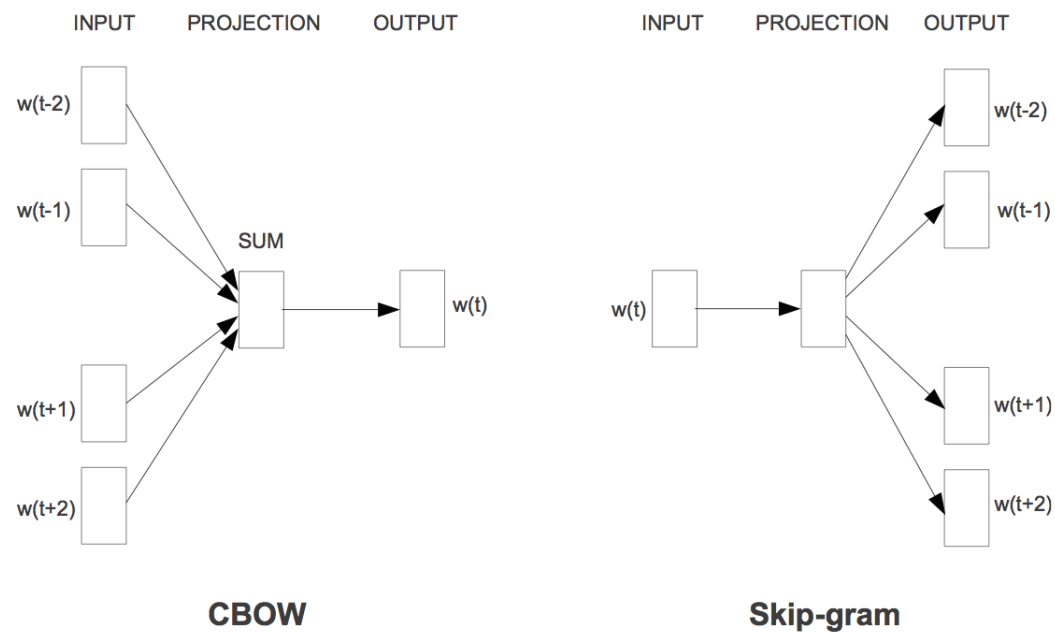


- Words can be labeled with their senses
 - A chemical **plant/PLANT-MANUFACTURING** opened in Baltimore.
 - She took great care and watered the exotic **plant/PLANT-BIOLOGICAL**.
- Features: directly neighboring words
 - **plant** life
 - manufacturing **plant**
 - assembly **plant**
 - **plant** closure
 - **plant** species
- More features
 - any content words in a 50 word window (**animal**, **equipment**, **employee**, ...)
 - syntactically related words, syntactic role in sense
 - topic of the text
 - part-of-speech tag, surrounding part-of-speech tags

Learning Lexical Semantics

The meaning of a word is its use.
Ludwig Wittgenstein, Aphorism 43

- Represent context of a word in a vector
→ Similar words have similar **context vectors**
- Learning with neural networks



Word Embeddings



Word Embeddings



Thematic Roles



- Words play **semantic roles** in a sentence

I see the woman with the telescope .
AGENT THEME INSTRUMENT

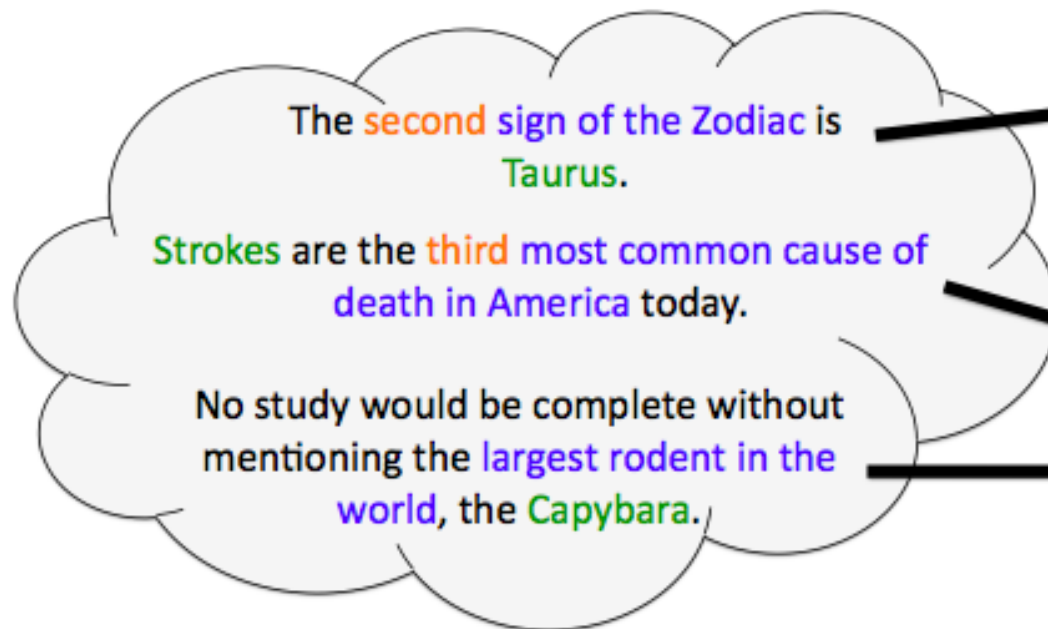
- Specific verbs typically require **arguments** with specific thematic roles and allow **adjuncts** with specific thematic roles.



Unstructured Web Text



Structured Sequences



Sign of the Zodiac:

1. Aries
2. Taurus
3. Gemini...

Most Common Cause of Death in America:

1. Heart Disease
2. Cancer
3. Stroke...

Largest rodent in the world:

1. Capybara
2. Beaver
3. Patagonian Cavies



questions?