Artificial Intelligence: Introduction to Natural Language Processing

Menu

1. Introduction 

2. Bag of word model
3. n-gram models
4. Linguistic features for NLP

Languages

- Artificial
  - Smaller vocabulary
  - Simple syntactic structures
  - Non-ambiguous
  - Not tolerant to errors (ex. Syntax error)
- Natural
  - Large and open vocabulary (new words everyday)
  - Complex syntactic structures
  - Very ambiguous
  - Robust (ex. forgot a comma, a word… still OK)

Question Answering: IBM's Watson

WATSON vs. HUMANS

Round	Watson	Rutter	Jennings
1 (Mon.)	$5000	$5000	$200
2 (Tues.)	$35,734	$10,800	$4,800
3 (Wed.)	$77,147	$21,600	$24,000
Final prize	$1,000,000	$200,000	$300,000

- Won Jeopardy on February 16, 2011!

WILLIAM WILKINSON'S
"AN ACCOUNT OF THE PRINCIPALITIES OF WALLACHIA AND MOLDOVIA"
INSPIRED THIS AUTHOR'S
MOST FAMOUS NOVEL

Who is Bram Stoker?
(Dracula)

Information Extraction

Subject: **curriculum meeting**
Date: January 15, 2012
To: Dan Jurafsky

Hi Dan, we've now scheduled the curriculum meeting.
It will be in Gates 159 tomorrow from 10:00-11:30.
-Chris

Create new Calendar entry

Event:	Curriculum mtg
Date:	Jan-16-2012
Start:	10:00am
End:	11:30am
Where:	Gates 159

Information Extraction & Sentiment Analysis

Attributes:

zoom
affordability
size and weight
flash
ease of use

Size and weight

✓ nice and compact to carry!

✓ since the camera is small and light, I won't need to carry around those heavy, bulky professional cameras either!

✗ the camera feels flimsy, is plastic and very light in weight you have to be very delicate in the handling of this camera

slide from Olga Veksler (U. Western Ontario)

Machine Translation

Fully automatic

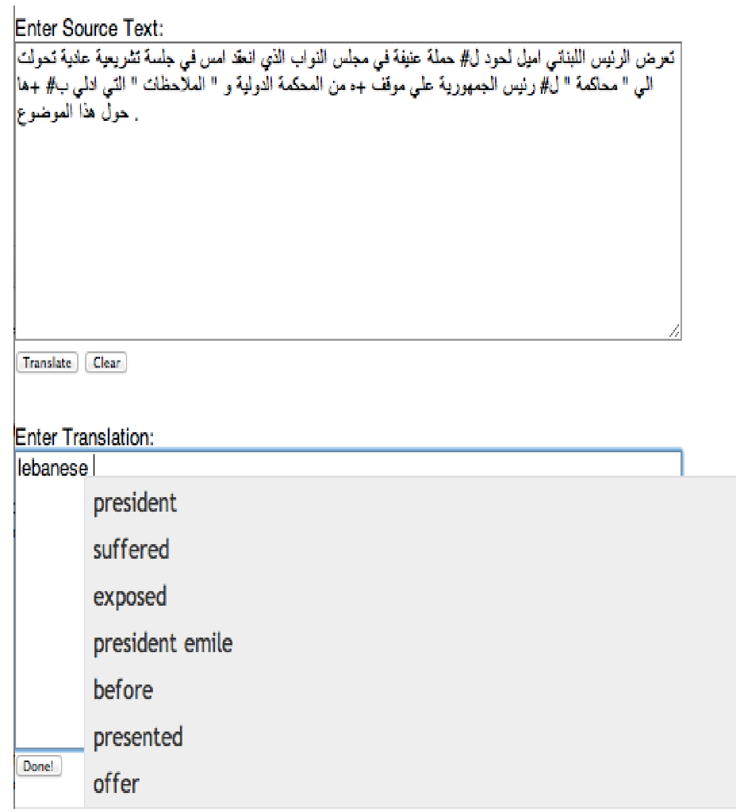
Enter Source Text:

这 不过 是 一 个 时间 的 问题 .

Translation from Stanford's *Phrasal*:

This is only a matter of time.

Helping human translators



Where we are today

mostly solved

Spam detection

Let's go to Agra! ✓

Buy V1AGRA … ✗

Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV

Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC

Einstein met with UN officials in Princeton

making good progress

Sentiment analysis

Best roast chicken in San Francisco!

The waiter ignored us for 20 minutes.

Coreference resolution

Carter told Mubarak he shouldn't run again.

Word sense disambiguation (WSD)

I need new batteries for my *mouse*.

Parsing

I can see Alcatraz from the window!

Machine translation (MT)

第13届上海国际电影节开幕…

The 13th Shanghai International Film Festival…

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30

Party May 27 add

Good progress by Deep Learning

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

The S&P500 jumped

Housing prices rose

Economy is good

Dialog

Where is Citizen Kane playing in SF?

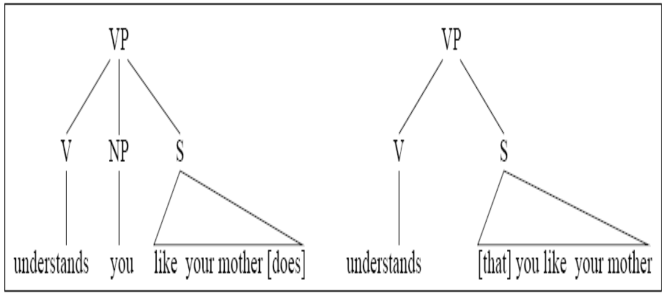
Castro Theatre at 7:30. Do you want a ticket?

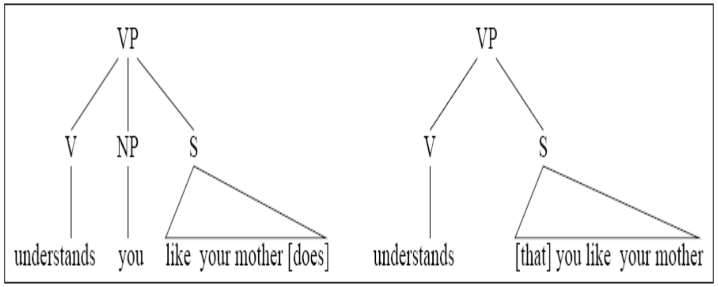
slide from Olga Veksler (U. Western Ontario)

Why is NLP hard?

"At last, a computer that understands you like your mother"

- Because it is ambiguous:
  1. The computer understands you as well as your mother understands you.
  1. The computer understands that you like (love) your mother.
  1. The computer understands you as well as it understands your mother.

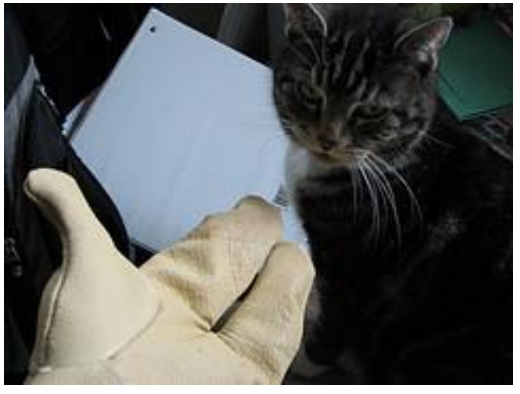




slide from Olga Veksler (U. Western Ontario)

Another Example of Ambiguity

- Even simple sentences are highly ambiguous
- *"Get the cat with the gloves"*



slide from Olga Veksler (U. Western Ontario)

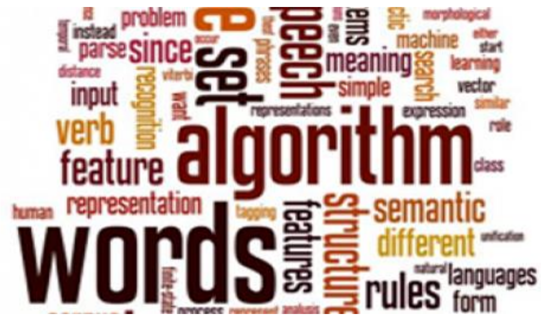
And Even More Examples of Ambiguity

- Iraqi Head Seeks Arms
- Ban on Nude Dancing on Governor's Desk
- Juvenile Court to Try Shooting Defendant
- Teacher Strikes Idle Kids
- Kids Make Nutritious Snacks
- British Left Waffles on Falkland Islands
- Red Tape Holds Up New Bridges
- Bush Wins on Budget, but More Lies Ahead
- Hospitals are Sued by 7 Foot Doctors
- Stolen Painting Found by Tree
- Local HS Dropouts Cut in Half

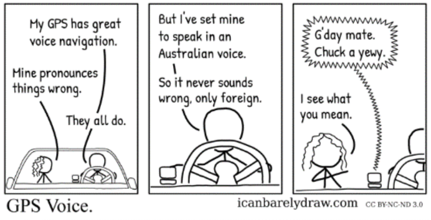
slide from Olga Veksler (U. Western Ontario)

NLP vs Speech Processing

- Natural Language Processing
  = automatic processing of written texts
  1. Natural Language Understanding
     - Input = text
  2. Natural Language Generation
     - Output = text



- Speech Processing
  = automatic processing of speech
  1. Speech Recognition
     - Input = acoustic signal
  2. Speech Synthesis
     - Output = acoustic signal



Remember these slides?

History of AI

- Another big "hype" … Expert Systems (70s - mid 80s)
  - people realized that general-purpose problem solving (weak methods) do not work for practical applications
  - systems need specific domain-dependent knowledge (strong methods)
  - development of knowledge-intensive, rule-based techniques
  - major expert systems
    - MYCIN (1972): expert system to diagnose blood diseases
  - In the industry (1980s): First expert system shells and commercial applications.

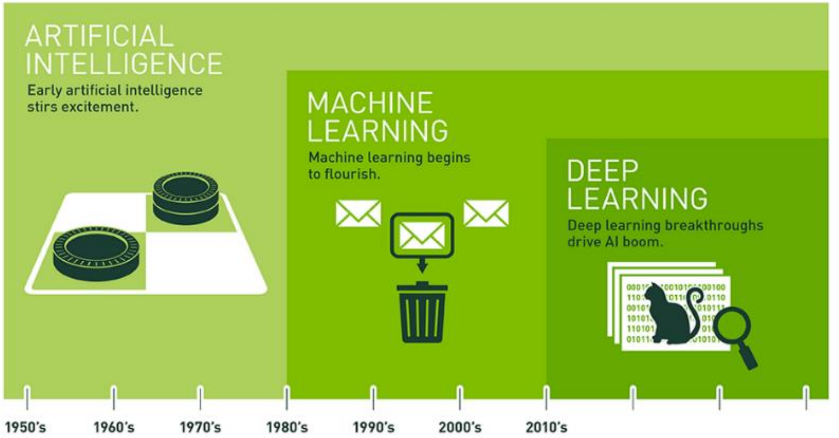
☹ HUMANS need to write the rules by hand…

History of AI

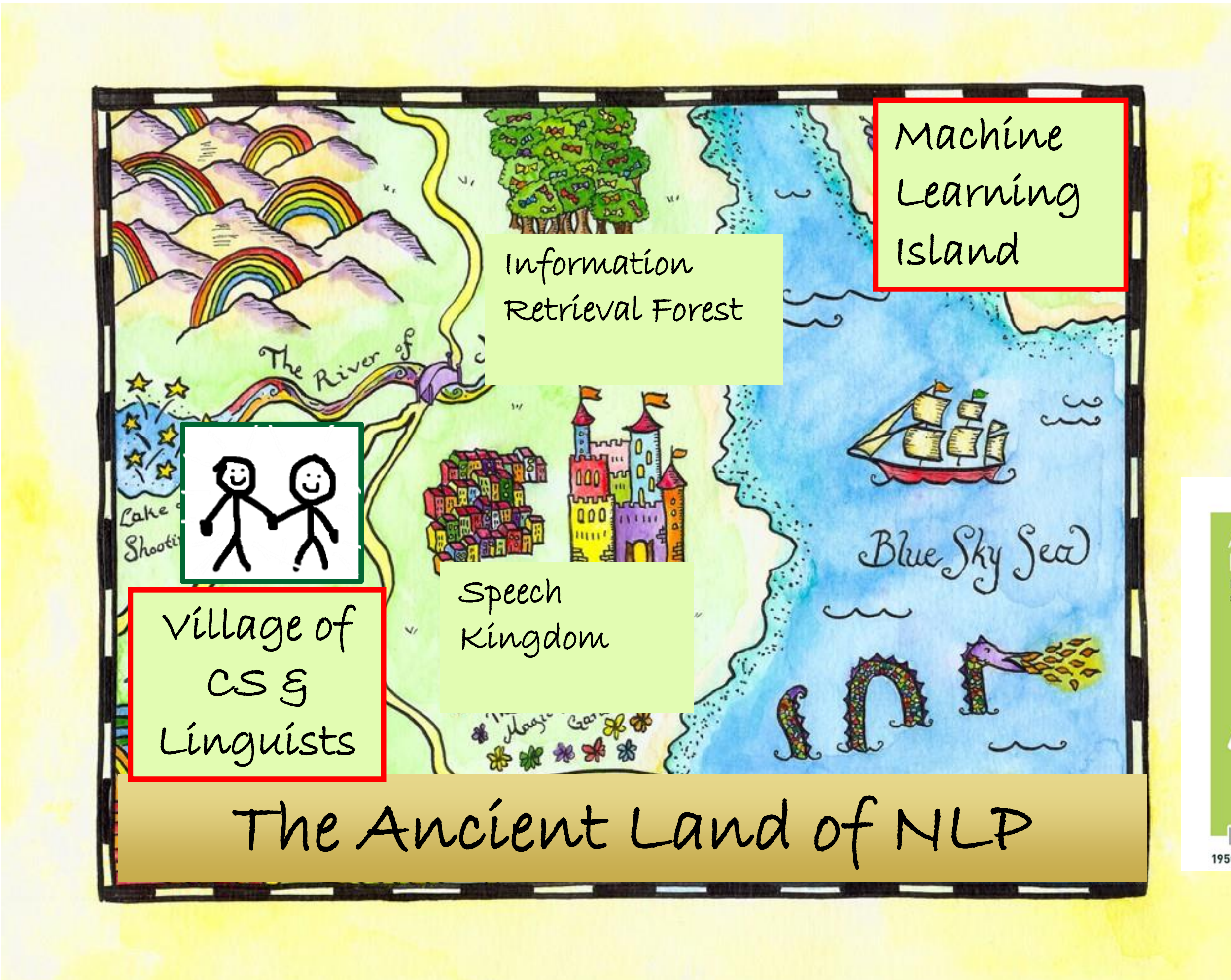
- The rise of Machine Learning (1980s - 2010)
  - More powerful CPUs-> usable implementation of neural networks
  - Big data -> Huge data sets are available to learn from
    - document repositories in NLP, datasets in ML, billions on images for image retrieval, billions of genomic sequences, …
  - ☺ Rules are now learned automatically !
  - AI adopts the Scientific Method

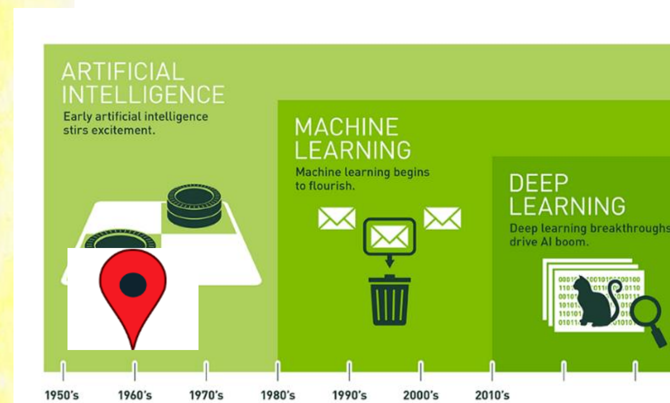
History of AI

- The era of Deep Learning (2010-today)
  - Development of "deep neural networks"
  - Trained on massive data sets
  - Use of GPU for computations
  - Use of "generic networks" for many applications



The Ancient Land of NLP (aka GOFAI) (circa A.D. 1950…mid 1980)



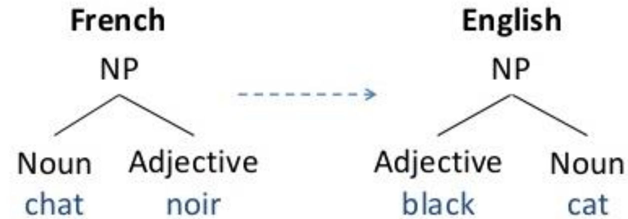


Rule-based NLP
(circa A.D. 1950...mid 1980)

```
s      --> np, vp.
vp     --> v, np.
vp     --> v.
np     --> n.
n      --> [john].     n      --> [lisa].
n      --> [house].
v      --> [died].     v      --> [kissed].

?- s([john, kissed, lisa], []).
   yes
?- s([lisa, died], []).
   yes
?- s([kissed, john, lisa], []).
   no
```

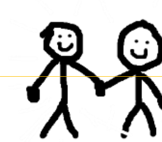
- Rules hand-written by linguists



- State of the art until early 2000's
  - e.g. Systran
- Expensive to create maintain and adapt

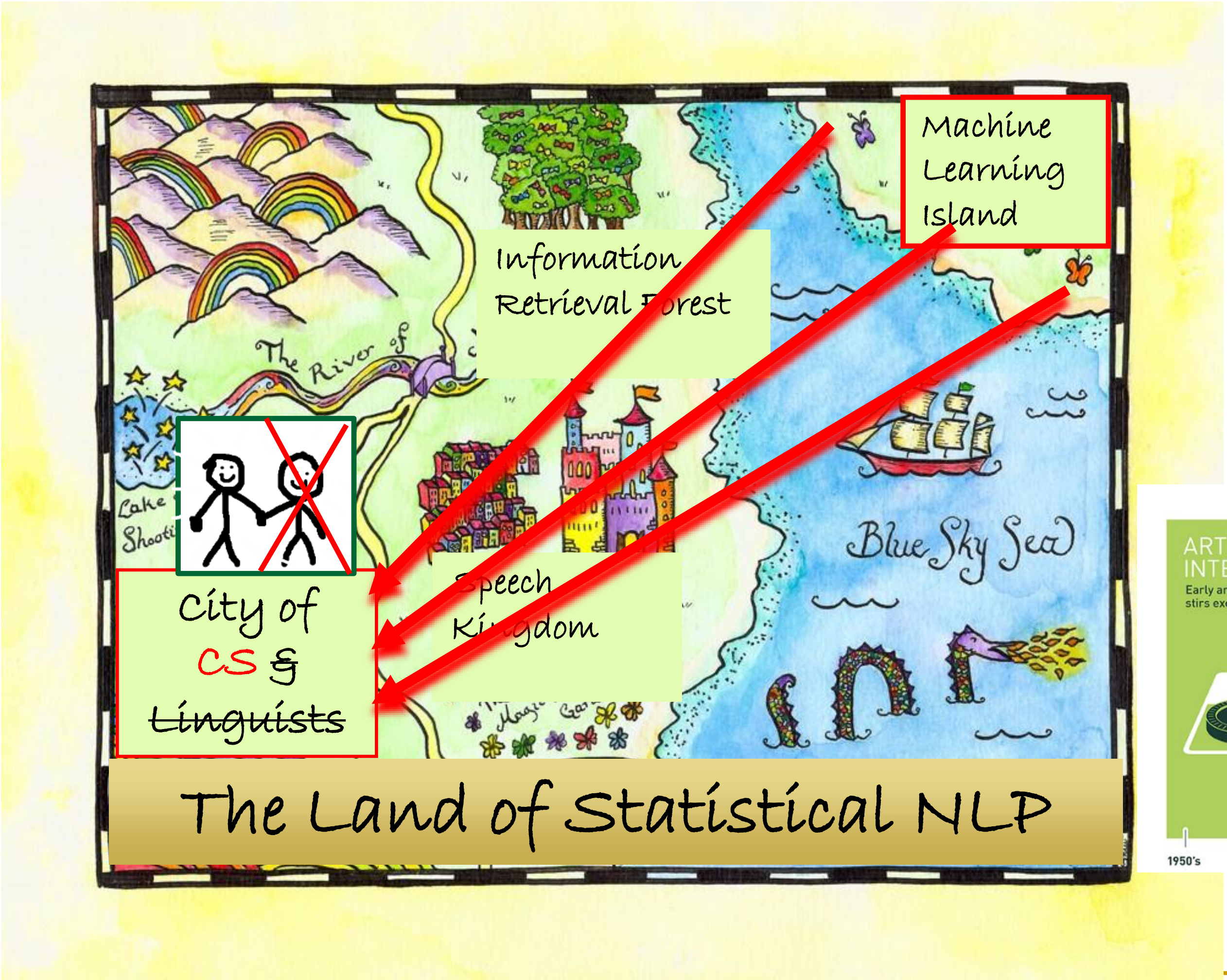
Symbolic methods / Linguistic approach / Knowledge-rich approach

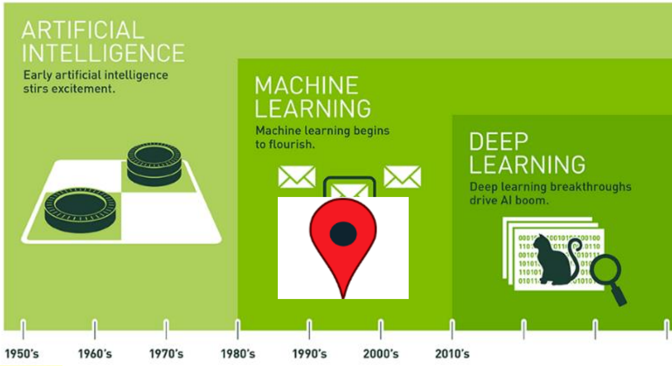
- Cognitive approach
- Rules are developed by hand in collaboration with linguists



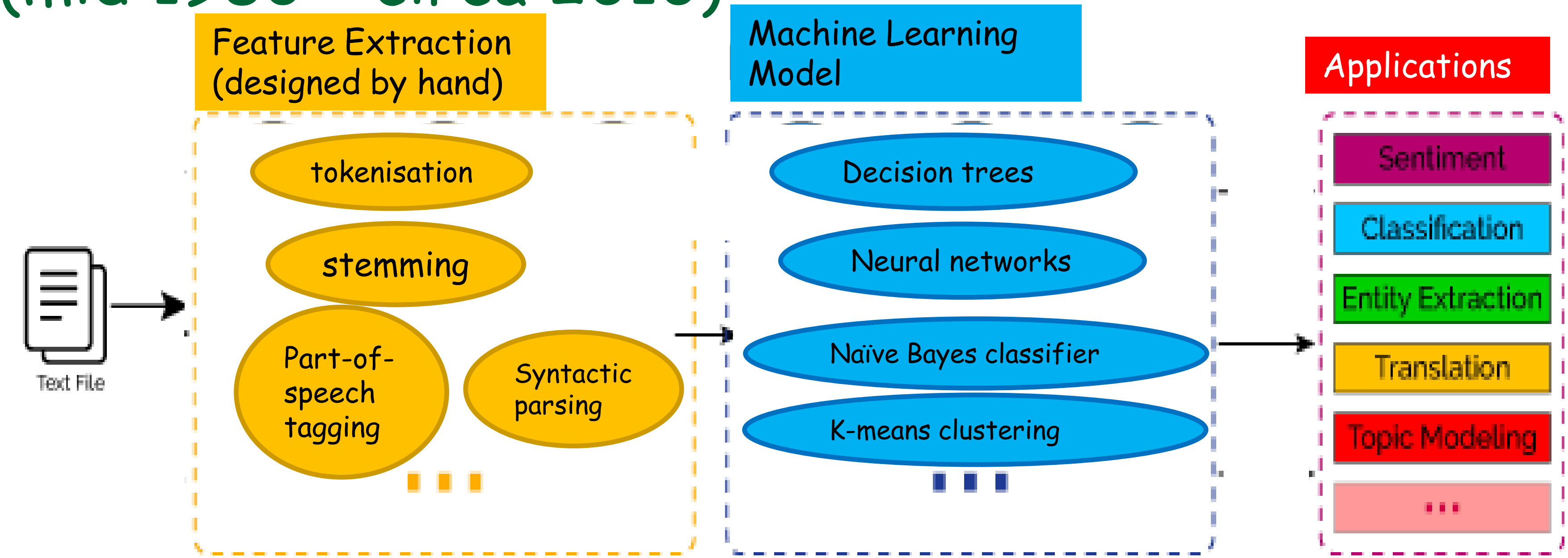
https://image.slidesharecdn.com/maryamsaihbani-161004094341/95/lefttoright-hierarchical-phrasebased-translation-and-its-application-in-simultaneous-speech-translation-maryam-siahbani-4-638.jpg?cb=1477004310

1st Invasion of NLP, from ML
(mid 1980 – circa 2010)





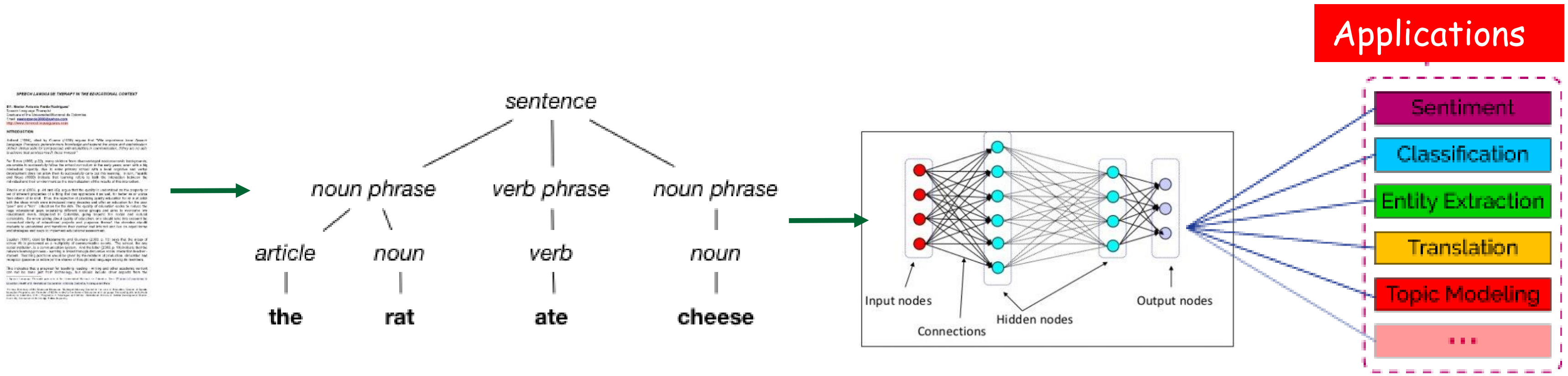
Statistical NLP
(mid 1980 – circa 2010)



Statistical methods / Machine Learning / Knowledge-poor method

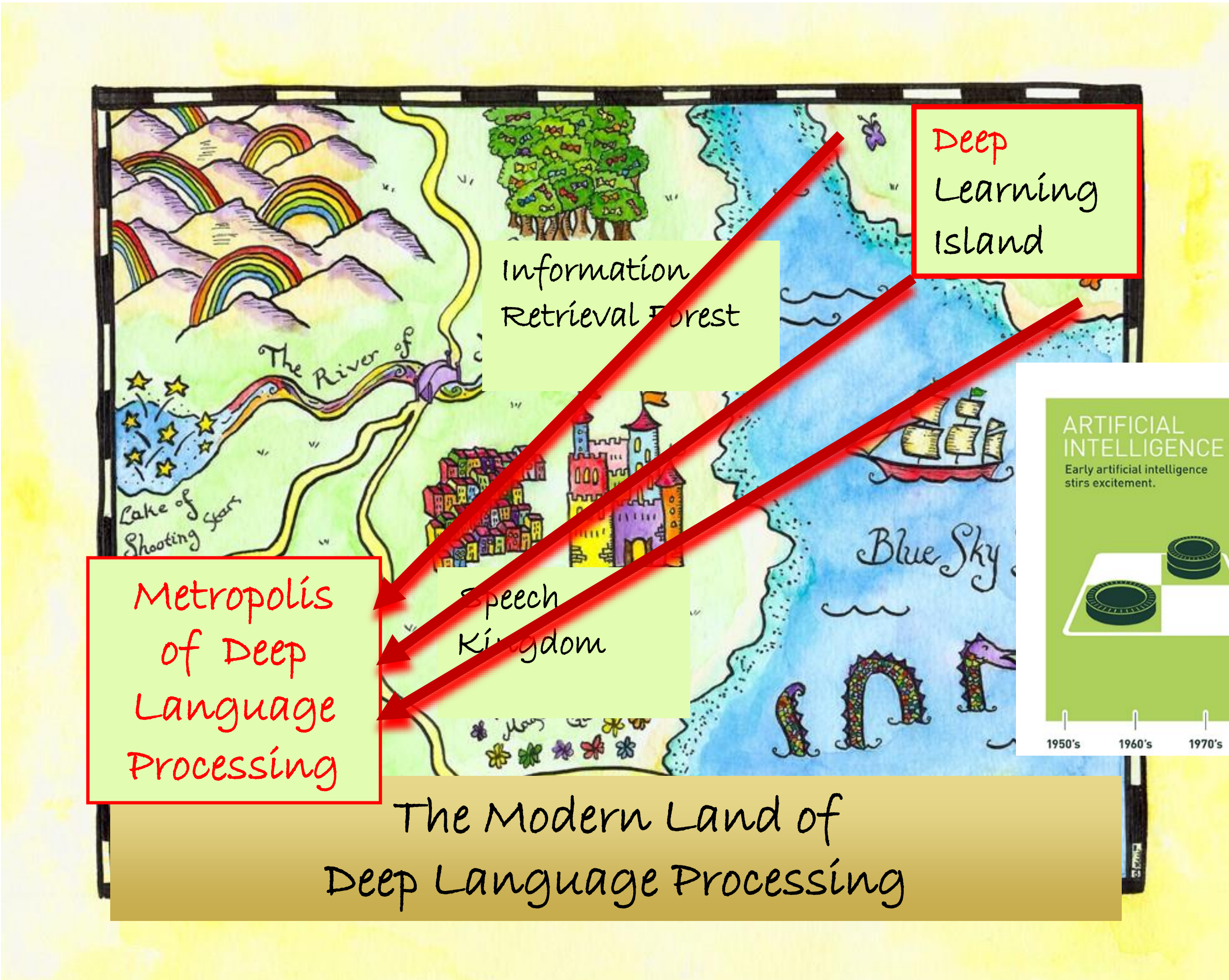
- Engineering Approach
- Rules are developed automatically (using machine learning)
- But the linguistic features are hand-engineered and fed to the ML model
- Applications: Information Retrieval, Predictive Text / Word Completion, Language Identification, Text Classification, Authorship Attribution...

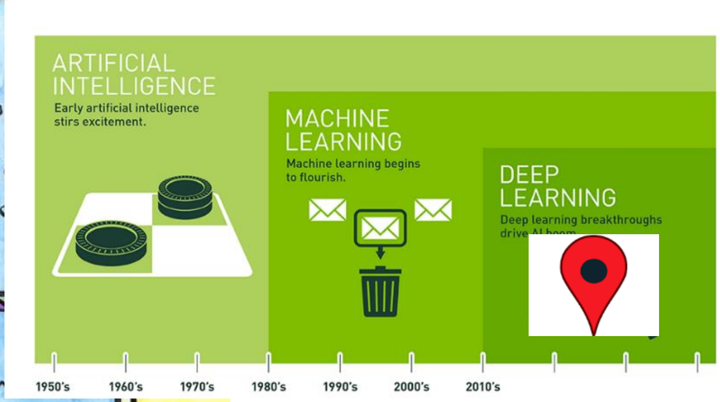
Statistical NLP
(mid 1980 - circa 2010)



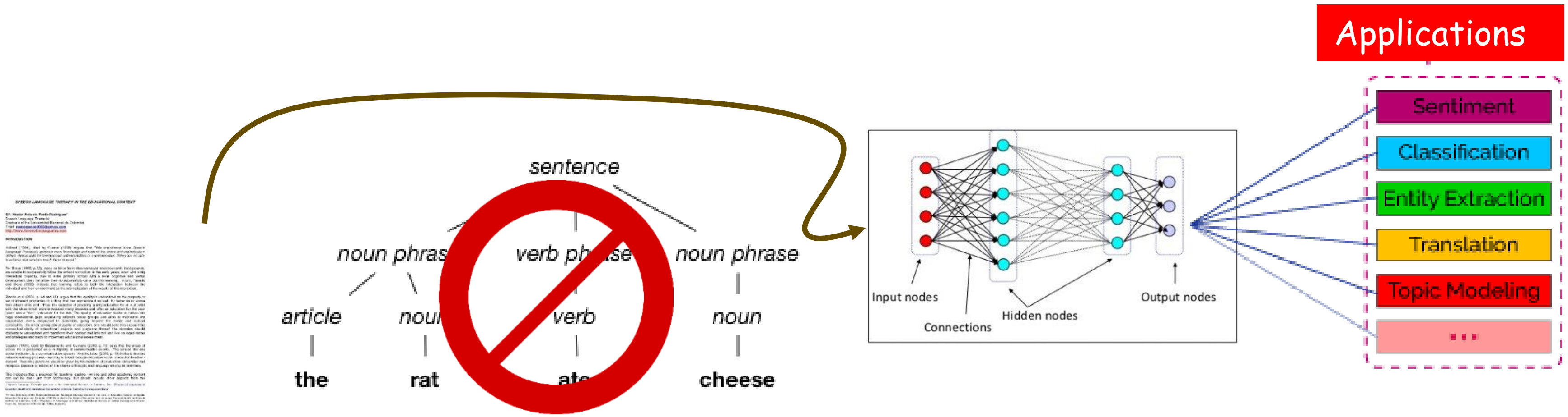
linguistic features are hand-engineered and fed to the ML model

2nd Invasion of NLP, by Deep Learning (circa 2010-today)





Deep Language Processing
(circa 2010-today)



Deep Neural Networks applied to NLP problems

- Rules are developed automatically (using machine learning)
- And the linguistic features are found automatically!

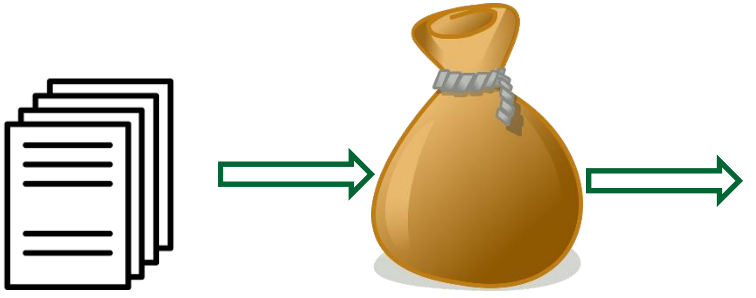
Menu

1. Introduction
2. Bag of word model

3. n-gram models
4. Linguistic features for NLP

Bag-of-word Model (BOW)

- A simple model where word order is ignored



Word	Freq.
Mary	2
apples	1
did	2
eat	1
John	1
kill	1
like	1
not	1
to	1

- used in many applications:
  - NB spam filter seen in class a few weeks ago
  - Information Retrieval (eg. google search)
  - ...
- But has severe limits to understand meaning of text...
- Maybe we should take word order into account...

Limits of BOW Model

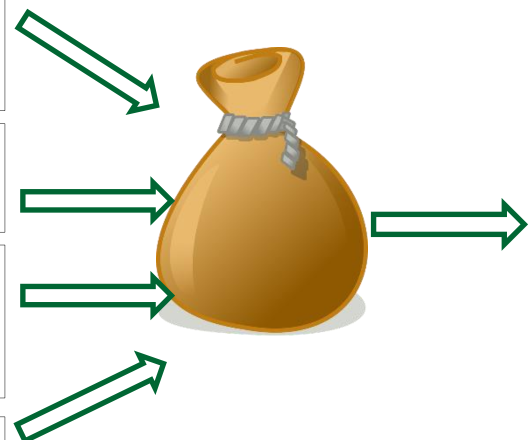
- word order is ignored ==> meaning of text is lost.

Mary did kill John.
Mary did not like to eat apples.

John did not kill Mary.
Mary did like to eat apples.

Mary did not like to kill John.
Mary did eat apples.

...



Word	Freq.
Mary	2
apples	1
did	2
eat	1
John	1
kill	1
like	1
not	1
to	1

- n-grams take [a bit of] word order into account

Menu

1. Introduction
2. Bag of word model
3. n-gram models
4. Linguistic features for NLP

n-gram Model

- An n-gram model is a probability distribution over <u>sequences</u> of events (grams/units/items)
- models the order of the events
- Used when the past sequence of events is a good indicator of the next event to occur in the sequence
- i.e. To predict the next event in a sequence of event
- E.g.:
  - next move of player based on his/her past moves
    - left right right up ... up? down? left? right?
  - next base pair based on past DNA sequence
    - AGCTTCG ... A? G? C? T?
  - next word based on past words
    - Hi dear, how are ... helicopter? laptop? you? magic?

What's a Language Model?

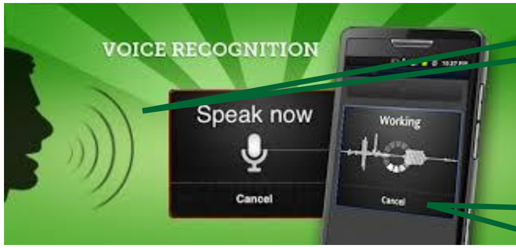
- A Language model is a n-gram model over word/character sequences
- ie: events = words or events = character

- P("I'd like a coffee with 2 sugars and milk") ≈ 0.001
- P("I'd hike a toffee with 2 sugars and silk") ≈ 0.000000001

Applications of Language Models

- Speech Recognition
- Statistical Machine Translation
- Language Identification
- Spelling correction
  - *He is trying to fine out.*
  - *He is trying to find out.*
- Optical character recognition / Handwriting recognition
- …

In Speech Recognition



Given: Observed sound - O

Find: The most likely word/sentence – S*

S1: How to recognize speech. ?

S2: How to wreck a nice beach. ?

S3: …

- Goal: find most likely sentence (S*) given the observed sound (O) …
- ie. pick the sentence with the highest probability: $S^* = \underset{S \in L}{\operatorname{argmax}} P(S \mid O)$
- We can use Bayes rule to rewrite this as: $$S^* = \underset{S \in L}{\operatorname{argmax}} \frac{P(O \mid S) P(S)}{P(O)}$$
- Since denominator is the same for each candidate S, we can ignore it for the argmax:

$$S^* = \underset{S \in L}{\operatorname{argmax}} \; P(O \mid S) \times P(S)$$

Acoustic model --
Probability of the possible phonemes in the language +
Probability of ≠ pronunciations

Language model -- P(a sentence)
Probability of the candidate sentence in the language

In Speech Recognition

$$S^* = \underset{S \in L}{\operatorname{argmax}}\, P(O \mid S) \times P(S)$$

Acoustic model

Language model

$$\underset{\text{word sequence}}{\operatorname{argmax}}\, P(\text{word sequence} \mid \text{acoustic signal})$$

$$\underset{\text{wordsequence}}{\operatorname{argmax}}\, \frac{P(\text{acoustic signal} \mid \text{word sequence}) \times P(\text{word sequence})}{P(\text{acoustic signal})}$$

$$\underset{\text{wordsequence}}{\operatorname{argmax}}\, P(\text{acoustic signal} \mid \text{word sequence}) \times P(\text{word sequence})$$

In Statistical Machine Translation

- Assume we translate from fr[foreign] to English i.e.: (en|fr)

Given: Foreign sentence - fr

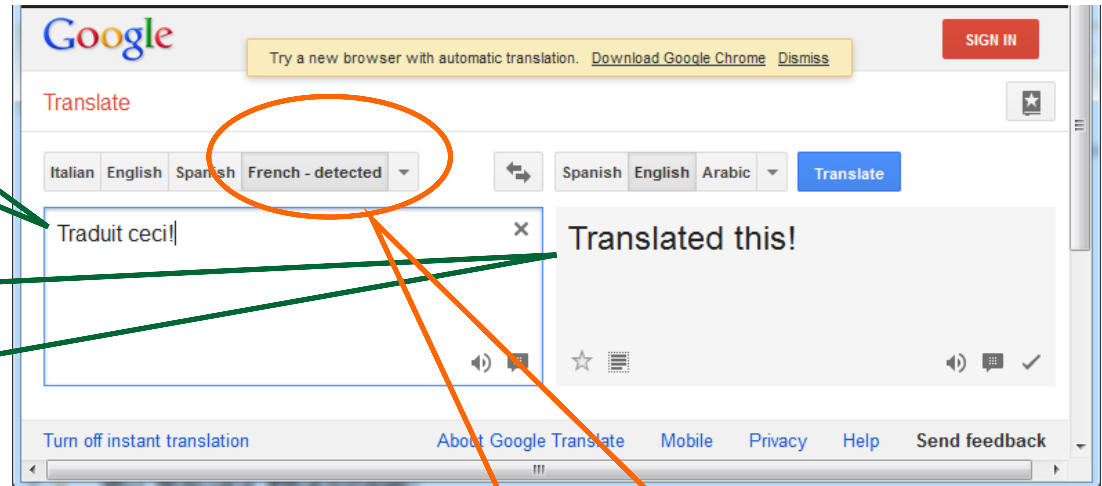
Find: The most likely English sentence – en*

S1: Translate that!

S2: Translated this!

S3: Eat your soup!

S4…



Automatic Language Identification… guess how that's done?

$$en^* = \underset{en}{\operatorname{argmax}} P(fr|en) \times P(en)$$

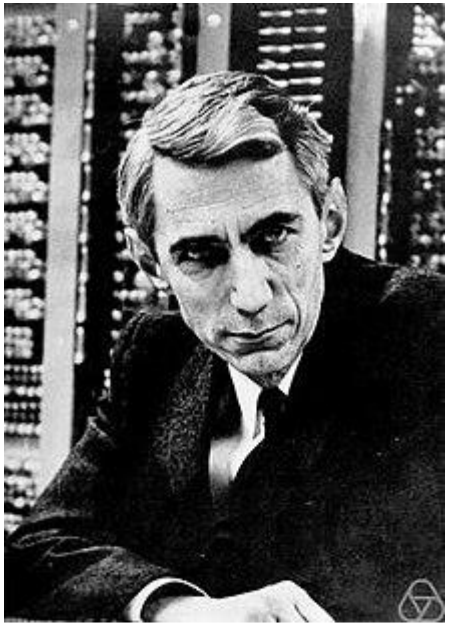
Translation model

Language model

"Shannon Game" (Shannon, 1951)

"I am going to make a collect …"

- Predict the next word/character given the *n-1* previous words/characters.



https://en.wikipedia.org/wiki/Claude_Shannon

1st approximation

- each word has an equal probability to follow any other
  - with 100,000 words, the probability of each word at any given point is .00001
- but some words are more frequent then others…
  - "*the*" appears many more times, than "*rabbit*"

2nd approximation: unigrams

- take into account the frequency of the word in some training corpus
  - at any given point, "*the*" is more probable than "*rabbit*"
- but does not take word order into account. This is the bag of word approach.
  - *"Just then, the white …"*
- so the probability of a word also depends on the previous words (the history)

$$P(w_n | w_1 w_2 \ldots w_{n-1})$$

n-grams

- *"the large green ______ ."*
  - *"mountain"*? *"tree"*?
- *"Sue swallowed the large green ______ ."*
  - *"pill"*? *"broccoli"*?

- Knowing that Sue "*swallowed*" helps narrow down possibilities
- i.e., going back 3 words before helps
- But, how far back do we look?

Bigrams

- first-order Markov models

$$P(w_n|w_{n-1})$$

- N-by-N matrix of probabilities/frequencies
- N = size of the vocabulary we are using

2nd word

1st word

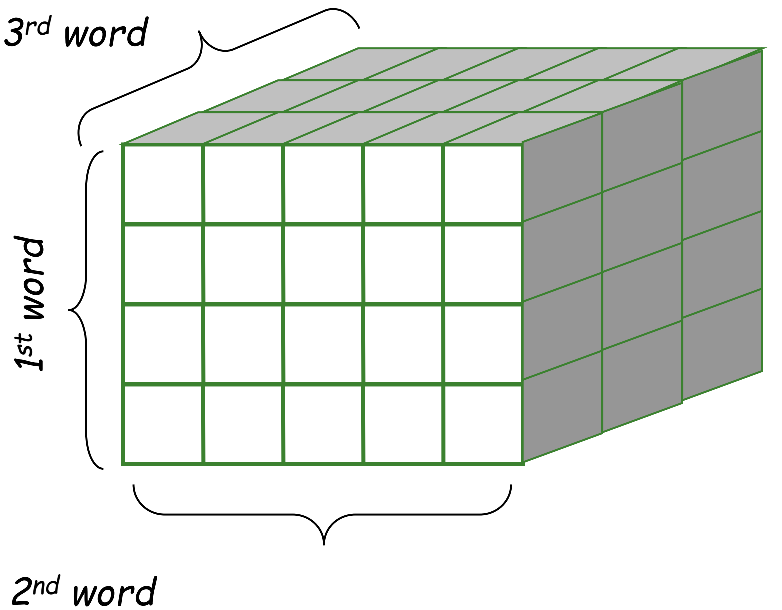
	a	aardvark	aardwolf	aback	…	zoophyte	zucchini
a	0	0	0	0	…	8	5
aardvark	0	0	0	0	…	0	0
aardwolf	0	0	0	0	…	0	0
aback	26	1	6	0	…	12	2
…	…	…	…	…	…	…	…
zoophyte	0	0	0	1	…	0	0
zucchini	0	0	0	3	…	0	0

Trigrams

- second-order Markov models

$$P(w_n | w_{n-1} w_{n-2})$$

- N-by-N-by-N matrix of probabilities/frequencies
- N = size of the vocabulary we are using



Why use only bi- or tri-grams?

- Markov approximation is still costly

  with a 20 000 word vocabulary:
  - bigram needs to store 400 million parameters
  - trigram needs to store 8 trillion parameters
  - using a language model > trigram is impractical

Building n-gram Models

1. Data preparation:
   - Decide on training corpus
   - Clean and tokenize
   - How do we deal with sentence boundaries?
     - I eat. I sleep.
       - (I eat) (eat I) (I sleep)
     - <s>I eat </s> <s> I sleep </s>
       - (<s> I) (I eat) (eat </s>) (<s> I) (I sleep) (sleep </s>)

Example 1:

- in a training corpus, we have 10 instances of "*come across*"
  - 8 times, followed by "*as*"
  - 1 time, followed by "*more*"
  - 1 time, followed by "*a*"

- so we have:
  - $P(\text{as} \mid \text{come across}) = \frac{C(\text{come across as})}{C(\text{come across})} = \frac{8}{10}$
  - P(more | come across) = 0.1
  - P(a | come across) = 0.1
  - P(X | come across) = 0 where X ≠ "*as*", "*more*", "*a*"

Building n-gram Models

2. Count words and build model
   - Let $C(w_1 \dots w_n)$ be the frequency of n-gram $w_1 \dots w_n$

$$P(w_n \mid w_1 \dots w_{n-1}) = \frac{C(w_1 \dots w_n)}{C(w_1 \dots w_{n-1})}$$

3. Smooth your model (see later)

Example 2:

$P(\text{on} \mid \text{eat}) =$ .16	$P(\text{want} \mid \text{I}) =$ .32	$P(\text{eat} \mid \text{to}) =$ .26
$P(\text{some} \mid \text{eat}) =$ .06	$P(\text{would} \mid \text{I}) =$ .29	$P(\text{have} \mid \text{to}) =$ .14
$P(\text{British} \mid \text{eat}) =$ .001	$P(\text{don't} \mid \text{I}) =$ .08	$P(\text{spend} \mid \text{to}) =$ .09
…	…	…
$P(\text{I} \mid \langle s \rangle) =$ .25	$P(\text{to} \mid \text{want}) =$ .65	$P(\text{food} \mid \text{British}) =$ .6
$P(\text{I'd} \mid \langle s \rangle) =$ .06	$P(\text{a} \mid \text{want}) =$ .5	$P(\text{restaurant} \mid \text{British}) =$ .15
…	…	…

P(*I want to eat British food*)

$= P(\text{I} \mid \langle s \rangle) \times P(\text{want} \mid \text{I}) \times P(\text{to} \mid \text{want}) \times P(\text{eat} \mid \text{to}) \times P(\text{British} \mid \text{eat}) \times P(\text{food} \mid \text{British})$

$= .25 \times .32 \times .65 \times .26 \times .001 \times .6$

$= .000008$

Remember this slide...

Be Careful: Use Logs

- if we really do the product of probabilities...
  - $\text{argmax}_{cj}\ P(c_j) \prod P(w_i|c_j)$
  - we soon have numerical underflow...
  - ex: $0.01 \times 0.02 \times 0.05 \times \ldots$
- so instead, we add the log of the probs
  - $\text{argmax}_{cj}\ \log(P(c_j)) + \sum \log(P(w_i|c))$
  - ex: $\log(0.01) + \log(0.02) + \log(0.05) + \ldots$

45

Some Adjustments

- product of probabilities… numerical underflow for long sentences
- so instead of multiplying the probs, we add the log of the probs

P(*I want to eat British food*)
= log(P(I|<s>)) + log(P(want|I)) + log(P(to|want)) + log(P(eat|to)) + log(P(British|eat)) + log(P(food|British))
= log(.25) + log(.32) + log(.65) + log (.26) + log(.001) + log(.6)

Problem: Data Sparseness

- What if a sequence never appears in training corpus? P(X)=0
  - *"come across the men" --> prob* = 0
  - *"come across some men" --> prob* = 0
  - *"come across 3 men" --> prob* = 0

- The model assigns a probability of zero to unseen events …
- probability of an n-gram involving unseen words will be zero!

- Solution: smoothing
  - decrease the probability of previously seen events
  - so that there is a little bit of probability mass left over for previously unseen events

Remember this other slide...

Be Careful: Smooth Probabilities

- normally: $P(w_i \mid c_j) = \dfrac{(\text{frequency of } w_i \text{ in } c_j)}{\text{total number of words in } c_j}$
- what if we have a $P(w_i|c_j) = 0$...?
  - ex. the word "dumbo" never appeared in the class SPAM?
  - then P("dumbo"| SPAM) = 0
- so if a text contains the word "dumbo", the class SPAM is completely ruled out !
- to solve this: we assume that every word always appears at least once (or a smaller value)
  - ex: add-1 smoothing:

$$P(w_i \mid c_j) = \frac{(\text{frequency of } w_i \text{ in } c_j) + 1}{\text{total number of words in } c_j + \text{size of vocabulary}}$$

44

Add-one Smoothing

- Pretend we have seen every n-gram at least once
- Intuitively:
  - new_count(n-gram) = old_count(n-gram) + 1

- The idea is to give a little bit of the probability space to unseen events

Add-one: Example

unsmoothed bigram counts (frequencies):

2nd word

1st word

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total**
I	8	1087	0	13	0	0	0		C(I)=3437
want	3	0	786	0	6	8	6		C(want)=1215
to	3	0	10	860	3	0	12		C(to)=3256
eat	0	0	2	0	19	2	52		C(eat)=938
Chinese	2	0	0	0	0	120	1		C(Chinese)=213
food	19	0	17	0	0	0	0		C(food)=1506
lunch	4	0	0	0	0	1	0		C(lunch)=459
…									…
									…
									N=10,000

- Assume a vocabulary of 1616 (different) words
  - V = {*a, aardvark, aardwolf, aback, … , I, …, want,… to, …, eat, Chinese, …, food, …, lunch, …, zoophyte, zucchini*}
  - |V| = 1616 words
- And a total of N = 10,000 bigrams (~word instances) in the training corpus

Add-one: Example

unsmoothed bigram counts:

2nd word

1st word

	I	want	to	eat	Chinese	food	lunch	…	**Total**
I	8	1087	0	13	0	0	0		C(I)=3437
want	3	0	786	0	6	8	6		C(want)=1215
to	3	0	10	860	3	0	12		C(to)=3256
eat	0	0	2	0	19	2	52		C(eat)=938
Chinese	2	0	0	0	0	120	1		C(Chinese)=213
food	19	0	17	0	0	0	0		C(food)=1506
lunch	4	0	0	0	0	1	0		C(lunch)=459
…									
									N=10,000

unsmoothed bigram conditional probabilities:

	I	want	to	eat	Chinese	food	lunch	…	**Total**
I	0.002 (8/3437)	0.316	0	0.0037	0	0	0		
want	0.0025	0	0.647 (786/1215)	0	0.0049 (6/1215)	0.0066	0.0049		
to	0.0009	0	0.0030	0.264	0.0009	0	0.0037		
eat	0	0	0.002	0	0.020	0.002	0.0554		
Chinese	0.0094	0	0	0	0	0.56	0.0047		
food	0.0126	0	0.0113	0	0	0	0		
lunch	0.0087	0	0	0	0	0.0002	0		
…									

note :

$$P(II) = \frac{8}{10\,000}$$

$$P(I \mid I) = \frac{8}{3\,437}$$

Add-one, more formally

$$P_{Add1}(w_1\, w_2 \ldots w_n) = \frac{C\,(w_1\, w_2 \ldots w_n) + 1}{N + B}$$

N: size of the corpus

i.e. nb of n-gram tokens in training corpus

B: number of "bins"

i.e. nb of <u>different</u> n-gram types

i.e. nb of cells in the matrix

e.g. for bigrams, it's (size of the vocabulary)^2

Add-one: Example (con't)

add-one smoothed bigram counts:

	I	want	to	eat	Chinese	food	lunch	…	**Total**
I	~~8~~ 9	~~1087~~ 1088	1	14	1	1	1		~~3437~~ C(I) + |V| = 5053
want	~~3~~ 4	1	787	1	7	9	7		C(want) + |V| = 2831
to	4	1	**11**	**861**	**4**	**1**	**13**		C(to) + |V| = 4872
eat	1	1	23	1	20	3	53		C(eat) + |V| = 2554
Chinese	3	1	1	1	1	121	2		C(Chinese) + |V| = 1829
food	20	1	18	1	1	1	1		C(food) + |V| = 3122
lunch	5	1	1	1	1	2	1		C(lunch) + |V| = 2075
…									~~total = 10,000~~ $N+\|V\|^2 = 10{,}000 + (1616)^2 = 2{,}621{,}456$

add-one bigram conditional probabilities:

	I	want	to	eat	Chinese	food	lunch	…
I	.0018 (9/5053)	.215	.00019	.0028	.00019	.00019	.00019	
want	.0014	.00035	.278	.00035	.0025	.0031	.00247	
to	.00082	.0002	.00226	.1767	.00082	.0002	.00267	
eat	.00039	.00039	.0009	.00039	.0078	.0012	.0208	
…								

Add-delta Smoothing

- every previously unseen n-gram is given a low probability
- but there are so many of them that too much probability mass is given to unseen events
- instead of adding 1, add some other (smaller) positive value δ

$$P_{AddD}(w_1\, w_2 \ldots w_n) = \frac{C\,(w_1\, w_2 \ldots w_n) + \delta}{N + \delta\, B}$$

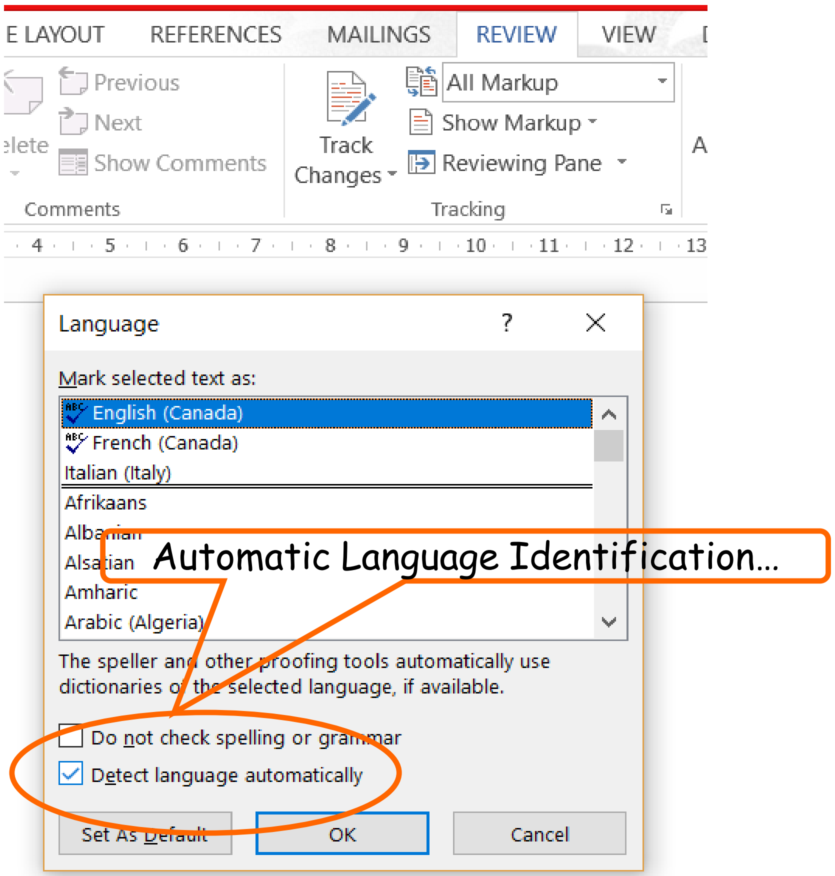
- most widely used value for δ = 0.5
- better than add-one, but still…

Factors of Training Corpus

- Size:
  - the more, the better
  - but after a while, not much improvement…
    - bigrams (characters) after 100's million words
    - trigrams (characters) after some billions of words
- Genre (adaptation):
  - training on cooking recipes and testing on aircraft maintenance manuals

Example: Language Identification

- hypothesis: texts that resemble each other (same author, same language) share similar character/word sequences
  - In English character sequence "*ing*" is more probable than in French
- Training phase:
  - construction of the language model
  - with pre-classified documents (known language/author)
- Testing phase:
  - apply language model to unknown text



Example: Language Identification

- bigram of characters
  - characters = 26 letters (case insensitive)
  - possible variations: case sensitivity, punctuation, beginning/end of sentence marker, …

Example: Language Identification

1. Train a character-based language model for Italian:

	A	B	C	D	…	Y	Z
A	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
B	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
C	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
D	0.0042	0.0014	0.0014	0.0014	…	0.0014	0.0014
E	0.0097	0.0014	0.0014	0.0014	…	0.0014	0.0014
…	…	…	…	…	…	…	0.0014
Y	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
Z	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014

2. Train a character-based language model for Spanish:

	A	B	C	D	…	Y	Z
A	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
B	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
C	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
D	0.0042	0.0014	0.0014	0.0014	…	0.0014	0.0014
E	0.0097	0.0014	0.0014	0.0014	…	0.0014	0.0014
…	…	…	…	…	…	…	0.0014
Y	0.0014	0.0014	0.0014	0.0014	…	0.0014	0.0014
Z	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014

3. Given a unknown sentence "che bella cosa" is it in Italian or in Spanish?

   P("che bella cosa") with the Italian LM

   P("che bella cosa") with the Spanish LM

4. Highest probability → language of sentence

Google's Web 1T 5-gram model

- 5-grams
- generated from 1 trillion words
- 24 GB compressed
  - Number of tokens: 1,024,908,267,229
  - Number of sentences: 95,119,665,584
  - Number of unigrams: 13,588,391
  - Number of bigrams: 314,843,401
  - Number of trigrams: 977,069,902
  - Number of fourgrams: 1,313,818,354
  - Number of fivegrams: 1,176,470,663
- See discussion: http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html
- See Google Ngram Viewer: http://en.wikipedia.org/wiki/Google_Ngram_Viewer

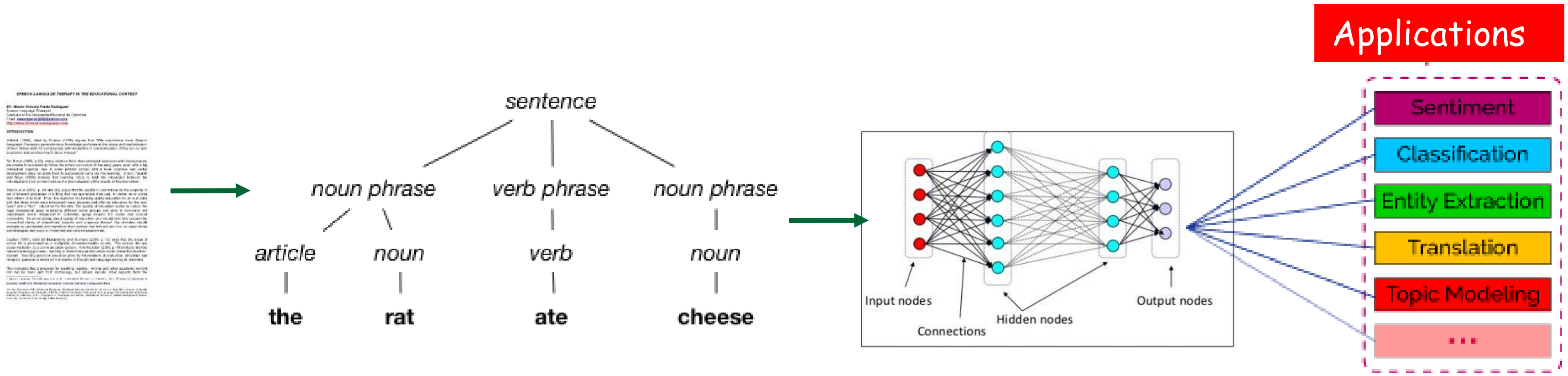
Problem with n-grams

- Natural language is not linear ....
- there may be *long-distance dependencies.*
  - Syntactic dependencies
    - *The <u>man</u> next to the large oak tree near … <u>is</u> tall.*
    - *The <u>men</u> next to the large oak tree near … <u>are</u> tall.*
  - *Semantic dependencies*
    - *The <u>bird</u> next to the large oak tree near … <u>flies</u> rapidly.*
    - *The <u>man</u> next to the large oak tree near … <u>talks</u> rapidly.*
  - *World knowledge*
    - *<u>Michael Jackson</u>, who was featured in ..., is <u>buried</u> in California.*
    - *<u>Michael Bublé</u>, who was featured in ..., is <u>living</u> in California.*
  - ...
- More complex models of language are needed to handle such dependencies.

Menu

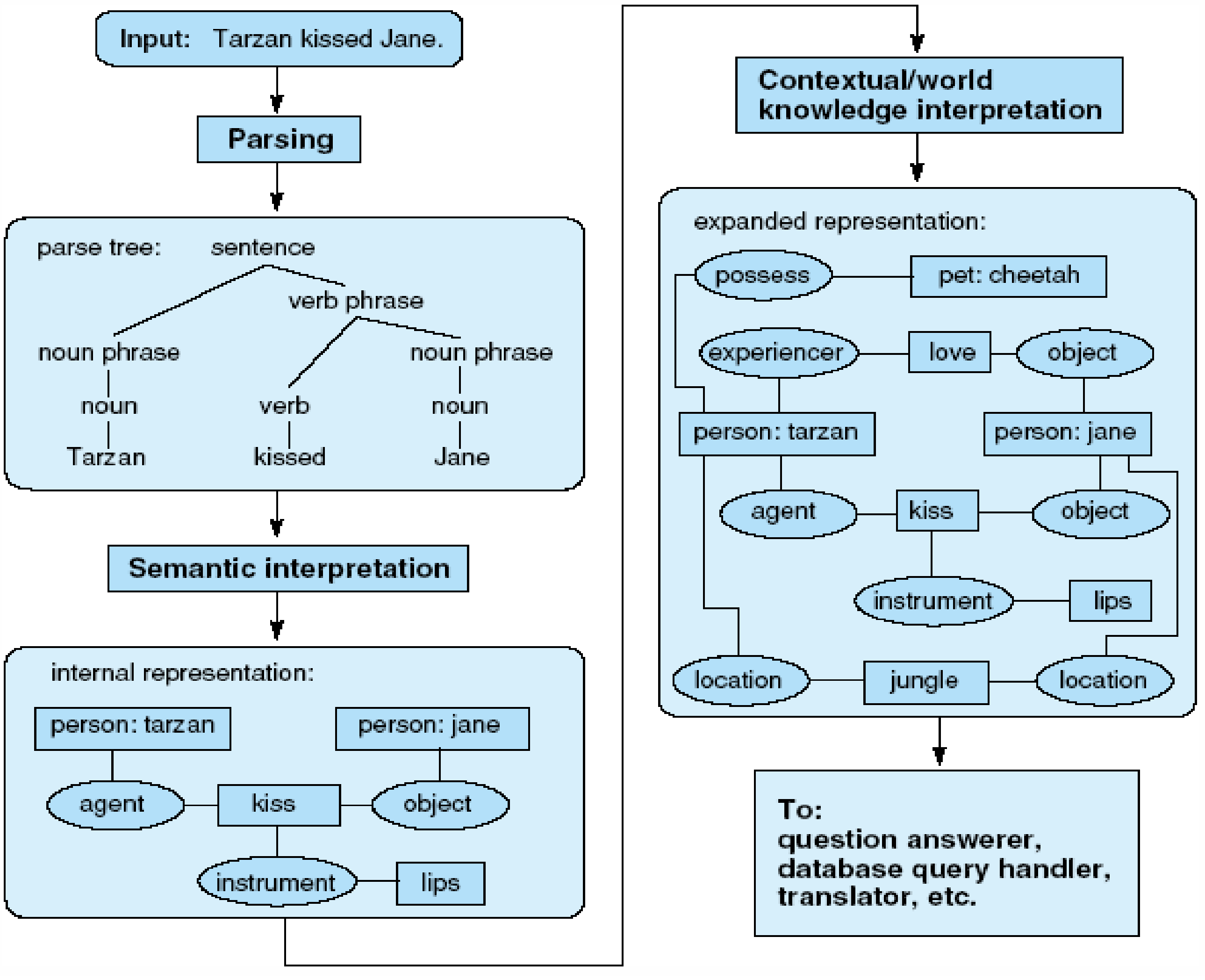
1. Introduction
2. Bag of word model
3. n-gram models
4. Linguistic features for NLP

Linguistic features used for what?



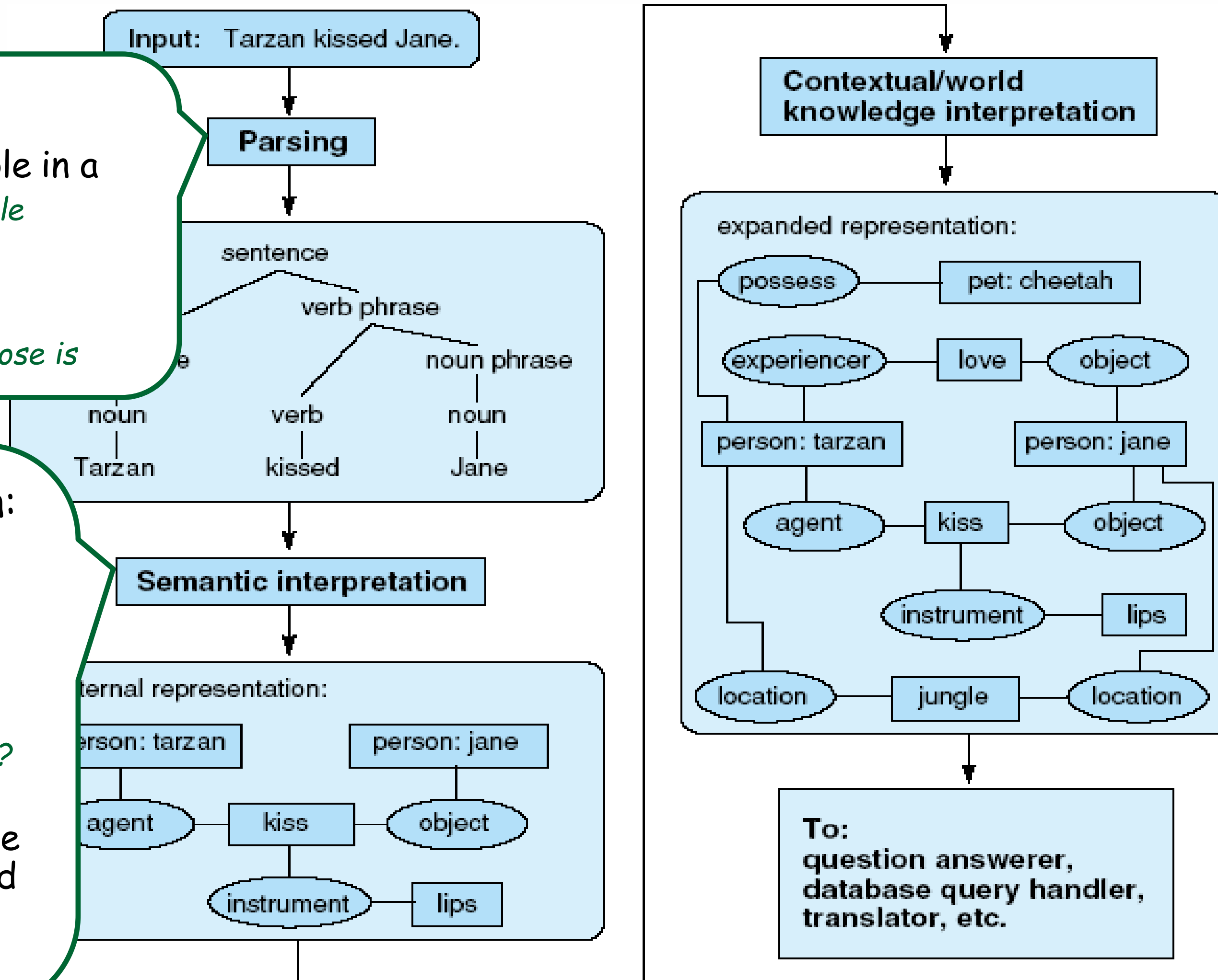
linguistic features are hand-engineered and fed to the ML model

Stages of NLU



source: Luger (2005)

Stages of NLU



Parsing (Syntax):

- What words are available in a language? *gfiioudd / table*
- How to arrange words together?
  the rose is red / red the rose is

Semantic interpretation:

- Lexical Semantics : What is the meaning/semantic relations between individual words?
  Chair: person? Furniture?
- Compositional Semantics: What is the meaning of phrases and sentences?
  The chair's leg is broken

source: Luger (2005)

Stages of NLU

- **Discourse Analysis**

How to relate the meaning of sentences to surrounding sentences?

I have to go to the store. I need butter.

I have to go to the university. I need butter.

- **Pragmatics**

How people use language in a social environment?

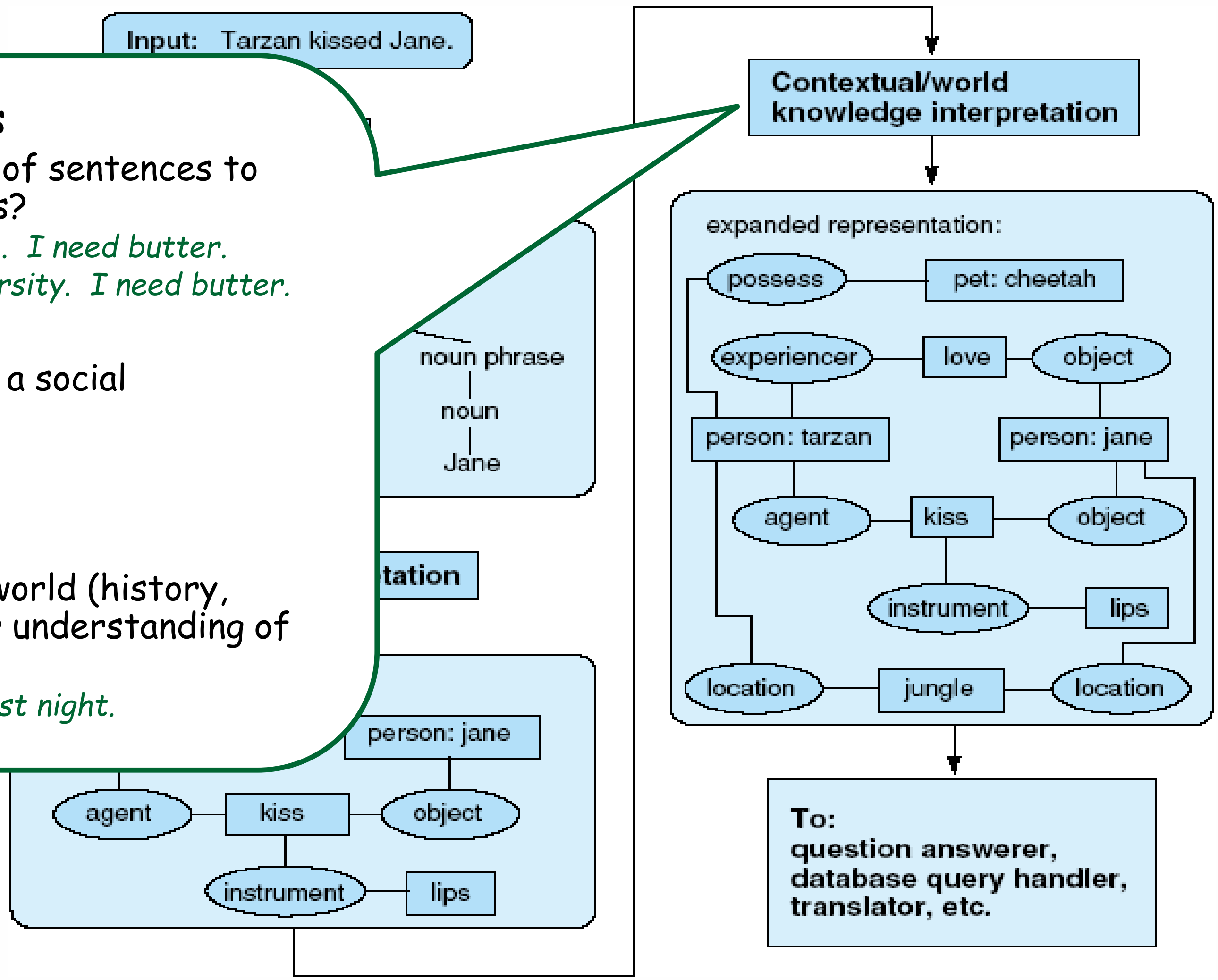
Do you have a child?

Do you have a quarter?

- **World Knowledge**

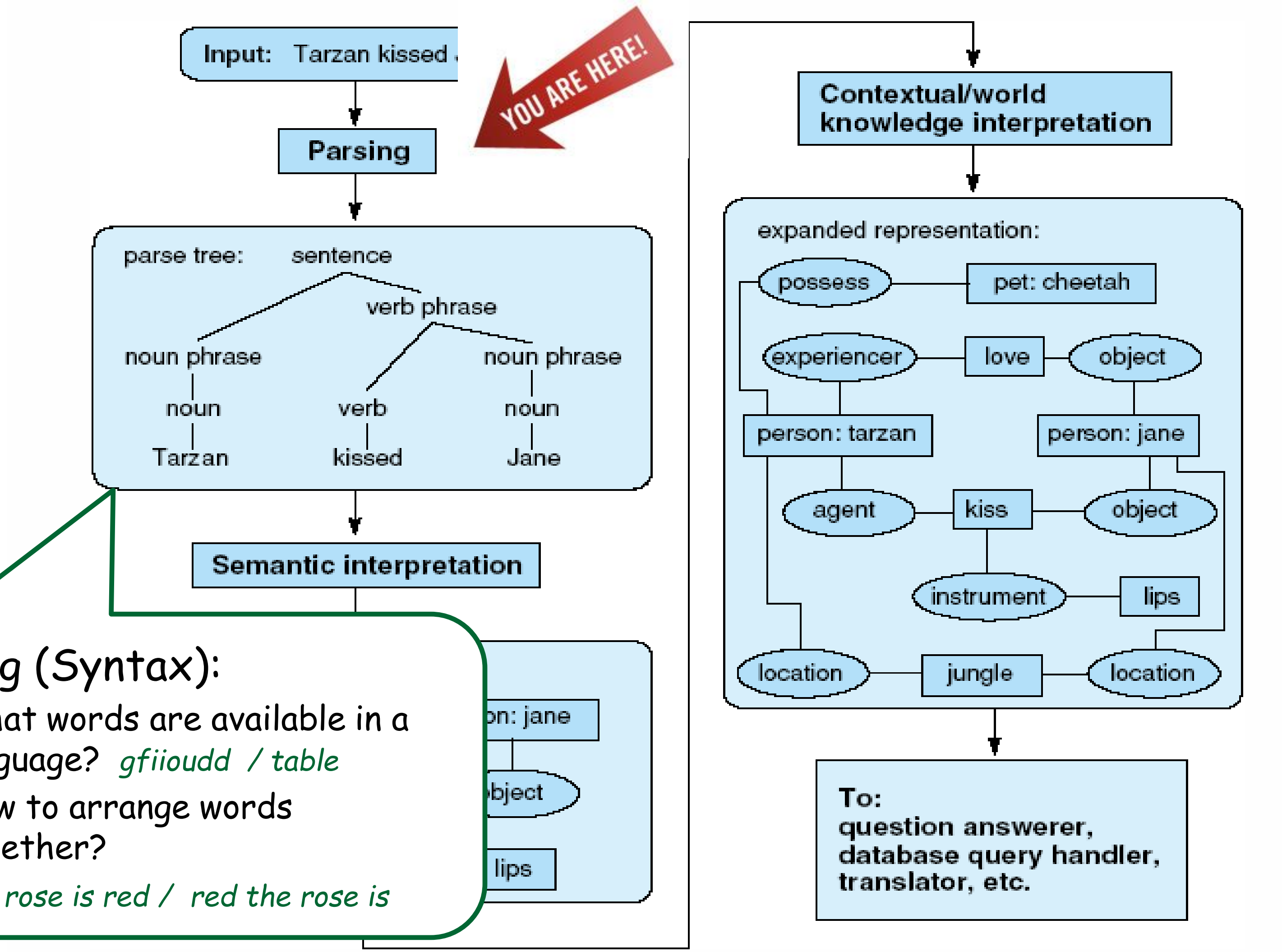
How knowledge about the world (history, facts, …) modifies our understanding of text?

Bill Gates passed away last night.



source: Luger (2005)

Stages of NLU



Parsing (Syntax):

- What words are available in a language? *gfiioudd / table*
- How to arrange words together?

the rose is red / red the rose is

source: Luger (2005)

Syntactic Parsing

1. Assign the right part of speech (NOUN, VERB, …) to individual words in a text

1. Determine how words are put together to form correct sentences
   - The/DET rose/NOUN is/VERB red/ADJ.
   - Is/VERB red/ADJ the/DET rose/NOUN.

English Parts-of-Speech

- Open (lexical) class words
  - new words can be added easily
  - nouns, main verbs, adjectives, adverbs
  - some languages do not have all these categories

Smurf talk on youtube:
https://www.youtube.com/watch?v=7BPx-vl8G0O

- Closed (functional) class words
  - generally function/grammatical words
  - aka *stop words*
  - ex. *the, in, and, over, beyond…*
  - relatively fixed membership
  - prepositions, determiners, pronouns, conjunctions, …

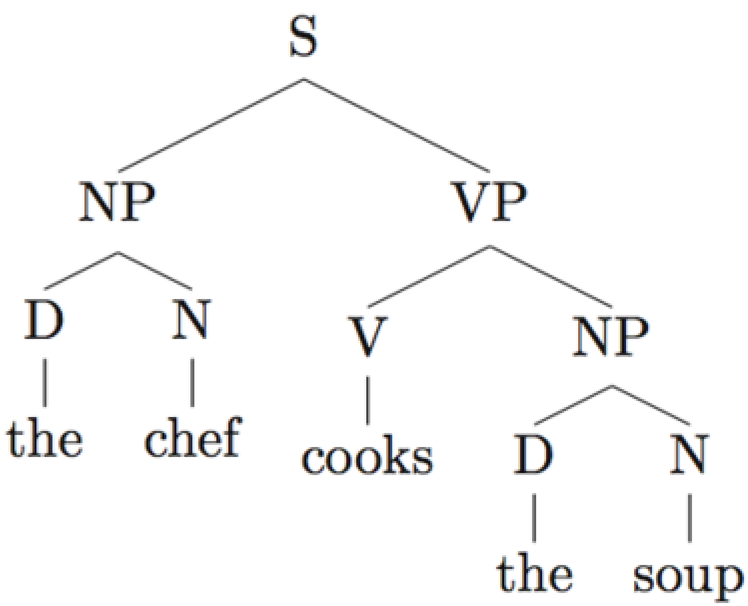
Syntax

- How parts-of-speech are organised into larger syntactic constituents

- Main Constituents:
  - S: sentence *The boy is happy.*
  - NP: noun phrase *the little boy from Paris, Sam Smith, I,*
  - VP: verb phrase *eat an apple, sing, leave Paris in the night*
  - PP: prepositional phrase *in the morning, about my ticket*
  - AdjP: adjective phrase *really funny, rather clear*
  - AdvP: adverb phrase *slowly, really slowly*

A Parse Tree

- a tree representation of the application of the grammar to a specific sentence.



a CFG consists of

- set of non-terminal symbols
  - constituents & parts-of-speech
  - S, NP, VP, PP, D, N, V, ...
- set of terminal symbols
  - words & punctuation
  - *cat, mouse, nurses, eat, ...*
- a non-terminal designated as the starting symbol
  - sentence S
- a set of re-write rules
  - having a single non-terminal on the LHS and one or more terminal or non-terminal in the RHS
  - S --> NP VP
  - NP --> Pro
  - NP --> PN
  - NP --> D N

An Example

- Lexicon:

```
N --> flight | trip | breeze | morning        // noun
V --> is | prefer | like                                  //
   verb
Adj --> direct | cheapest | first                  //
   adjective
Pro --> me | I | you | it                                //
   pronoun
PN --> Chicago | United | Los Angeles        // proper noun
D --> the | a | this                                         //
   determiner
Prep --> from | to | in                                    //
   preposition
Conj --> and | or | but                                    //
   conjunction
```

- Grammar:

```
S --> NP VP                                           // I +
   prefer United
NP --> Pro | PN | D N | D Adj N       // I, Chicago, the morning
VP --> V | V NP | V NP PP                     // is, prefer +
   United,
PP --> Prep NP                                                //
   to Chicago, to I ??
```

Parsing

- parsing:
  - goal:
    - assign syntactic structures to a sentence
  - result:
    - (set of) parse trees
- we need:
  - a grammar:
    - description of the language constructions
  - a parsing strategy:
    - how the syntactic analysis are to be computed

Parsing Strategies

- parsing is seen as a search problem through the space of all possible parse trees
  - bottom-up (data-directed): words --> grammar
  - top-down (goal-directed): grammar --> words

  - breadth-first: compute all paths in parallel
  - depth-first: exhaust 1 path before considering another
  - Heuristic search

Example: *John ate the cat*

- Bottom-up parsing / breadth first

1. John ate the cat
2. PN ate the cat
3. PN V the cat
4. PN V ART cat
5. PN V ART N
6. NP V ART N
7. NP V NP
8. NP VP
9. S

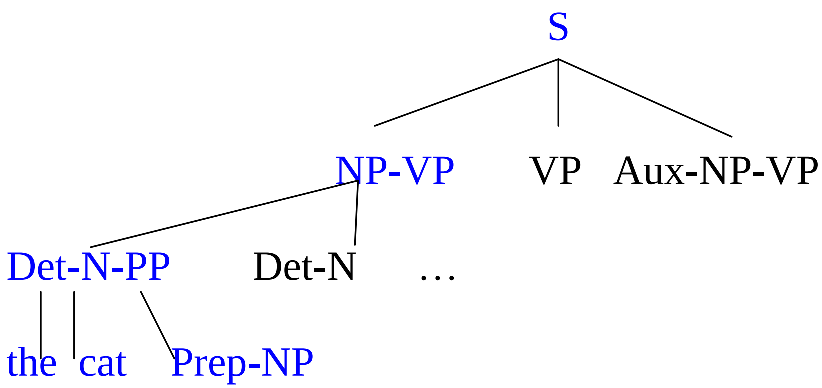
- Top-down parsing / depth first

1. S
2. NP VP
3. PN VP
4. John VP
5. John V NP
6. John ate NP
7. John ate ART N
8. John ate the N
9. John ate the cat

Depth-first vs Breadth-first

the cat eats the mouse.

- depth-first: exhaust 1 path before considering another



- breadth-first:
  - compute 1 level at a time
- Heuristic search:
  - e.g. preference to shorter rules

```
Grammar:
(1) S --> NP VP
(2) S --> VP
(3) S --> Aux NP VP
(4) NP --> Det N PP
(5) NP -- > Det N
(6) PP -- > Prep N
…

Lexicon:
(10) Det --> the
(11) N --> cat
(12) VB --> eats
…
```

Summary of Parsing Strategies

	Depth First	Breath First	Heuristic Search
Top down	✓	✓	✓
Bottom up	✓	✓	✓

Problem: Multiple parses

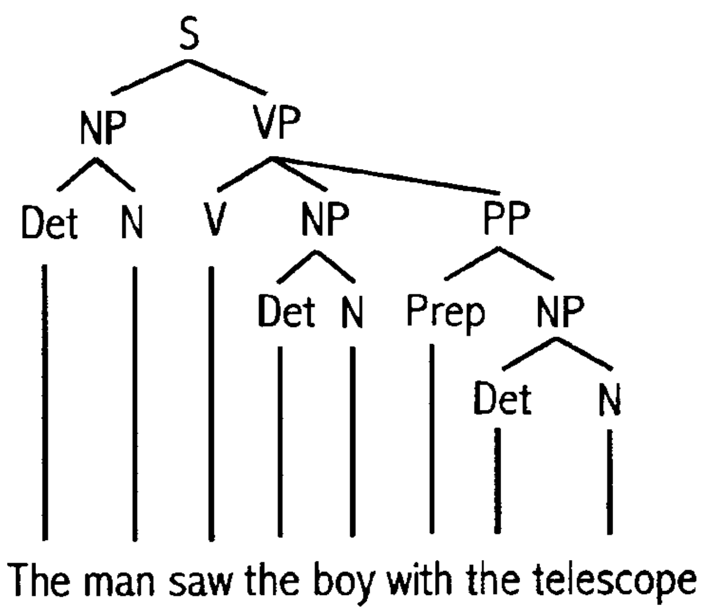
- Many possible parses for a single sentence happens very often…
  - Prepositional phrase attachment (PP-attachment)
    - *We painted the wall with cracks.*
    - *The man saw the boy with the telescope.*
    - *I shot an elephant in my pyjamas.*
  - Conjunctions and appositives
    - *Maddy, my dog, and Samy*

    -- > (Maddy, my dog), and (Samy)

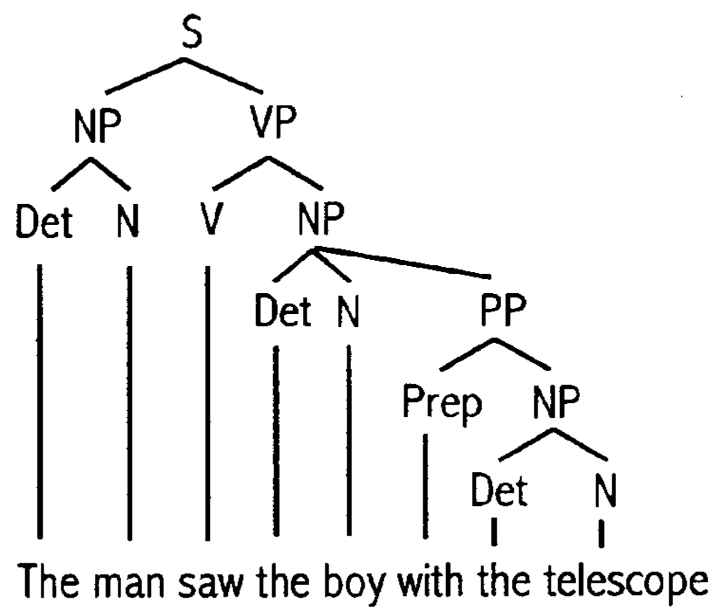
    -- > (Maddy), (my dog), and (Samy)

- These phenomena can quickly increase the number of possible parse trees!

PP attachment:
The man saw the boy with the telescope.



Correct parse 1



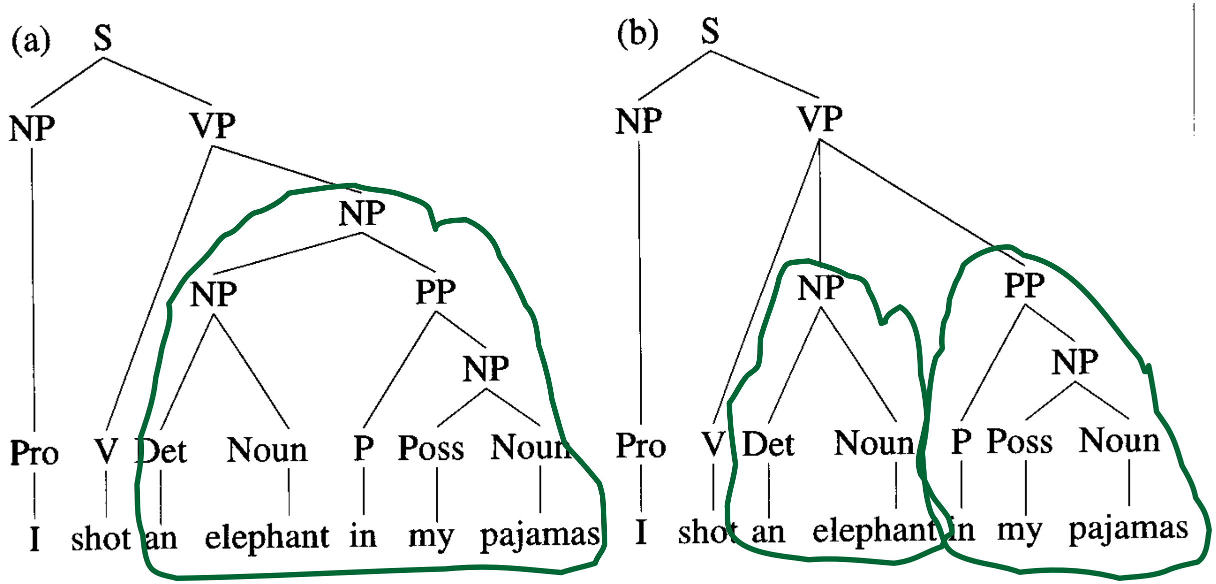
Correct parse 2

source: Robert Dale.

Probabilistic Parsing

"One morning I shot an elephant in my pyjamas. How he got into my pyjamas, I don't know."

G. Marx, *Animal Crackers*, 1930.



- Sentences can be very ambiguous…
  - A non-probabilistic parser may find a large set of possible parses
  - --> need to pick the most probable parse one from the set

Example of a PCFG

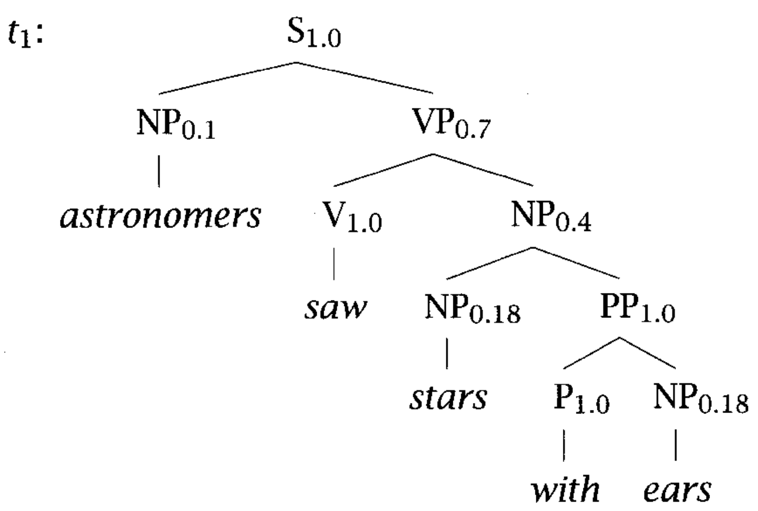
$S \rightarrow NP\ VP$	1.0	$NP \rightarrow NP\ PP$	0.4
$PP \rightarrow P\ NP$	1.0	$NP \rightarrow$ *astronomers*	0.1
$VP \rightarrow V\ NP$	0.7	$NP \rightarrow$ *ears*	0.18
$VP \rightarrow VP\ PP$	0.3	$NP \rightarrow$ *saw*	0.04
$P \rightarrow$ *with*	1.0	$NP \rightarrow$ *stars*	0.18
$V \rightarrow$ *saw*	1.0	$NP \rightarrow$ *telescopes*	0.1

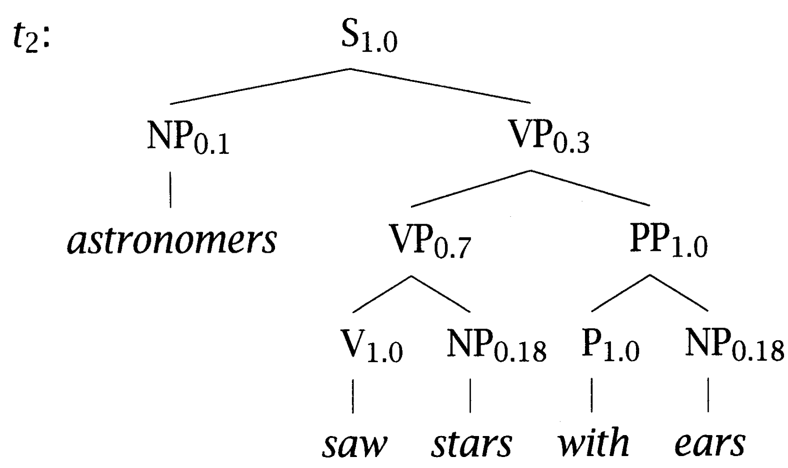
- Intuitively, P(VP $\rightarrow$ V NP) is:
  - the probability of expanding VP by a V NP, as opposed to any other rules for VP
- So for:
  - VP: $\forall i\ \sum_i P(VP \dashrightarrow B) = .7 + .3 = 1$
  - NΠ: $\forall i\ \sum_i P(NP \dashrightarrow B) = .4 + .1 + .18 + .04 + .18 + .1 = 1$

source: Manning, and Schütze, Foundations of Statistical Natural Language Processing, MIT Press (1999)

Probability of a parse tree

- Product of the probabilities of the rules used in subtrees
- Ex: "*Astronomers saw stars with ears.*"





$P(t_1) = 1 \times .1 \times .7 \times 1 \times .4 \times .18 \times 1 \times 1 \times .18$
$= .0009072$

$P(t_2) = 1 \times .1 \times .3 \times .7 \times 1 \times 1 \times .18 \times 1 \times .18$
$= .0006804$

source: Manning, and Schütze, Foundations of Statistical Natural Language Processing, MIT Press (1999)

Stages of NLU

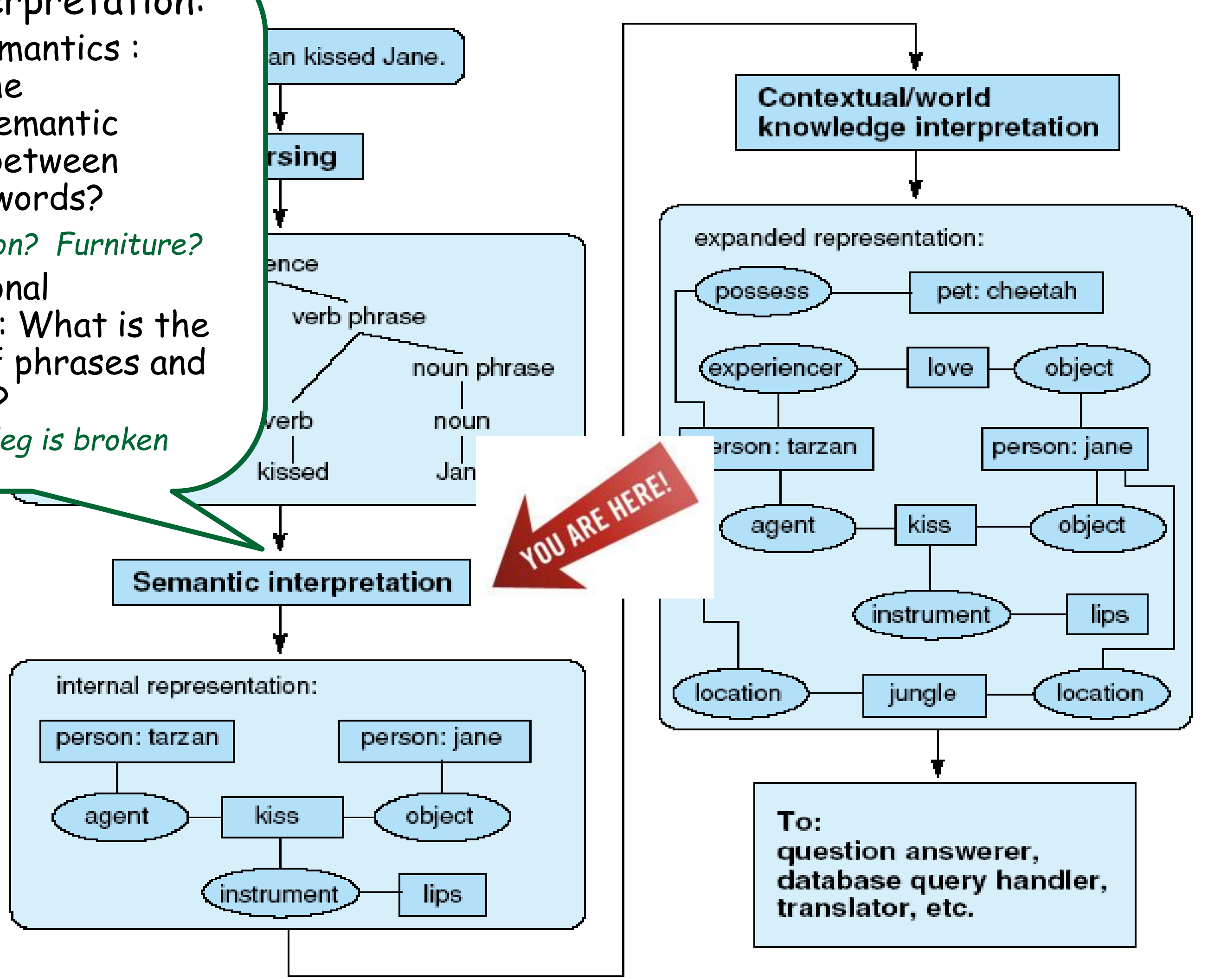
Semantic interpretation:

- Lexical Semantics : What is the meaning/semantic relations between individual words?

  Chair: person? Furniture?

- Compositional Semantics: What is the meaning of phrases and sentences?

  The chair's leg is broken



source: Luger (2005)

Semantic Interpretation

- Map sentences to some representation of its meaning
  - e.g., logics, knowledge graph, embedding…

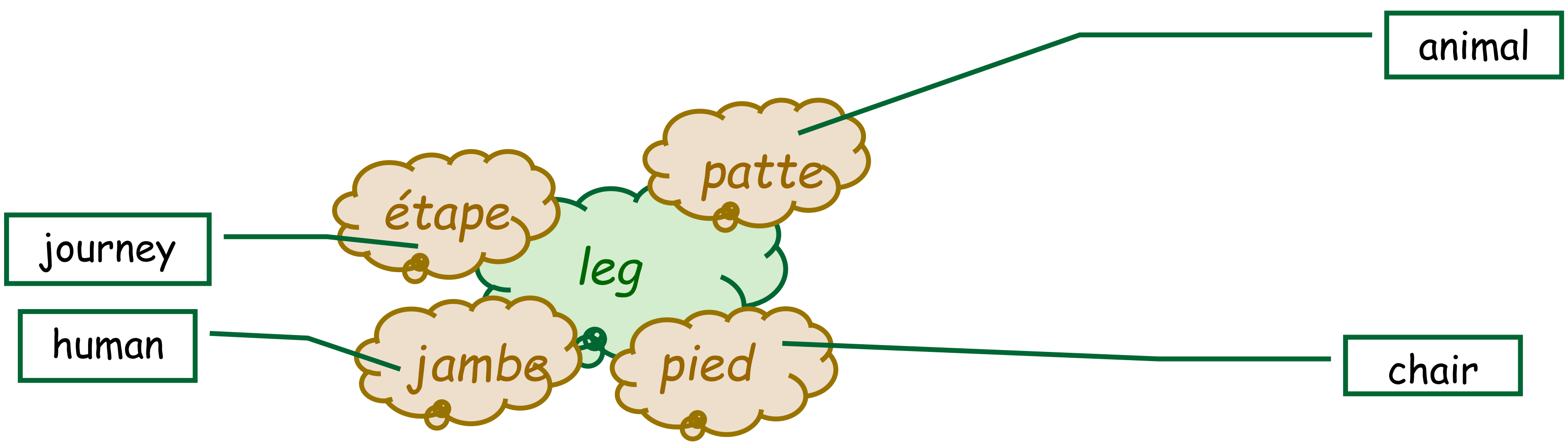
1. Lexical Semantics
   - i.e., Meaning of individual words

1. Compositional Semantics
   - i.e., Meaning of combination of words

Lexical Semantics

- ie. The meaning of individual words
  - A word may denote different things (ex. chair)
  - The meaning/sense of words is not clear-cut

  - E.g. Overlapping of word senses across languages



Word Sense Disambiguation (WSD)

- Determining which sense of a word is used in a specific sentence
  - *I went to the <u>bank</u> of Montreal and deposited 50$.*
  - *I went to the <u>bank</u> of the river and dangled my feet.*

WSD as a Classification Problem

- WSD can be viewed as typical classification problem
  - use machine learning techniques (ex. Naïve Bayes classifier, decision tree) to train a system
  - that learns a classifier (a function *f*) to assign to unseen examples one of a fixed number of senses (categories)
- Input:
  - Target word: The word to be disambiguated
  - Features?
- Output:
  - Most likely sense of the word

Features for WSD

- intuition:
  - sense of a word depends on the sense of surrounding words
- ex: *bass* = fish, musical instrument, ...

Surrounding words	Most probable sense
…river…	fish
…violin…	instrument
…salmon…	fish
…play…	instrument
…player…	instrument
…striped…	fish

- So use a window of words around the target word as features

Features for WSD

- Take a window of *n* words around the target word
- Encode information about the words around the target word
  - *An electric guitar and **bass** player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.*

Naïve Bayes WSD

- Goal: choose the most probable sense s^* for a word given a vector V of surrounding words

- Feature vector V contains:
  - Features: words [*fishing, big, sound, player, fly, rod, …*]
  - Value: frequency of these words in a window before & after the target word [0, 0, 0, 2, 1, 0, …]

- Bayes decision rule:
  - $s^* = \text{argmax}_{s_k} P(s_k|V)$
  - where:
    - S is the set of possible senses for the target word
    - s_k is a sense in S
    - V is the feature vector

Naïve Bayes WSD

$$s^* = \underset{s_k}{\operatorname{argmax}} \left(\log P(s_k) + \sum_{j=1}^{n} \log P(v_j \mid s_k) \right)$$

- Training a Naïve Bayes classifier
  = estimating $P(v_j|s_k)$ and $P(s_k)$ from a sense-tagged training corpus
  = finding the most likely sense k

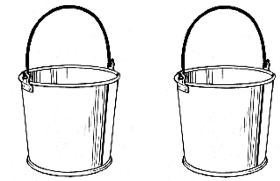
$$P(v_j \mid s_k) = \frac{\text{count}(v_j, s_k)}{\sum_t \text{count}(v_t, s_k)}$$

Nb of occurrences of feature j over the total nb of features appearing in windows of S_k

$$P(s_k) = \frac{\text{count}(s_k)}{\text{count}(\text{word})}$$

Nb of occurrences of sense k over nb of all occurrences of ambiguous word

Example



BANK1 BANK2

- Training corpus (context window = ±3 words):

  ...Today the World **Bank**/BANK1 and partners are calling for greater relief...
  ...Welcome to the **Bank**/BANK1 of America the nation's leading financial institution...
  ...Welcome to America's Job **Bank**/BANK1 Visit our site and...
  ...Web site of the European Central **Bank**/BANK1 located in Frankfurt...
  ...The Asian Development **Bank**/BANK1 ADB a multilateral development finance...

  ...lounging against verdant **banks**/BANK2 carving out the...
  ...for swimming, had warned her off the **banks**/BANK2 of the Potomac. Nobody...

- Training:
  - P(*the*|BANK1) = 5/30 P(*the*|BANK2) = 3/12
  - P(*world*|BANK1) = 1/30 P(*world*|BANK2) = 0/12
  - P(*and*|BANK1) = 1/30 P(*and*|BANK2) = 0/12
  - ... ...
  - P(*off*|BANK1) = 0/30 P(*off*|BANK2) = 1/12
  - P(*Potomac*|BANK1) = 0/30 P(*Potomac*|BANK2) = 1/12
  - P(BANK1) = 5/7 P(BANK2) = 2/7

- Disambiguation: "*I lost my left shoe on the* ***banks*** *of the river* Nile."
  - Score(BANK1)=log(5/7) + log(P(*shoe*|BANK1))+log(P(*on*|BANK1))+log(P(*the*|BANK1)) ...
  - Score(BANK2)=log(2/7) + log(P(*shoe*|BANK2)+log(P(*on*|BANK2))+log(P(*the*|BANK2)) ...

Example (with *add 0.5 smoothing*)

- Training corpus (context window = ±3 words):

  …Today the World **Bank/**BANK1 and partners are calling for greater relief…
  …Welcome to the **Bank/**BANK1 of America the nation's leading financial institution…
  …Welcome to America's Job **Bank/**BANK1 Visit our site and…
  …Web site of the European Central **Bank/**BANK1 located in Frankfurt…
  …The Asian Development **Bank/**BANK1 ADB a multilateral development finance…

  …lounging against verdant **banks/**BANK2 carving out the…
  …for swimming, had warned her off the **banks/**BANK2 of the Potomac. Nobody…

- Assume V = 50
- Training:
  - P(*the*|BANK1) = (5+.5) / (30+.5V) P(*the*|BANK2) = (3+.5) / (12 + .5V)
  - P(*world*|BANK1) = (1+.5) / 55 P(*world*|BANK2) = (0+.5) / 37
  - P(*and*|BANK1) = (1+.5) / 55 P(*and*|BANK2) = (0+.5) / 37
  - …
  - P(*off*|BANK1) = (0+.5) / 55 P(*off*|BANK2) = (1+.5) / 37
  - P(*Potomac*|BANK1) = (0+.5) / 55 P(*Potomac*|BANK2) = (1+.5) / 37
  - P(BANK1) = 5/7
    P(BANK2) = 2/7
- Disambiguation: "*I lost my left shoe on the **banks** of the river* Nile."
  - Score(BANK1)=log(5/7) + log(P(*shoe*|BANK1))+log(P(*on*|BANK1))+log(P(*the*|BANK1)) …
  - Score(BANK2)=log(2/7) + log(P(*shoe*|BANK2))+log(P(*on*|BANK2))+log(P(*the*|BANK2)) …

Stages of NL Understanding

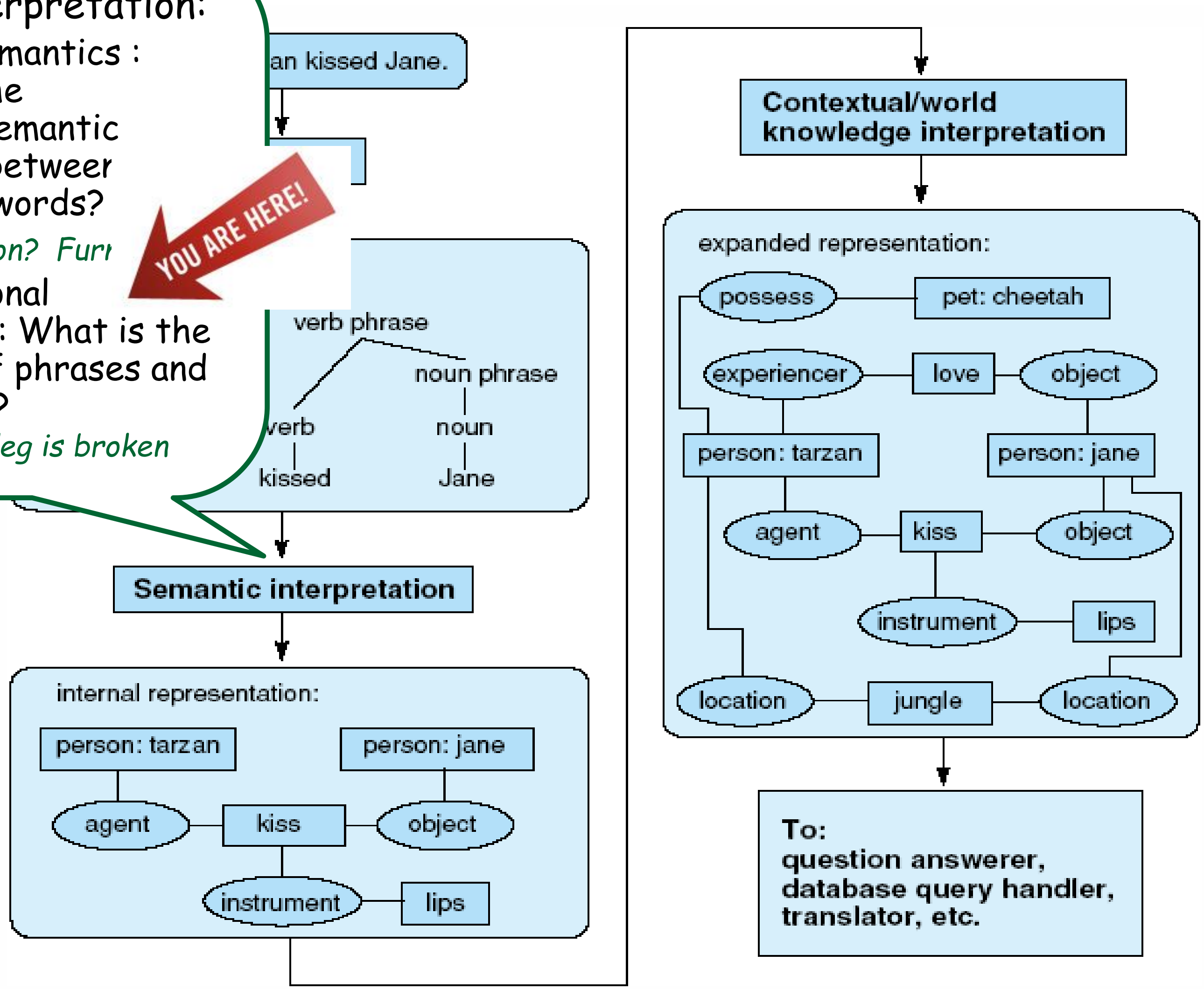
Semantic interpretation:

- Lexical Semantics : What is the meaning/semantic relations between individual words?

  Chair: person? Furr

- Compositional Semantics: What is the meaning of phrases and sentences?

  The chair's leg is broken



source: Luger (2005)

Compositional Semantics

- *The cat eats the mouse = The mouse is eaten by the cat.*
- Goal:
  - map an expression into a **knowledge representation**
    - a representation of context-independent, literal meaning
    - e.g. first-order predicate logic, conceptual graph, embedding...
  - to assign semantic roles (different from grammatical roles):
    - Semantic roles: Agent, Patient, Instrument, Time, Location, …
    - Grammatical roles: subject, direct object, …
- E.g.
  - *The child hid the candy under the bed.*

```
Hide (agent=child, patient=candy,
          location=under_the_bed, time=past)
```

Some Difficulties

- Syntax is not enough
  - *I ate spaghetti with a fork.* <instrument>
  - *I ate spaghetti with my sister.* <accompanying person>
  - *I ate spaghetti with meat balls.* <attribute of food>
  - *I ate spaghetti with lots of appetite.* <manner>

    …

  - *Gun* = instrument that can kill
  - *Metal gun*… a gun made out of metal
  - *Water gun*… a gun made out of water?
  - *Fake gun*… it is a gun anyways? Can it kill?

  - *General Kane*… person but *General Motors* … corporation
- Parallel problems to syntactic ambiguity
  - *Happy [cats and dogs] live on the farm*
  - *[Happy cats] and dogs live on the farm*
- Quantifier Scoping
  - *Every man loves a woman.*
    - $\forall$ m ($\exists$ f man(m) $\land$ woman(f) $\land$ loves(m, f))
    - $\exists$ f($\forall$ m man(m) $\land$ woman(f) $\land$ loves(m, f))

Stages of NLU

- Discourse Analysis

How to relate the meaning of sentences to surrounding sentences?

I have to go to the store. I need butter.
I have to go to the university. I need butter.

- Pragmatics

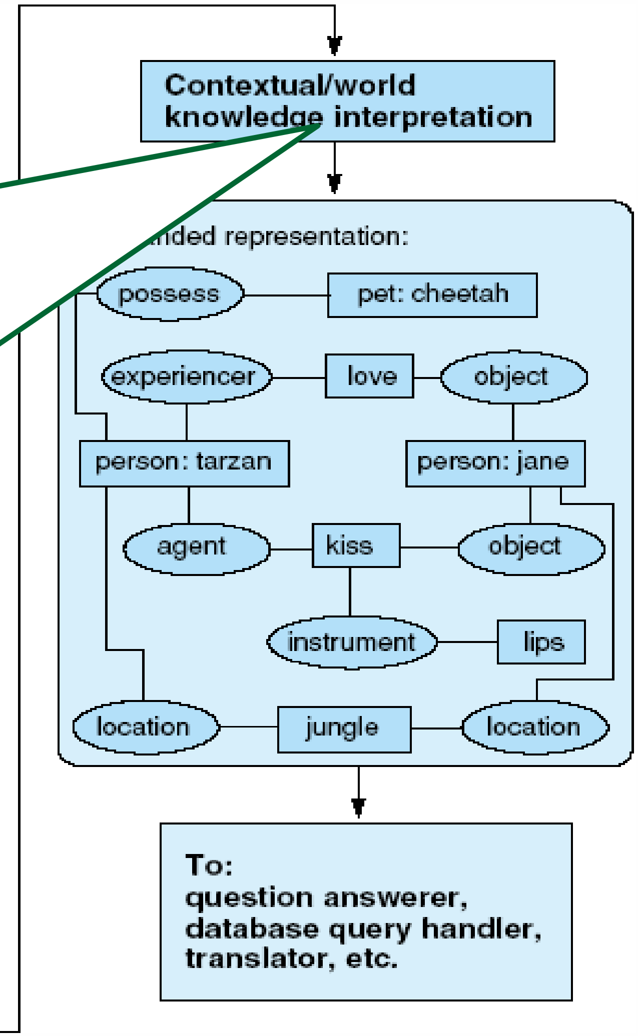
How people use language in a social environment?

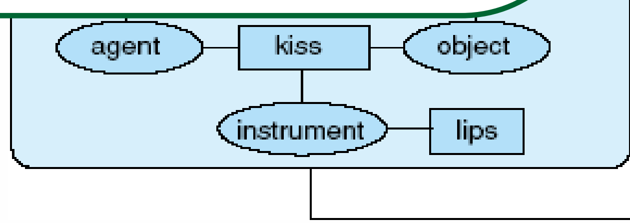
Do you have a child?
Do you have a quarter?

- World Knowledge

How knowledge about the world (history, facts, …) modifies our understanding of text?

Bill Gates passed away last night.





source: Luger (2005)

Discourse Analysis

- In logics: $A \wedge B \wedge C \Leftrightarrow C \wedge B \wedge A$
- Not in NL:
  - *John visited Paris. He bought Mary some expensive perfume. Then he flew home. He went to Walmart. He bought some underwear.*
  - *John visited Paris. Then he flew home. He went to Walmart. He bought Mary some expensive perfume. He bought some underwear.*
- Humans infer relations between sentences that may not be explicitly stated in order to make a text coherent.
  - *(?) I am going to Concordia. I need butter.*

Examples of Discourse Relations

CONDITION *If it rains, I will go out.*

SEQUENCE *Do this, then do that.*

CONTRAST *This is good, but this is better.*

CAUSE *Because I was sick, I could not do my assignment.*

RESULT *Click on the button, the red light will blink.*

PURPOSE *To use the computer, get an access code.*

ELABORATION *The solution was developed by Alan Turing. Turing was a great mathematician living in Great Britain. He was an atheist as well as gay.*

Another Classification Problem, again!

- Discourse tagging can be viewed as typical classification problem
  - use machine learning techniques (ex. Naïve Bayes classifier, decision tree) to train a system
  - that learns a classifier to assign to unseen sentences one of a fixed number of discourse relations (categories)

- Input:
  - Sentence Ex. *If it rains, I will go out.*
  - Features?
    - Connectives such as "*if*", "*however*", "*in conclusion*"
    - Tense of verb (future, past)
    - …
- Output:
  - Most likely relation in the sentence (none, condition, contrast, purpose, …)

Stages of NLU

- Discourse Analysis

How to relate the meaning of sentences to surrounding sen

I have to go to the *utter.*

I have to go to the *ed butter.*

- Pragmatics

How people use language in a social environment?

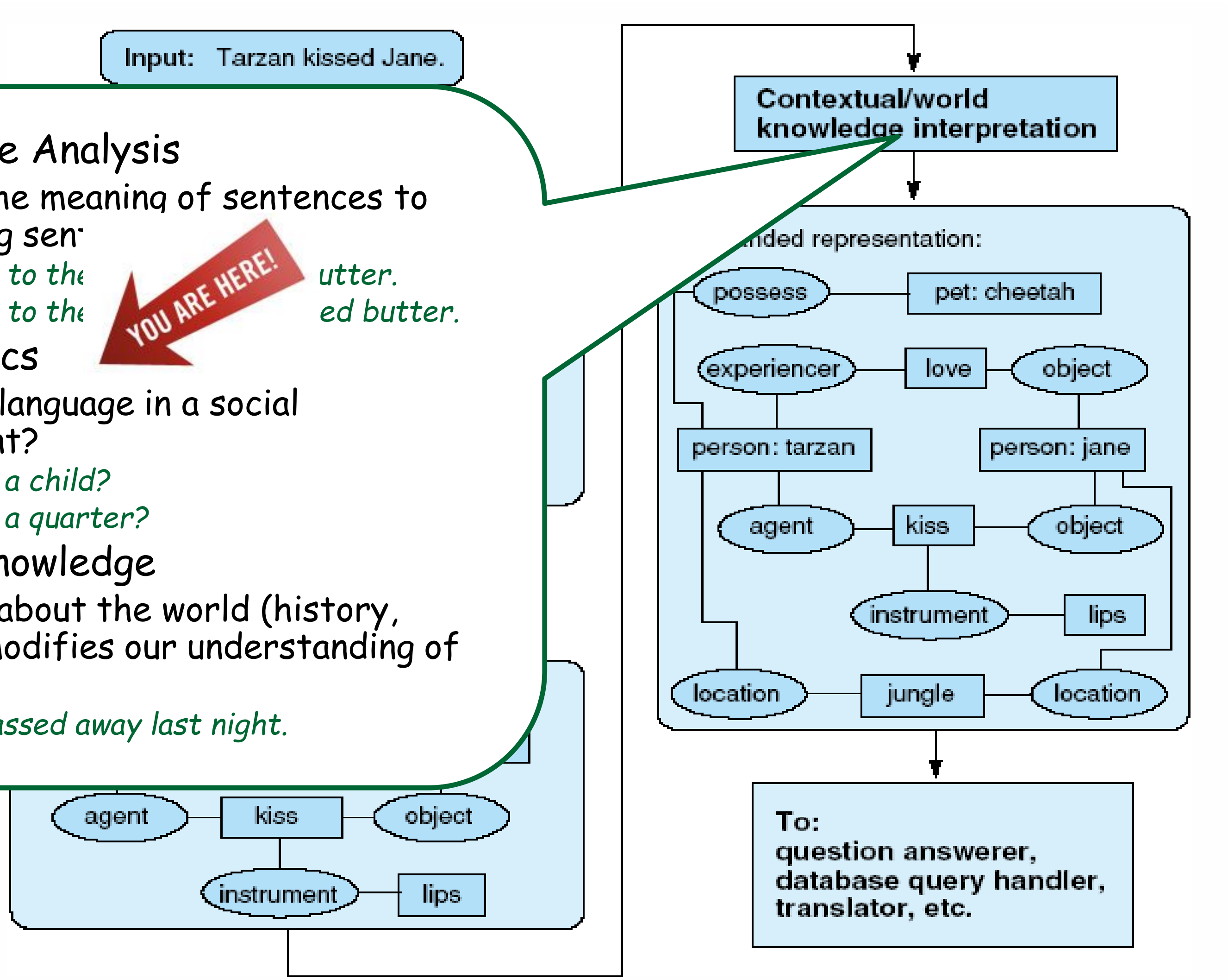
Do you have a child?

Do you have a quarter?

- World Knowledge

How knowledge about the world (history, facts, …) modifies our understanding of text?

Bill Gates passed away last night.



source: Luger (2005)

Pragmatics

- goes beyond the literal meaning of a sentence
- tries to explain what the speaker is really expressing
- understanding how people use language socially
  - E.g.: figures of speech, …
  - E.g.: *Could you spare some change?*

Stages of NLU

- **Discourse Analysis**

How to relate the meaning of sentences to surrounding sentences?

I have to go to the store. I need butter.
I have to go to the university. I need butter.

- **Pragmatics**

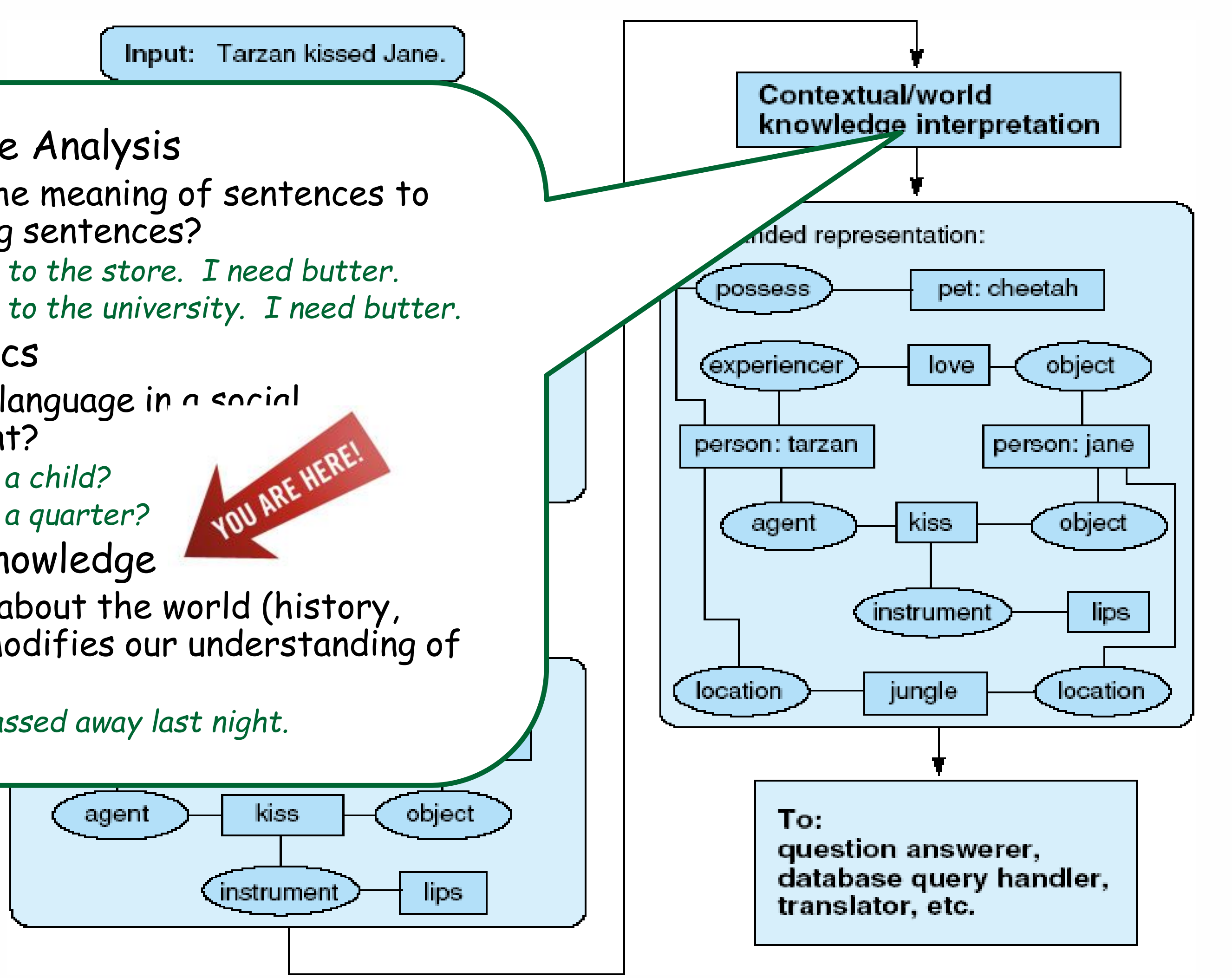
How people use language in a social environment?

Do you have a child?
Do you have a quarter?

- **World Knowledge**

How knowledge about the world (history, facts, …) modifies our understanding of text?

Bill Gates passed away last night.



source: Luger (2005)

Using World Knowledge

- Using our general knowledge of the world to interpret a sentence/discourse
- E.g.:

  The trophy would not fit in the brown suitcase because ...

  ... it was too big.

  ... it was too small.

  The professor sent the student to see the principal because…

  …he wanted to see him.

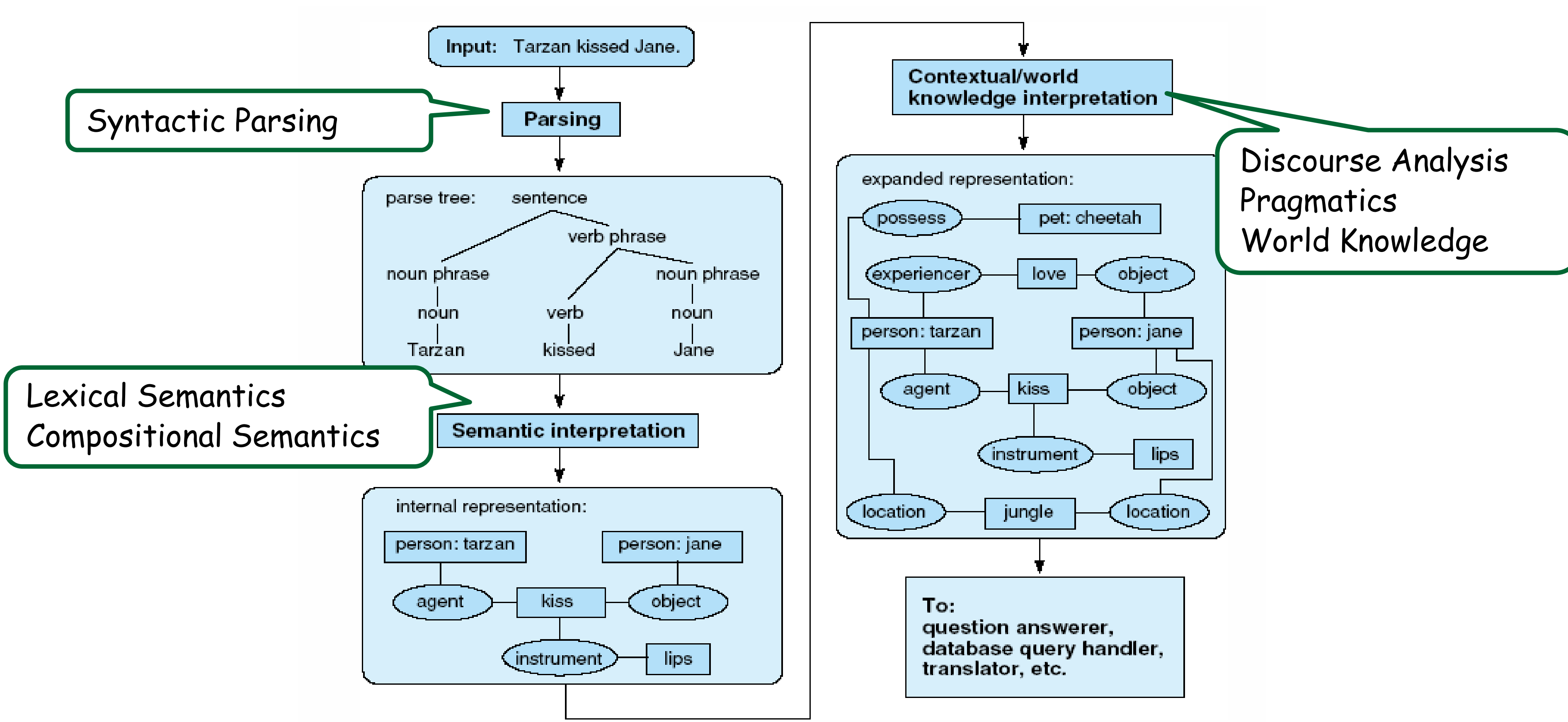
  …he was throwing paper balls in class.

  …he could not take it anymore.

  - *Ex: Silence of the lambs…*

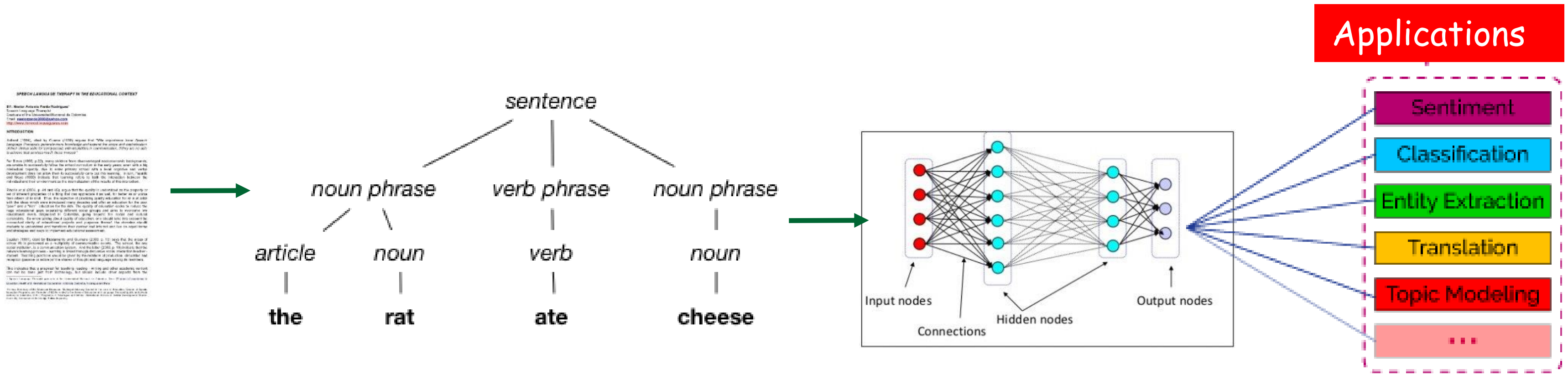
Current Research area: see Winograd Schema Challenge

Summary of NLU



source: Luger (2005)

Recap



linguistic features are hand-engineered and fed to the ML model

Remember these slides?

History of AI

- Another big "hype" … Expert Systems (70s - mid 80s)
  - people realized that general-purpose problem solving (weak methods) do not work for practical applications
  - systems need specific domain-dependent knowledge (strong methods)
  - development of knowledge-intensive, rule-based techniques
  - major expert systems
    - MYCIN (1972): expert system to diagnose blood diseases
  - In the industry (1980s): First expert system shells and commercial applications.

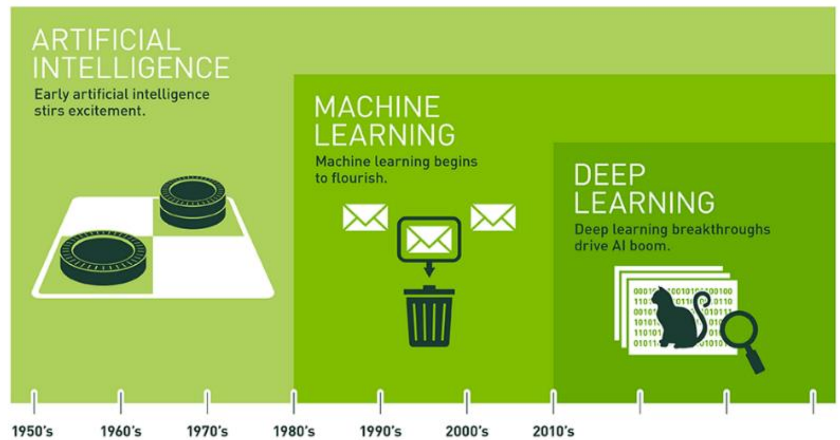
☹ HUMANS need to write the rules by hand…

History of AI

- The rise of Machine Learning (1980s - 2010)
  - More powerful CPUs-> usable implementation of neural networks
  - Big data -> Huge data sets are available to learn from
    - document repositories in NLP, datasets in ML, billions on images for image retrieval, billions of genomic sequences, …
  - ☺ Rules are now learned automatically !
  - AI adopts the Scientific Method

History of AI

- The era of Deep Learning (2010-today)
  - Development of "deep neural networks"
  - Trained on massive data sets
  - Use of GPU for computations
  - Use of "generic networks" for many applications



(to see in a few classes)