Introduction to Natural Language Processing

Ch. 22, 23

Jane Wagner: *"We speculated what it was like before we got language skills. When we humans had our first thought, most likely we didn't know what to think. It's hard to think without words cause you haven't got a clue as to what you're thinking. So if you think we suffer from a lack of communication now, think what it must have been like then, when people lived in a verbal void - made worse by the fact that there were no words such as verbal void."*

(Slides adapted from Paula Matuszek, Villanova)

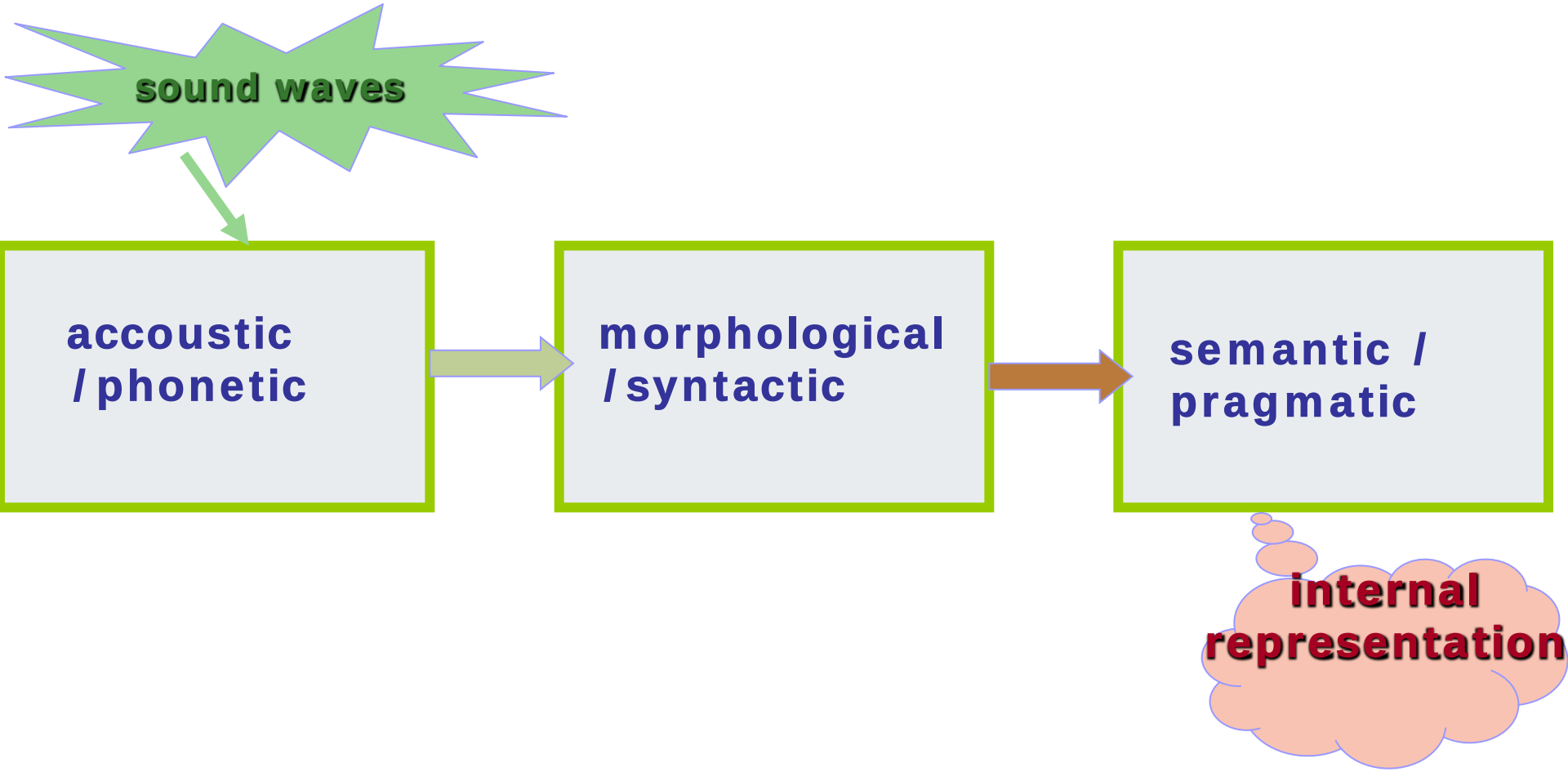
Natural Language Processing

- speech recognition
- natural language understanding
- computational linguistics
- psycholinguistics
- information extraction
- information retrieval
- inference
- natural language generation
- speech synthesis
- language evolution

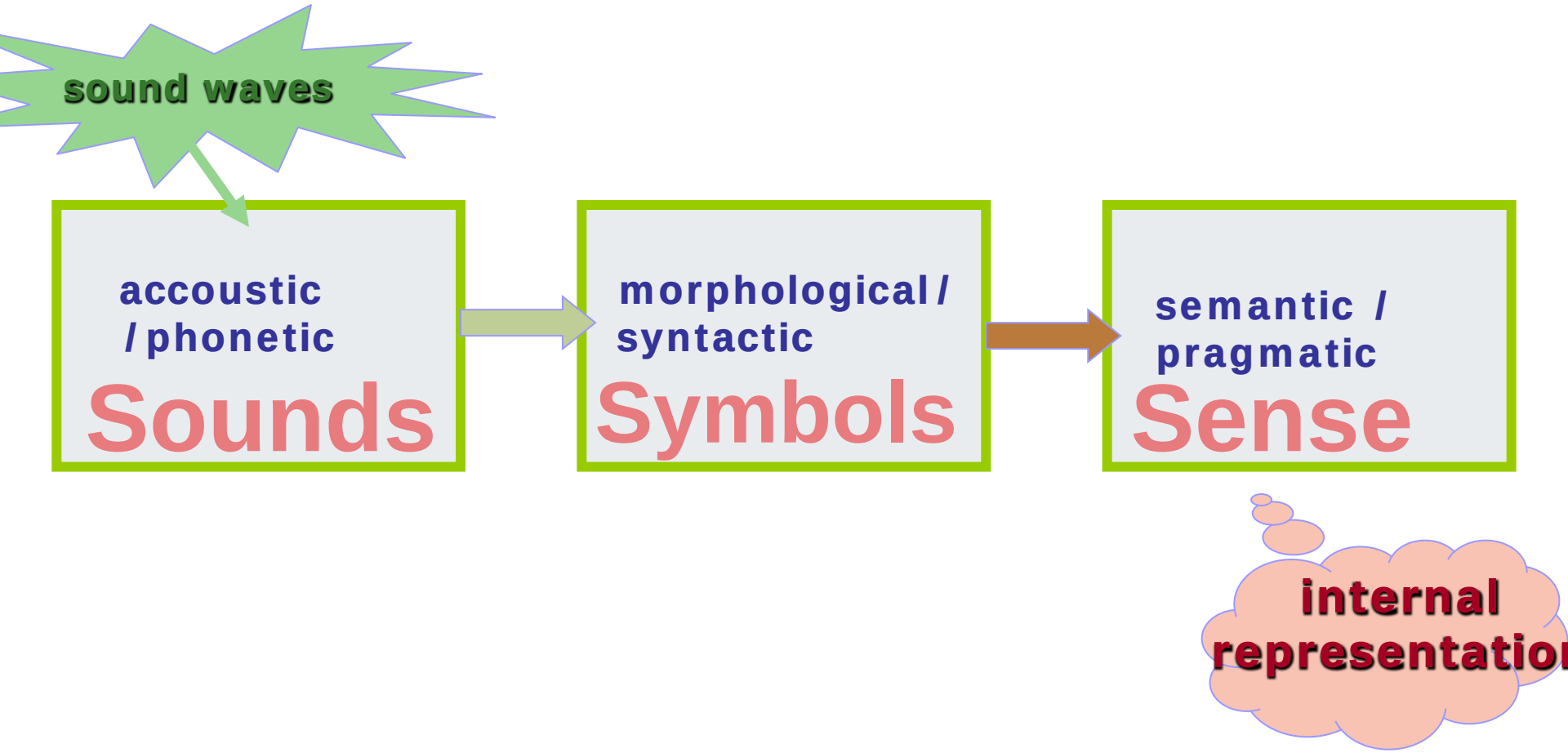
Applied NLP

- Machine translation
- Spelling/grammar correction
- Information Retrieval
- Data mining
- Document classification
- Question answering, conversational agents

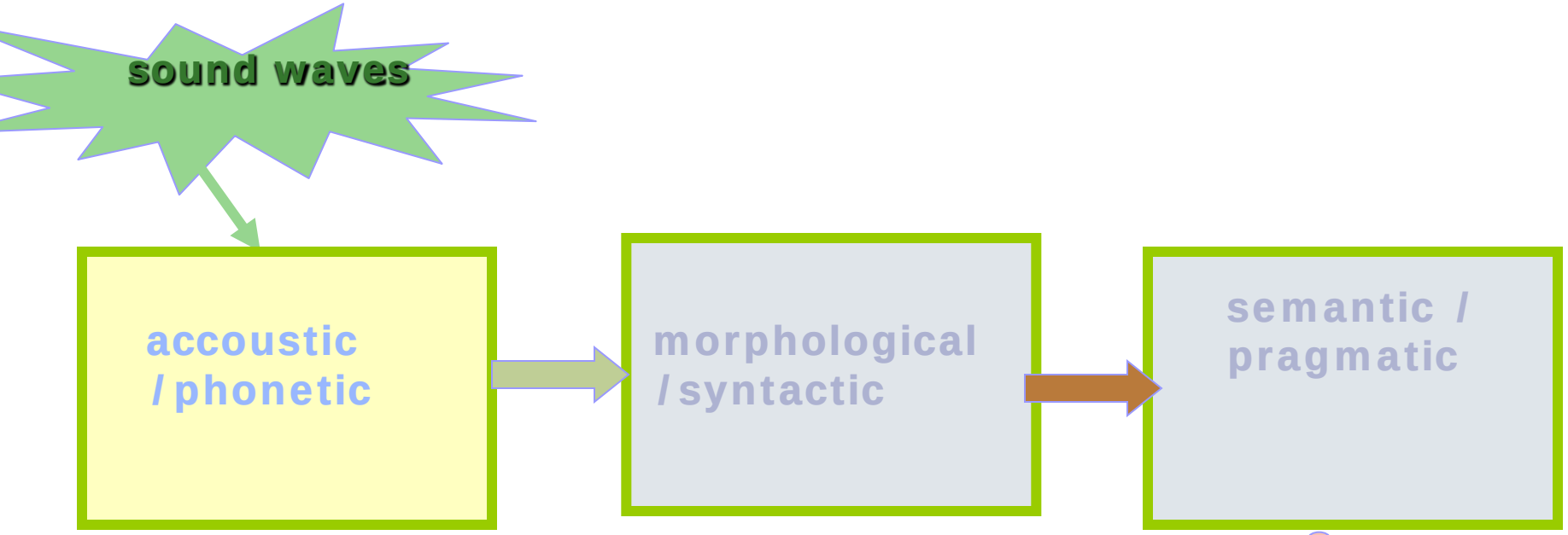
Natural Language Understanding



Natural Language Understanding



Where are the words?

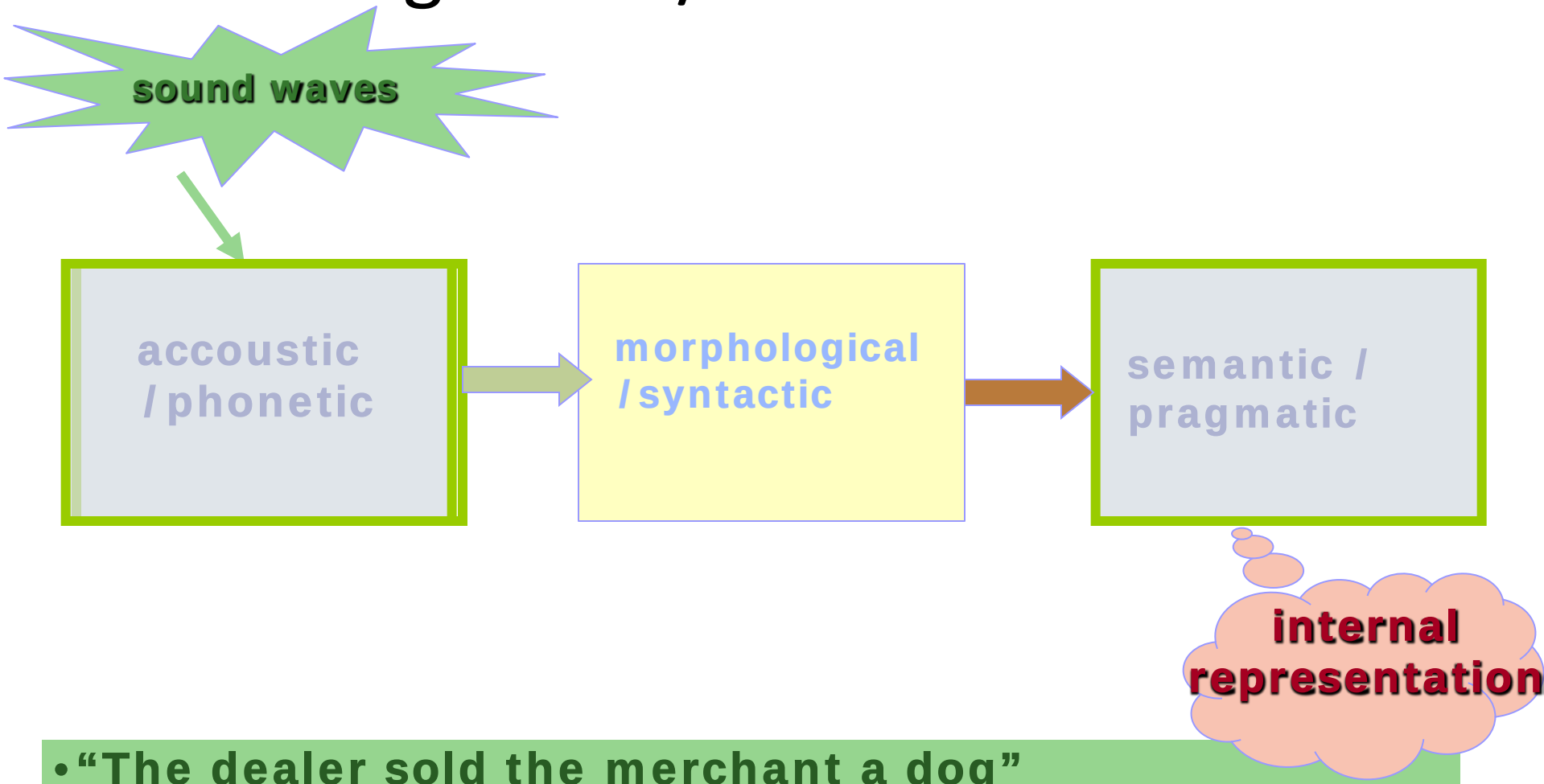


- "How to recognize speech, not to wreck a nice beach"
- "The cat scares all the birds away"
- "The cat's cares are few"

internal representation

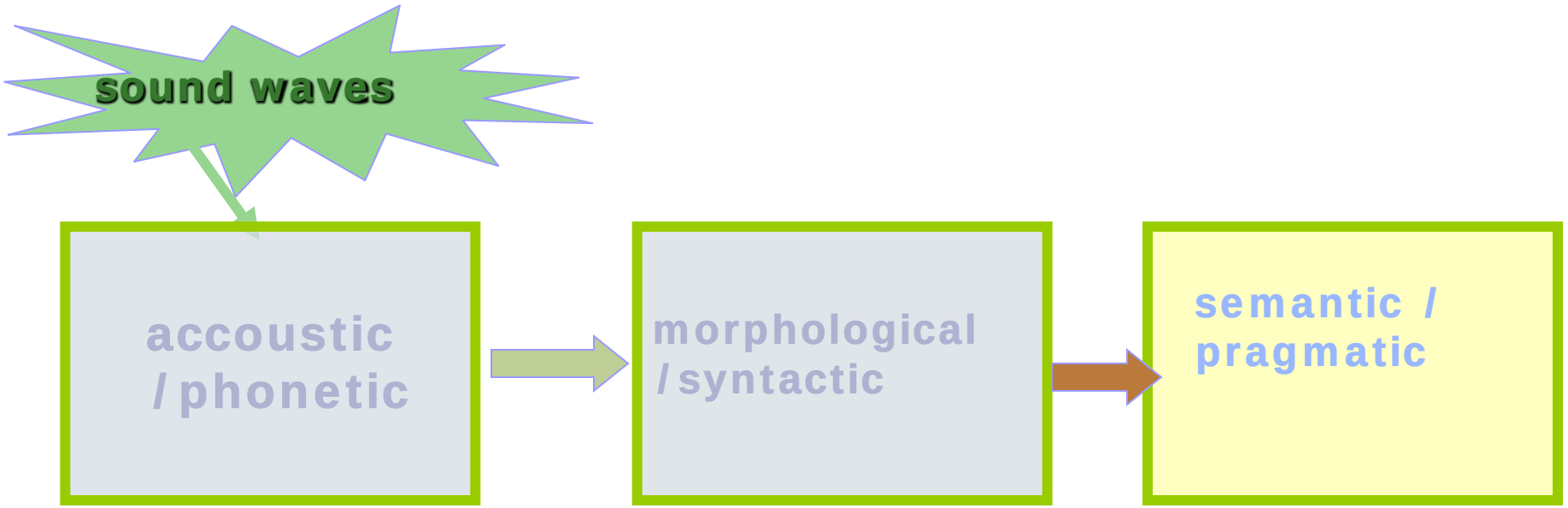
- pauses in speech bear little relation to word breaks
+ intonation offers additional clues to meaning

Dissecting words/sentences



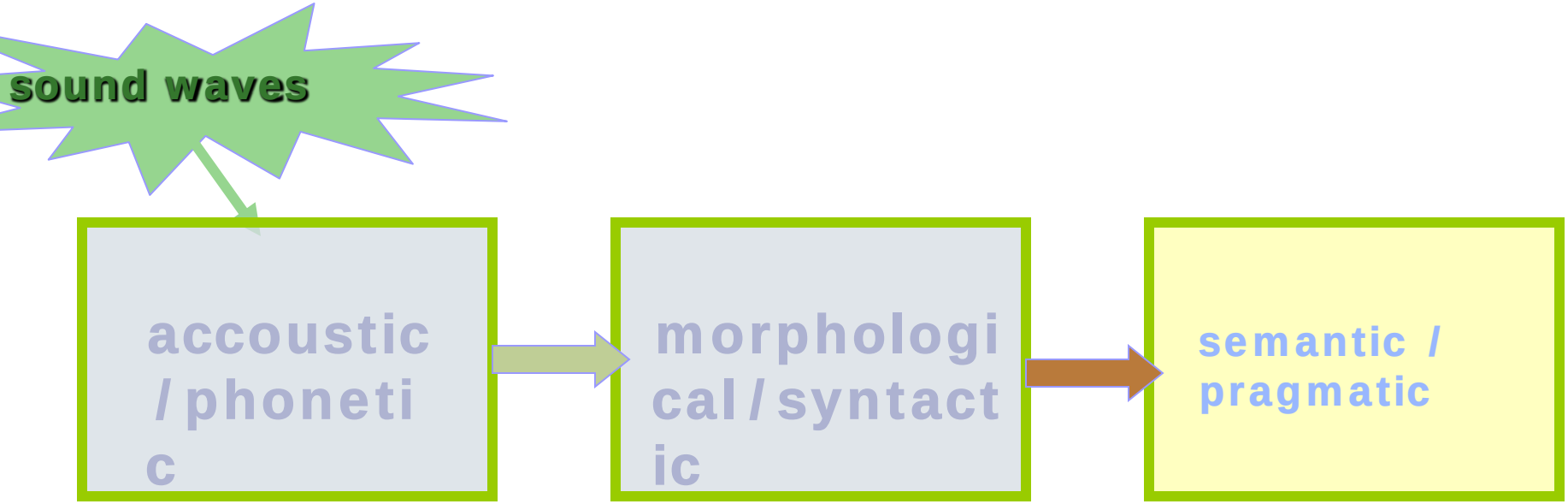
- "The dealer sold the merchant a dog"
- "I saw the Golden bridge flying into San Francisco"

What does it mean?



- "I saw Pathfinder on Mars with a telescope"
- "Pathfinder photographed Mars"
- "The Pathfinder photograph from Ford has arrived"
- "When a Pathfinder fords a river it sometimes mars its paint job."

What does it mean?



- "Jack went to the store. He found the milk in aisle 3. He paid for it and left."
- " Q: Did you read the report?

A: I read Bob's email."

The steps in NLP

- Morphology: Concerns the way words are built up from smaller meaning bearing units.
- Syntax: concerns how words are put together to form correct sentences and what structural role each word has
- Semantics: concerns what words mean and how these meanings combine in sentences to form sentence meanings

The steps in NLP (Cont.)

- Pragmatics: concerns how sentences are used in different situations and how use affects the interpretation of the sentence
- Discourse: concerns how the immediately preceding sentences affect the interpretation of the next sentence

Some of the Tools

- Regular Expressions and Finite State Automata
- Part of Speech taggers
- N-Grams
- Grammars
- Parsers
- Semantic Analysis

Parsing (Syntactic Analysis)

- Assigning a syntactic and logical form to an input sentence
  - uses knowledge about word and word meanings (lexicon)
  - uses a set of rules defining legal structures (grammar)
- Charlie ate the apple.
- (S (NP (NAME Charlie))
- (VP (V ate)
- (NP (ART the)
- (N apple))))

Word Sense Resolution

- Many words have many meanings or senses
- We need to resolve which of the senses of an ambiguous word is invoked in a particular use of the word
- I made her duck. (made her a bird for lunch or made her move her head quickly downwards?)

Human Languages

- You know ~50,000 words of primary language, each with several meanings
- Six year old knows ~13000 words
- First 16 years we learn 1 word every 90 min of waking time
- Mental grammar generates sentences -virtually every sentence is novel
- 3 year olds already have 90% of grammar
- ~6000 human languages – none of them simple!

Human Spoken language

- Most complicated mechanical motion of the human body
  - Movements must be accurate to within mm
  - synchronized within hundredths of a second
- We can understand up to 50 phonemes/sec (normal speech 10-15ph/sec)
  - but if sound is repeated 20 times /sec we hear continuous buzz!
- All aspects of language processing are involved and manage to keep apace

Why Language is Hard

- NLP is “AI-complete”
- Abstract concepts are difficult to represent
- LOTS of possible relationships among concepts
- Many ways to represent similar concepts
- Tens of hundreds or thousands of features/dimensions

Why Language is Easy

- Highly redundant

- Many relatively crude methods provide fairly good results

History of NLP

- Prehistory (1940s, 1950s)
  - Automata theory, formal language theory, Markov processes (Turing, McCullock & Pitts, Chomsky)
  - Information theory and probabilistic algorithms (Shannon)
  - Turing test – can machines think?

History of NLP

- Early work:
  - Symbolic approach
    - Generative syntax – e.g., Transformations and Discourse Analysis Project (TDAP- Harris)
    - AI – pattern matching, logic-based, special-purpose systems
      - Eliza -- Rogerian therapist http://www.manifestation.com/neurotoys/eliza.php3
  - Stochastic
    - Bayesian methods
  - Early successes -- $$$$ grants!
  - By 1966 US government had spent 20 million on machine translation

History of NLP

- Critics:
  - Bar Hillel – “no way to disambiguation without deep understanding”
  - Pierce NSF 1966 report: “no way to justify work in terms of practical output”

History of NLP

- The middle ages (1970-1990)
  - stochastic
    - speech recognition and synthesis (Bell Labs)
  - logic-based
    - compositional semantics (Montague)
    - definite clause grammars (Pereira&Warren)
  - ad hoc AI-based NLU systems
    - SHRDLU robot in blocks world (Winograd)
    - knowledge representation systems at Yale (Shank)
  - discourse modeling
    - anaphora
    - focus/topic (Groz et al)
    - conversational implicature (Grice)

History of NLP

- NLP Renaissance (1990-2000)
  - Lessons from phonology & morphology successes:
  - Finite-state models are very powerful
  - Probabilistic models pervasive
  - Web creates new opportunities and challenges
  - Practical applications driving the field again
- 21st Century NLP
  - The web changes everything:
  - Much greater use for NLP
  - Much more data available

Document Features

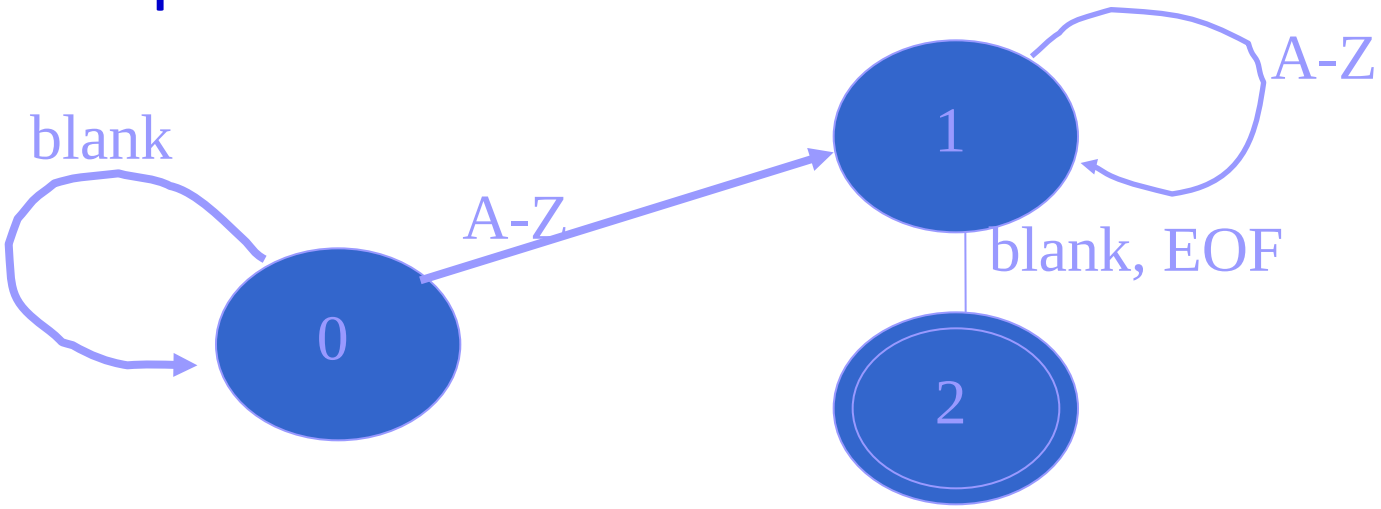
- Most NLP is applied to some quantity of unstructured text.
- For simplicity, we will refer to any such quantity as a document
- What features of a document are of interest?
- Most common is the actual terms in the document.

Tokenization

- Tokenization is the process of breaking up a string of letters into words and other meaningful components (numbers, punctuation, etc.
- Typically broken up at white space.
- Very standard NLP tool
- Language-dependent, and sometimes also domain-dependent.
  - 3,7-dihydro-1,3,7-trimethyl-1H-purine-2,6-dione
- Tokens can also be larger divisions: sentences, paragraphs, etc.

Lexical Analyser

- Basic idea is a finite state machine
- Triples of input state, transition token, output state



- Must be very efficient; gets used a LOT

Design Issues for Tokenizer

- Punctuation
  - treat as whitespace?
  - treat as characters?
  - treat specially?
- Case
  - fold?
- Digits
  - assemble into numbers?
  - treat as characters?
  - treat as punctuation?

NLTK Tokenizer

- Natural Language ToolKit
- http://text-processing.com/demo/tokenize/

- Call me Ishmael. Some years ago--never mind how long precisely--having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world.

N-Grams

- N-Grams are sequences of tokens.
- The N stands for how many terms are used
  - Unigram: 1 term
  - Bigram: 2 terms
  - Trigrams: 3 terms
- You can use different kinds of tokens
  - Character based n-grams
  - Word-based n-grams
  - POS-based n-grams
- N-Grams give us some idea of the context around the token we are looking at.

N-Gram Models of Language

- A language model is a model that lets us compute the probability, or likelihood, of a sentence S, P(S).
- N-Gram models use the previous N-1 words in a sequence to predict the next word
  - unigrams, bigrams, trigrams,…
- How do we construct or train these language models?
  - Count frequencies in very large corpora
  - Determine probabilities using Markov models, similar to POS tagging.

Counting Words in Corpora

- What is a word?
  - e.g., are cat and cats the same word?
  - September and Sept?
  - zero and oh?
  - Is _ a word? * ? ‘(‘ ?
  - How many words are there in don’t ? Gonna ?
  - In Japanese and Chinese text -- how do we identify a word?

Terminology

- Sentence: unit of written language
- Utterance: unit of spoken language
- Word Form: the inflected form that appears in the corpus
- Lemma: an abstract form, shared by word forms having the same stem, part of speech, and word sense
- Types: number of distinct words in a corpus (vocabulary size)
- Tokens: total number of words

Simple N-Grams

- Assume a language has V word types in its lexicon, how likely is word x to follow word y?
  - Simplest model of word probability: 1/ V
  - Alternative 1: estimate likelihood of x occurring in new text based on its general frequency of occurrence estimated from a corpus (unigram probability)
    - popcorn is more likely to occur than unicorn
  - Alternative 2: condition the likelihood of x occurring in the context of previous words (bigrams, trigrams,…)
    - mythical unicorn is more likely than mythical popcorn

Computing the Probability of a Word Sequence

- Compute the product of component conditional probabilities?
  - P(the mythical unicorn) = P(the) P(mythical|the) P(unicorn|the mythical)
- The longer the sequence, the less likely we are to find it in a training corpus
    - P(Most biologists and folklore specialists believe that in fact the mythical unicorn horns derived from the narwhal)
- Solution: approximate using n-grams

Bigram Model

- Approximate $P(w_n|w_1^{n-1})$, $P(w_n|w_{n-1})$
  - P(unicorn|the mythical) by P(unicorn|mythical)
- Markov assumption: the probability of a word depends only on the probability of a limited history
- Generalization: the probability of a word depends only on the probability of the ***n*** previous words
  - Trigrams, 4-grams, …
  - The higher n is, the more data needed to train
  - The higher n is, the sparser the matrix.

Using N-Grams

- For N-gram models
  - $P(w_n|w_1^{n-1})$ $P(w_n|w_{n-N+1}^{n-1})$
  - $P(w_{n-1},w_n) = P(w_n \mid w_{n-1})\, P(w_{n-1})$
  - By the Chain Rule we can decompose a joint probability, e.g. $P(w_1,w_2,w_3)$

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

    For bigrams then, the probability of a sequence is just the product of the conditional probabilities of its bigrams

    P(the,mythical,unicorn) = P(unicorn|mythical) P(mythical|the) P(the|<start>)

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$$

A Simple Example

- P(I want to eat Chinese food) =

  P(I | <start>) × P(want | I) ×

  P(to | want) × P(eat | to) ×

  P(Chinese | eat) × P(food | Chinese)

Counts from the Berkeley Restaurant Project

Nth term

N-1 term

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

BeRP Bigram Table

Nth term

N-1 term

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

A Simple Example

P(I want to eat Chinese food) =

P(I | <start>) × P(want | I) ×

P(to | want) × P(eat | to) ×

P(Chinese | eat) × P(food | Chinese)

$.25 \times .32 \times .65 \times .26 \times .02 \times .56 = .00015$

So What?

- P(I want to eat British food) = P(I|<start>) P(want|I) P(to|want) P(eat|to) P(British|eat) P(food|British) = .25 $\times$.32 $\times$.65 $\times$.26 $\times$.001 $\times$.60 = .000080
- vs. I want to eat Chinese food = .00015
- Probabilities seem to capture ``syntactic'' facts, ``world knowledge''
  - eat is often followed by an NP
  - British food is not too popular

Approximating Shakespeare

- As we increase the value of N, the accuracy of the n-gram model increases, since choice of next word becomes increasingly constrained
- Generating sentences with random unigrams...
  - Every enter now severally so, let
  - Hill he late speaks; or! a more to leg less first you enter
- With bigrams...
  - What means, sir. I confess she? then all sorts, he is trim, captain.
  - Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry.

- Trigrams
  - Sweet prince, Falstaff shall die.
  - This shall forbid it should be branded, if renown made it empty.
- Quadrigrams
  - What! I will go seek the traitor Gloucester.
  - Will you not tell me who I am?

- There are 884,647 tokens, with 29,066 word form types, in about a one million word Shakespeare corpus
- Shakespeare produced 300,000 bigram types out of 844 million possible bigrams: so, 99.96% of the possible bigrams were never seen (have zero entries in the table)
- Quadrigrams worse: What's coming out looks like Shakespeare because it ***is*** Shakespeare

N-Gram Training Sensitivity

- If we repeated the Shakespeare experiment but trained our n-grams on a Wall Street Journal corpus, what would we get?
- This has major implications for corpus selection or design

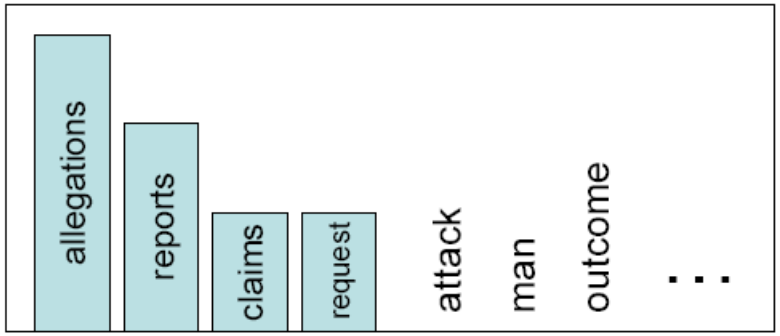
Some Useful Empirical Observations

- A few events occur with high frequency
- Many events occur with low frequency
- You can quickly collect statistics on the high frequency events
- You might have to wait an arbitrarily long time to get valid statistics on low frequency events
- Some of the zeroes in the table are really zeros But others are simply low frequency events you haven't seen yet. We smooth the frequency table by assigning small but non-zero frequencies to these terms.

Smoothing is like Robin Hood: Steal from the rich and give to the poor (in probability mass)

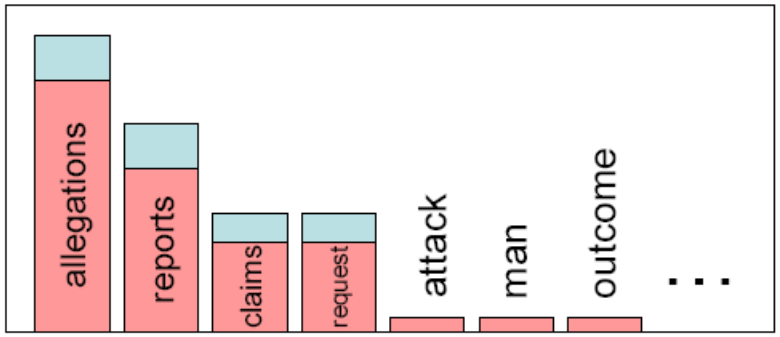
- We often want to make predictions from sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



- Smoothing flattens spiky distributions so they generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



- Very important all over NLP, but easy to do badly!

From Snow, http://www.stanford.edu/class/linguist236/lec11.ppt

All Our N-Grams Are Belong to You

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

- Google uses n-grams for machine translation, spelling correction, other NLP
- In 2006 they released a large collection of n-gram counts through the Linguistic Data Consortium, based on a trillion web pages
  - a trillion tokens, 300 million bigrams, about a billion tri-, four-and five-grams. (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13)
- This quantity of data makes a qualitative change in what we can do with statistical NLP.

Summary

- N-gram probabilities can be used to *estimate* the likelihood
  - Of a word occurring in a context (N-1)
  - Of a sentence occurring at all
- Smoothing techniques deal with problems of unseen words in a corpus
- N-grams are useful in a wide variety of NLP tasks