# Introduction to Natural Language Processing

Alok Govil

Technology Architect (Anything EE & CS)

alokgovil@hotmail.com

www.linkedin.com/in/alokgovil

www.alokgovil.com

Oct 12, 2014

Silicon Valley Code Camp

# Scope of the Presentation

- Well-rounded:  Linguistics, Algorithms, Insights
- Focused on language understanding, not on keywords-based approaches or text classification
- English only, not machine translation
- Does not cover speech recognition and synthesis
- Theoretical foundations, not available libraries to get things done

# Outline

- Introduction
- Linguistics
- Natural Language Processing/Understanding tasks
- Algorithms
- Datasets
- A few Modern Approaches, State of the Art, and Challenges
- Open-Source Libraries (List)
- Suggested Books and Websites

# Introduction

Moravec's Paradox

What and why of NLP/NLU

The Turing Test and other challenges

# Moravec's Paradox [1]

"it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility."

- Hans Moravec, 1988

The main lesson of thirty-five years of AI research is that the hard problems are easy and the easy problems are hard.

- Steven Pinker, The Language Instinct

Marvin Minsky emphasizes that the most difficult human skills to reverse engineer are those that are *unconscious*. [1]

[1] Moravec's paradox. (2013, December 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 00:55, September 29, 2014, from http://en.wikipedia.org/w/index.php?title=Moravec%27s_paradox&oldid=588240388

# Natural vs. Artificial Languages

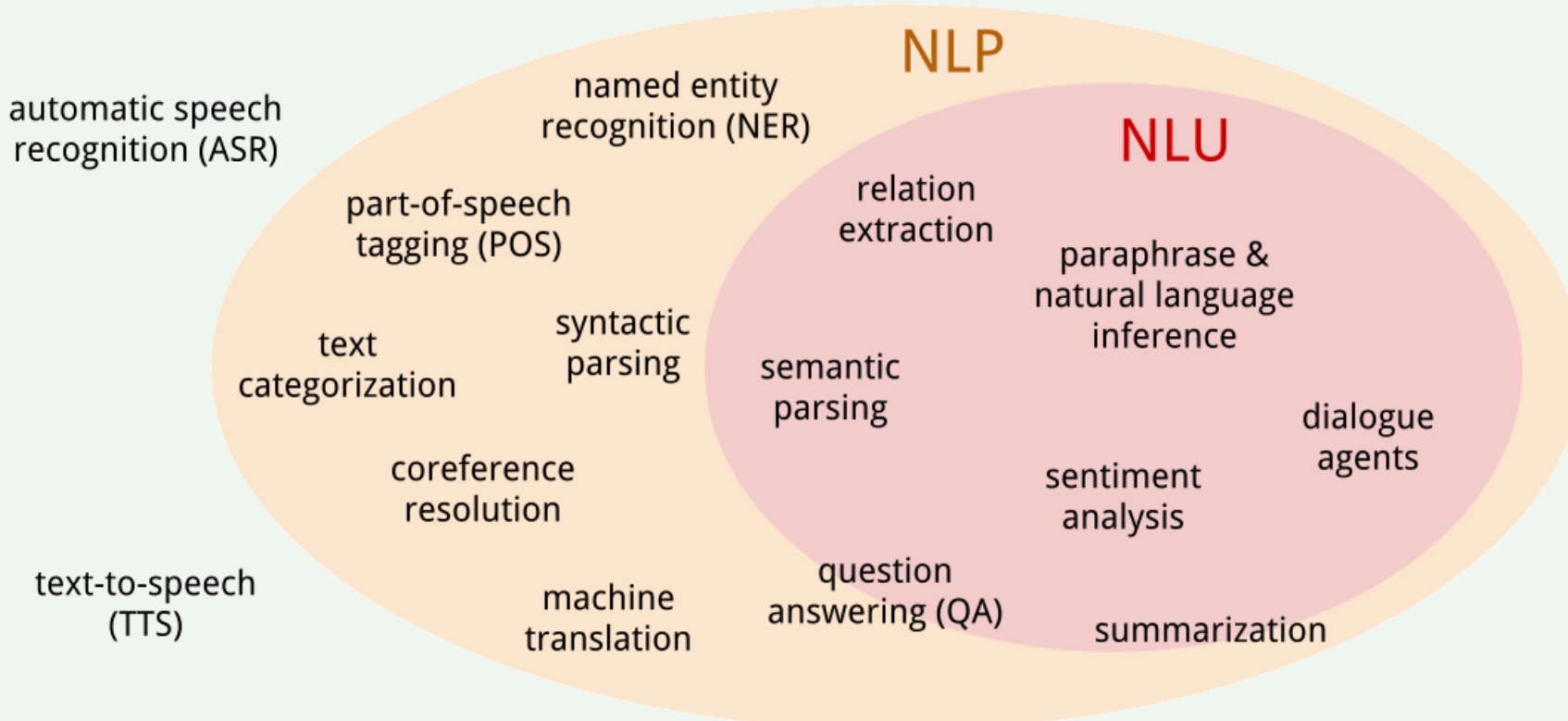
| Programming Languages (Manually designed for machines) | Natural Languages (Evolved in humans) |
|---|---|
| Clear distinction between syntax, semantics, run-time, algorithm | Mixes syntax, semantics, logic/inference, context, emotions |
| Deterministic / Largely unambiguous, Rigid | Often highly ambiguous, Fault-tolerant, Imperfect |
| Standardized, ~Static | Evolving, Involves real-time learning |
| Written*/read | Also spoken** (Note: Pronunciation and intonation often relevant for language understanding) |
| One-way communication (human to machine) | Centered on dialog (at least evolved around it) |
| Significantly smaller vocabulary and grammar | Large expanding vocabulary; Grammar huge |

* By humans ** Many have never been written at all. See Introducing Linguistics, Trask and Mayblin, page 84

# Definitions and Acronyms

NLP: Natural Language Processing

NLU: Natural Language Understanding




Thanks to Prof. Bill MacCartney [1] for the slide [2]. Reproduced with permission.

[1] http://nlp.stanford.edu/~wcmac/

[2] http://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf

# Application Categories

- Natural man-machine communication;  Dialog Systems
  - Reduce digital divide [1]
- Augment human knowledge
  - Like mechanical engineering augmented our physical bodies
  - Question answering,  information retrieval from unstructured text
- Human-human communication: Machine translation
- Social / behavioral analytics
  - Consumer sentiment,  Stock market prediction from news, etc.
- Reduce information overload
  - E.g. text summarization, news personalization, spam reduction
- Many others:
  - Editorial / Text Proofing / Spellings & Grammar Checking
  - Education / Answer Grading / Essay Scoring

[1] Digital divide. (2014, September 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 02:42, September 29, 2014, from http://en.wikipedia.org/w/index.php?title=Digital_divide&oldid=624677131

# Just Some Examples

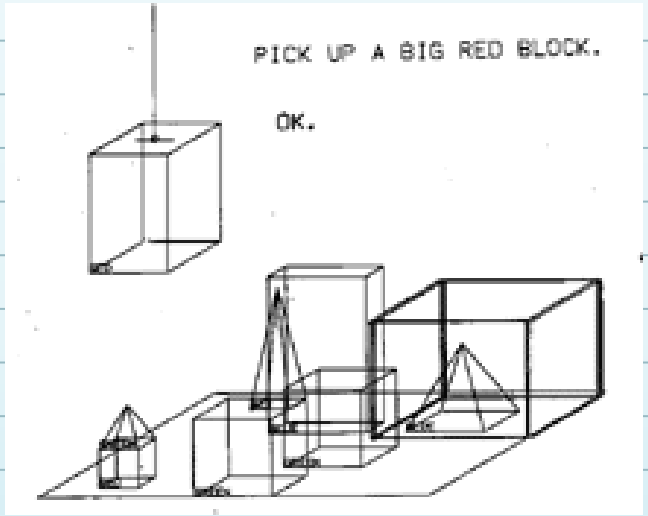
- Intelligent Personal Assistants [1]
  - SIRI, GoogleNow!, Cortana, et al.
- IBM Watson
- Wolfram Alpha
- Facebook Graph Search
- SHRDLU [2], et al.
- Chatterbots [3]:
  - Alicebot [4], Jabberwacky [5], Eugene Goostman [6], etc.
- MIT START [7]
  - "the world's first Web-based question answering system, … on-line and continuously operating since December, 1993"

[1] http://en.wikipedia.org/wiki/Intelligent_personal_assistant
[2] http://hci.stanford.edu/~winograd/shrdlu/
[3] http://en.wikipedia.org/wiki/Chatterbot
[4] http://www.alicebot.org/, http://en.wikipedia.org/wiki/A.L.I.C.E.
[5] http://www.jabberwacky.com/, http://en.wikipedia.org/wiki/Jabberwacky
[6] http://en.wikipedia.org/wiki/Eugene_Goostman
[7] http://start.csail.mit.edu/index.php

# SHRDLU Conversation [1]

- "Written by Terry Winograd at the M.I.T. Artificial Intelligence Laboratory in 1968-70"
- "Carried on a simple dialog with a user, about a small world of objects"

| | |
|---|---|
| Person | PICK UP A BIG RED BLOCK. |
| Computer | OK. (does it) |
| ... | |
| Person | WHAT DOES THE BOX CONTAIN? |
| Computer | THE BLUE PYRAMID AND THE BLUE BLOCK. |
| Person | WHAT IS THE PYRAMID SUPPORTED BY? |
| Computer | THE BOX. |
| Person | HOW MANY BLOCKS ARE NOT IN THE BOX? |
| Computer | FOUR OF THEM. |
| ... | |
| Person | WHEN DID YOU PICK IT UP? |
| Computer | WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE. |
| Person | WHY? |
| Computer | TO GET RID OF IT. |
| ... | |
| Person | WHY DID YOU DO THAT? |
| Computer | BECAUSE YOU ASKED ME TO. |




[1] http://hci.stanford.edu/~winograd/shrdlu/

# Turing Test and Others

- The Turing Test and Loebner prize [1]

- The Winograd Schema Challenge [2, 3]

The trophy would not fit in the brown suitcase because it was too big. What was too big?
Answer 0: the trophy
Answer 1: the suitcase
Joan made sure to thank Susan for all the help she had given. Who had given the help?
Answer 0: Joan
Answer 1: Susan

"what we are after is a new type of Turing Test that has these desirable features:
- it involves the subject responding to a broad range of English sentences;
- native English-speaking adults can pass it easily;
- it can be administered and graded without expert judges;
- no less than with the original Turing Test …"

[1] http://en.wikipedia.org/wiki/Loebner_Prize
[2] http://www.cs.toronto.edu/~hector/Papers/winograd.pdf, 2011
[3] http://www.kurzweilai.net/an-alternative-to-the-turing-test-winograd-schema-challenge-annual-competition-announced

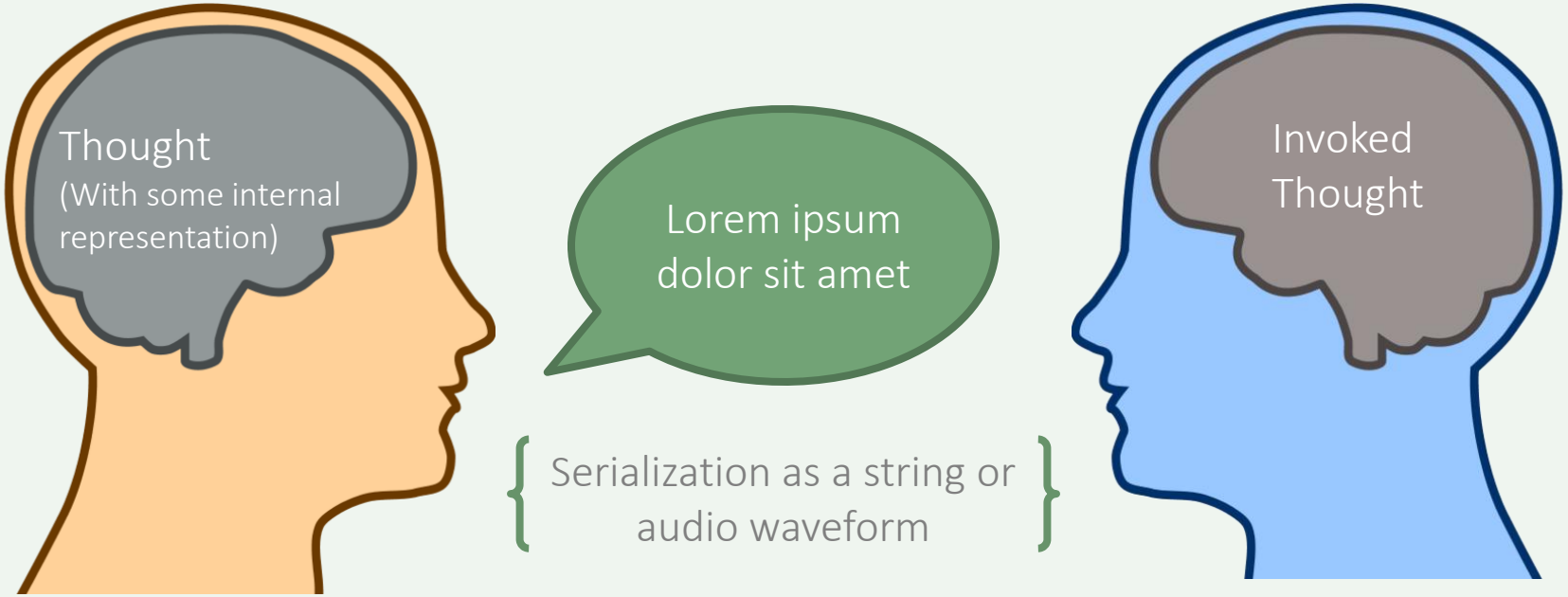
# Linguistics

Language as representation of thoughts

Words and parts of speech hierarchy

Grammar, Sentence forms

# Language as Serialization/Deserialization




Language could have evolved using “Simultaneous minimization in the effort of both hearer and speaker” [1, 2]

“Words and rules … work by different principles, are learned and used in different ways, and may even reside in different parts of the brain.” [3]

[1] Zipf's law. (2014, September 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 04:41, September 29, 2014, from http://en.wikipedia.org/w/index.php?title=Zipf%27s_law&oldid=626376322
[2] Ramon Ferrer i Cancho and Ricard V. Sole (2003). "Least effort and the origins of scaling in human language". Proceedings of the National Academy of Sciences of the United States of America 100 (3): 788–791
[3] Steven Pinker, Words and Rules, The Ingredients of Language, page 2

# Impact of Effort Minimization

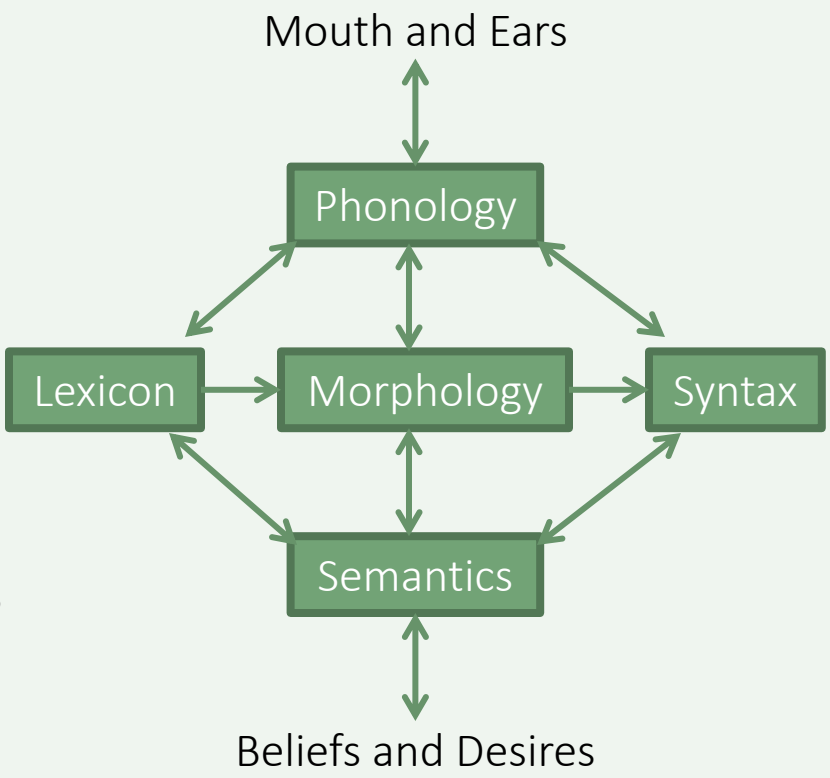
- Compression at multiple levels
  - Examples:
    - More frequent concepts have simpler words
      - "the" vs. "encyclopedia"
      - Adjustments as usage frequency changes, e.g. acronyms
    - Reuse:
      - Multiple meanings for same words, specially simpler ones: E.g. ~57 different senses for "run" → Ambiguity
      - Morphology: bump*ed*, rock*ed*, hack*ed*
- Redundancy
  - E.g. Most alphabet combinations not meaningful
  - Too much mental effort for a perfect language (canonical + precise)

Conflicting goals!

(Consider speaker and listener separately and together.)

# Kinds of Knowledge of Language [1]

- Morphemes – Meaningful components of words
  - lemmas (e.g. walk), affixes (e.g., -ing)
  - Morphology - How morphemes are combined. E.g. make + ing = making
- Listemes – Memorized chunks
  - Morphemes, idioms, collocations, …
- Phonetics and phonology
  - Linguistic sounds
- Orthography – Writing
  - Spellings, capitalization, punctuation, …
- Syntax
  - Grammatical relationships between words
- Semantics – Meaning
- Pragmatics
  - Goals and intentions of the speaker
- Discourse
  - Knowledge beyond sentence-level




Words and Rules, Steven Pinker, p 23

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 1.1

# Parts of Speech and Vocabulary

- Parts of speech (POS)
  - Syntactical or morphological category of a word
  - Nouns, pronouns, verbs, adjectives, adverbs, conjunctions, prepositions, determiners, particles, interjections, numerals, negatives, …
- Open and closed-class words
  - Open: New words keep getting added to the language: Nouns, verbs, adverbs, adjectives
    - E.g. I logged on to the web with my laptop … to find her home page. [1]
  - Closed: Nearly fixed set of words: Prepositions, conjunctions, determiners, etc. Also called function words.
- Dictionary size
  - WordNet: ~155K words (open-class only)
  - Wiktionary: 3.7 million entries for English as of Oct 2014 [2]
  - Heap's Law: Vocabulary size grows with at least the square root of the number of tokens in the corpus, i.e., $V > \mathbf{O}(\sqrt{N})$ [3, 4, 5]


[1] Introducing Linguistics, Trask and Mayblin, page 88
[2] http://en.wikipedia.org/wiki/Wiktionary
[3] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 4.1
[4] http://en.wikipedia.org/wiki/Heaps%27_law
[5] http://www.ccs.neu.edu/home/jaa/CSG339.06F/Lectures/text_stats.pdf

# Verbs (1)

- Types [1]:
  - Main verbs: eat, sleep — I eat
  - Modal: will, should — I will eat
  - Primary: be, have, do — I will be eating
- Regular and Irregular Verbs
  - Regular: walk, walks, walked, walking
  - Irregular: make, makes, *made*, making
    - Wikipedia lists 225 irregular verbs + additional 360 prefixed forms (e.g. write → cowrite) [2]
- Phrasal verbs: take down, take off, pull off
- Five inflected forms [3]: go, goes, went, gone, going
  - Be has eight different forms: be, am, are, is, was, were, being, been
- Can change according to:
  - Tense/Aspect — walk, walking, had been walking
  - Singular/plural — John and I walk, John walks
  - First/second/third person — I walk, You walk, She walks
  - Gender agreement (in some languages)
  - See [1, 3, 4] for others
- Orthography:
  - carry, carrying, carried  agree, agreeing, agreed  shop, shopping

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Sections 3.1.1 and 12.3.4
[2] http://en.wikipedia.org/wiki/List_of_English_irregular_verbs
[3] http://en.wikipedia.org/wiki/Uses_of_English_verb_forms
[4] Steven Pinker, Words and Rules, The Ingredients of Language, page 30 onwards

# Verbs (2)

- Valency [1, 2]: Number of "arguments" of verbs including subject
  - 1: He disappeared.
  - 2: He found a flight. ~~He disappeared a flight.~~
  - 3: He gave a watch to John.
  - 4: He sold the watch to John for $40.
  - 4: I bet you $10 it will rain
- Frames
  - Rules for verbs and their mandatory/optional arguments
  - E.g. [3]: Frame: Wagering
    - Definition: A Gambler commits an Asset to a prediction that an Uncertain situation will have a particular Outcome (or class of outcomes). He or she loses the Asset if the prediction ends up being incorrect, and gains it back plus additional winnings if the prediction ends up correct.
    - He bet $30 on the horse race.

[1] http://en.wikipedia.org/wiki/Verb#Valency
[2] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 12.3.5
[3] https://framenet2.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Wagering

# Morphology

- Words are built from smaller units called morphemes
  - Example: walk + ing
- Classes of morphemes: Stems and affixes
  - Prefixes: un-wanted
  - Suffixes: want-ed
  - Infixes: cupful -> cupsful
- Combining morphemes [1]:
  - Inflection: walk -> walks, walking
  - Derivation walk -> walker (meaning changes)
  - Compounding biochemistry, boardwalk
  - Cliticization We will -> We'll
- Stemming: Removing affixes
  - walking -> walk making -> make noisy -> noise
  - Lexicon-free (e.g. Porter Stemmer) and lexicon-based
    - icy -> ice ~~policy -> police~~

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed

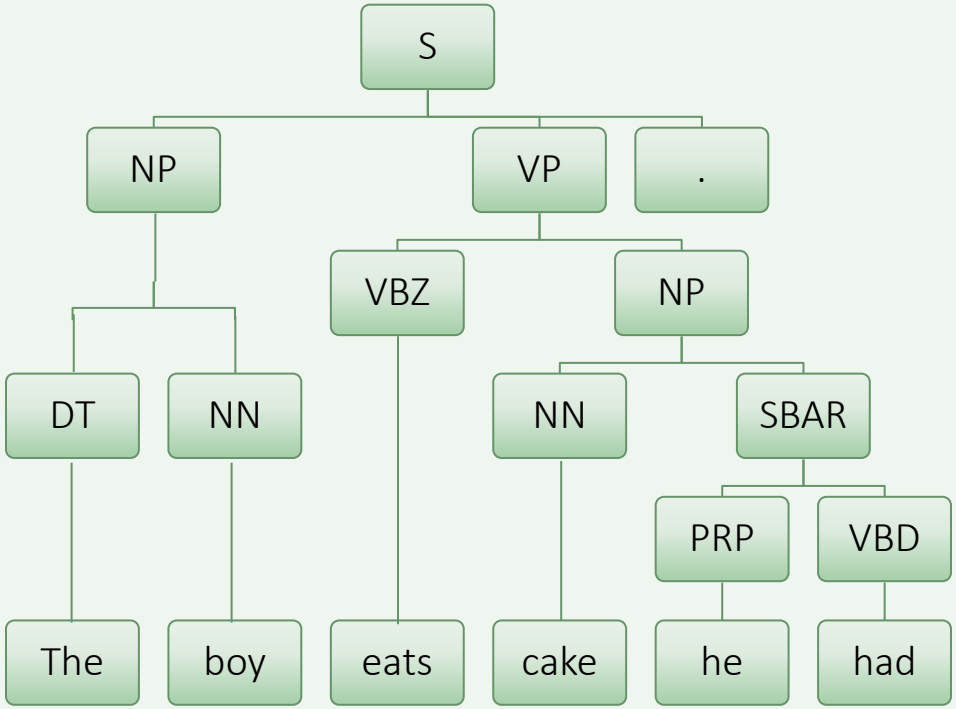
# Rules/Grammars (1)

- Define how to combine listemes into phrases and sentences
- Allow new constructions never spoken/written before
- At several levels: Morphology, Orthography, Syntax, Semantics, …
- Abstract
  - Re-use across morphemes, POS, subjects/objects. E.g.: {Noun Phrase} works
- Combinatorial and Recursive
  - John went outside. She told me that John went outside.
  - He likes apples, oranges, …, and bananas. {The list can be arbitrarily long.}
- Arbitrary (uses pure conventions)
  - E.g. Sentence constructions: Subject-Verb-Object, Subject-Object-Verb, etc.
    - Q: Which construction (SVO, SOV, …) is used by object-oriented programming languages? Subject.Action (Arguments) ⇒ SVO.
  - E.g. Future is ahead of us. Well, it is *behind* us in Greek! [2] {~Arbitrary coupling of time and space?!}
- And weird!
  - http://idibon.com/the-weirdest-languages/
  - E.g.: A primitive language uses different words for numbers when counting up and counting down! [TBD: Find and cite the reference]
- World Atlas of Language Structures: http://wals.info/

[1] Steven Pinker, Words and Rules, The Ingredients of Language
[2] Introducing Linguistics, Trask and Mayblin, page 54

# Rules/Grammars (2)

- A sentence is not a chain but a tree [1]




(Simplified)

| | |
|---|---|
| S | Sentence |
| NP | Noun Phrase |
| VP | Verb Phrase |
| DT | Determiner |
| NN | Noun (singular or mass) |
| VBZ | Verb (3sg pres) |
| PRP | Pronoun |
| etc. | |

<u>Tagsets</u>:

- Penn Treebank [2]: 45 tags
- Brown tagset: 87 tags
- Several others

- And a graph when the pronoun "he" is resolved to the boy.

[1] Steven Pinker, The Language Instinct, page 90
[2] http://web.mit.edu/6.863/www/PennTreebankTags.html

# Rules/Grammars (2)

- Constituency grammars
  - To parse sentences into constituents: noun phrases, verb phrases, etc.

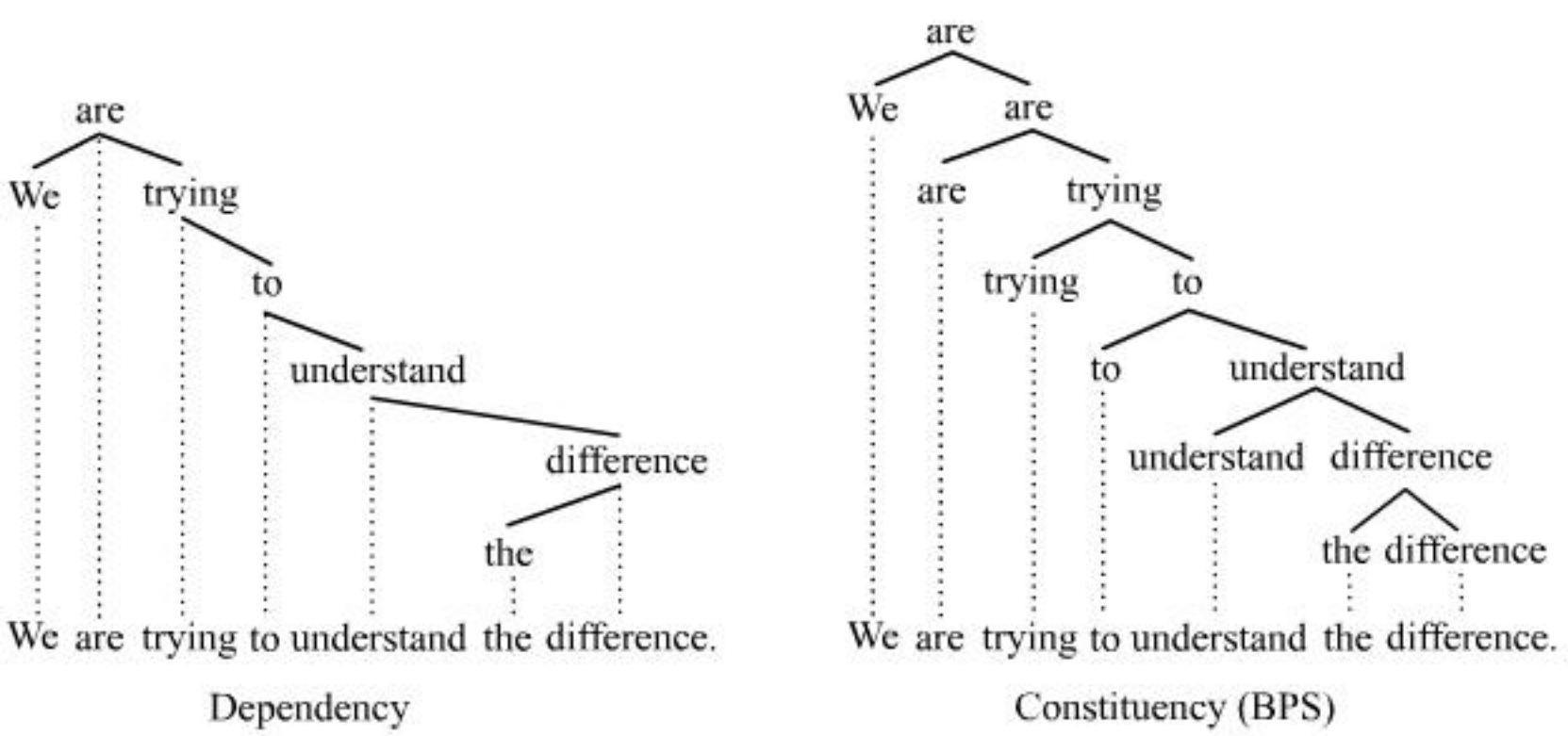
| | |
|---|---|
| • S → NP VP . | John eats the cake from Mary . |
| • VP → VBZ NP | John eats the cake from Mary . |
| • NP → NP PP | John eats the cake from Mary . |
| • NP → DT NN | John eats the cake from Mary . |
| • PP → IN NP | John eats the cake from Mary . |

- Rules are sometimes hand-crafted, often learned
  - Stanford Parser [1] uses ~46,000 binary rules (v2.0.2, 2012)

[1] http://nlp.stanford.edu/software/lex-parser.shtml

# Rules/Grammars (3)

- Dependency and Link Grammars [1]
  - Structure is determined by the relation between a word (a head) and its dependents. These handle free word order more easily.




[1] http://en.wikipedia.org/wiki/Dependency_grammar

# Rules/Grammars (4)

- Ambiguities
  - Attachment ambiguity [1]. E.g.: Prepositional Phrase attachment:
    - I killed an elephant in my pyjamas. OR …
    - I killed an elephant in my pyjamas.
      - A syntactic parser does not know elephants do not wear pyjamas!
  - Coordinating ambiguity [1]
    - Old men and women OR …
    - Old men and women
  - Natural languages are considered "embarrassingly" ambiguous. Not uncommon to have hundreds of parse trees for simple-looking sentences.
- Adjective Order [2, 3]
  - "green loose top" or "loose green top"
  - "… it is just a reminder that the invisible rules of language assert themselves everywhere, even when we break them, even when we don't think about them, even when they don't strike us as a great important big old deal." [2]

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 13.2
[2] http://www.slate.com/articles/arts/the_good_word/2014/08/the_study_of_adjective_order_and_gsssacpm.single.html
[3] http://english.stackexchange.com/questions/1155/what-is-the-rule-for-adjective-order

# Examples

- What a keywords-based algorithm sees:
  - what are word a the in in it letter at researcher n't matter to the the first only thing According be important Cambridge , does order right , a that University letters at and . is last the place
- What a purely syntax-based parser would see [1]:
  - VBG TO DT NN IN NNP NNP , PRP VBZ RB VB IN WDT NN DT NNS IN DT NN VBP , DT JJ JJ NN VBZ IN DT JJ CC JJ NN VBP IN DT JJ NN .
- With jumbled alphabet order (for humans) [2]:
  - Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

[1] Note: It would find it easier to read this than us though because this is what it is trained on.
[2] Sentence taken from: http://www.mrc-cbu.cam.ac.uk/personal/matt.davis/Cmabrigde/

# NLP / NLU Tasks

"Machine Reading"

Question Answering

Inference and Logic

...

# NLP / NLU Tasks (1)

- "Reading"
  - Sentence boundary detection and word tokenization detection
  - POS tagging, Stemming / morphological processing
  - Syntactic and Semantic Parsing
  - Word Sense Disambiguation
  - Named-Entity Recognition
  - Anaphora Resolution
  - Discourse analysis
- Question Answering
  - Wh- question parsing and categorization
  - Conversion to database queries

# NLP / NLU Tasks (2)

- Inference and Logic
- Language Synthesis
- Summarization
- Semantic Text Comparison and Scoring
- …

# Word and Sentence Tokenization

- Word Tokenization
  - Splitting on space and punctuation not enough!
    - Ph.D. 1,263.5 There's
    - book's over vs. book's cover
- Sentence Segmentation
  - Splitting on end-of-sentence markers again not enough
    - Period, question mark, exclamation
  - Period also used for:
    - Abbreviations: etc., m.p.h., A.D., Mr. J. Smith
    - Numbers: 23.5
    - Dates: 10.12.2014
    - Others: 127.0.0.1
  - Examples:
    - This test sentence has "etc." This test sentence has "etc." in it.
    - This sentence ends with you and I. This sentence ends with you and I. Smith.
    - "Ending quote is a part of this sentence." (Likewise for this.)
- Sequence of regular expressions, aided by abbreviations lexicon, works reasonably well.

# Word Sense Disambiguation (WSD)

- WordNet dictionary lists ~57 senses for the word "run" [1]
  - **Verb**: run, run an operation, argument runs as follows, run for treasurer, talent runs in the family, run a software executable, run an errand, interest rates run from 5 to 10 percent, …
  - **Noun**: scored 3 runs, experimental runs, half-mile run, his daily run keeps him fit, …
- WSD Task: Given a sentence, find intended senses for some/all ambiguous words.

[1] http://wordnetweb.princeton.edu/perl/webwn?s=run

# Named-Entity Recognition and Anaphora Resolution [1]

- Example:
  - Victoria Chen, Chief Financial Officer of Megabucks Banking Corp since 2004, saw her pay jump 20%, to $1.3 million, as the 37-year-old also became the Denver-based financial-services company's president. It has been ten years since she came to Megabucks from rival Lostabucks.
- Named-entity recognition (NER)
  - Victoria Chen, Megabucks Banking Corp, Denver, Lostabucks
- Anaphora Resolution: Multiple references to the same entity
  - Coreference resolution
    - Victoria Chen, the 37-year-old, Chief Financial Officer of Megabucks Banking Corp since 2004, her, the 37-year-old, the Denver-based financial-services company's president, she
    - Megabucks Banking Corp, the Denver-based financial-services company, Megabucks
  - Pronoun Resolution
    - her, she
    - What does "It" refer to? ☺

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 21.3

# Algorithms

Symbols, Strings and Chomsky's Hierarchy of Grammars

Sequence Labeling

Parsing Algorithms (Syntax, Semantics)

Word Sense Disambiguation

Inference

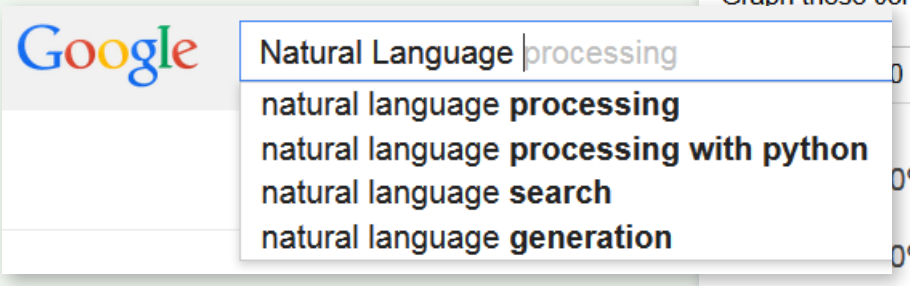
# Algorithm Classes in NLP [1]

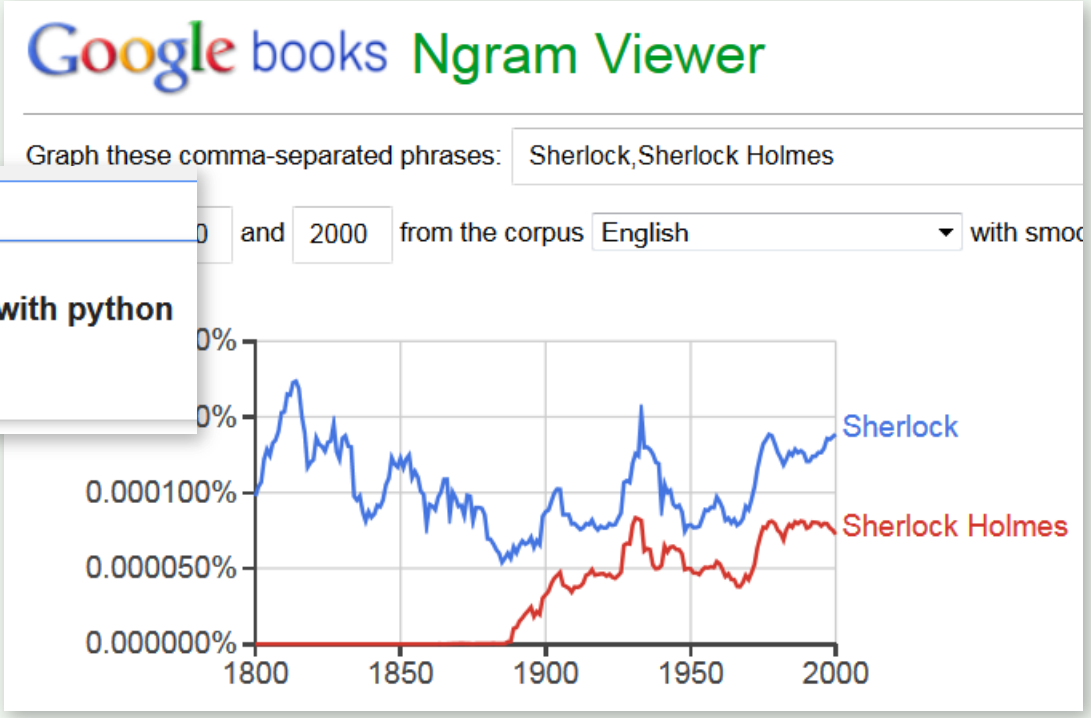
- State Machines and Grammars
  - Finite automata, push-down automata, etc.
  - Regular expressions, context-free grammars, etc.
- Logic and Inference
  - E.g.: First-order logic (Predicate calculus)
  - Search algorithms (depth-first, A* search, etc.)
- Probabilistic Models
  - E.g. Weighted Automation or Markov Models
  - Motivation: Resolve ambiguity by choosing most probable answer
    - Inherent ambiguity. E.g.: "old men and women"
    - Strengthen weaker algorithms with training data
- Machine Learning
  - Classifiers, sequence models, etc.
- Vector-Space Models [2]
  - Distributional models: Based on distribution contexts of words
    - Term-frequency / Inverse document frequency (TF-IDF), etc.
  - Distributed models
    - Learned using neural network / machine learning (e.g. word2vec)

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 1.3
[2] Prof. Bill MacCartney, Understanding NLU, http://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf, Slide 50

# Sequence Modeling: N-Grams (1)

- Examples:




https://books.google.com/ngrams

- Definition:
  - Probabilities of all potential $N^{th}$ items in a sequence given previous N-1 items
  - The items can be phonemes, letters, words, Part-of-Speech, etc.

# N-Grams (2)

- Computing: Estimate probabilities from large enough known sequences
  - "Smoothing" is used to avoid assigning zero probabilities to unseen examples
- Applications
  - Speech recognition, optical character recognition (OCR), machine translation, spelling correction, language detection, etc.
  - Use N-grams to estimate "prior" probabilities.
    - E.g.: Speech recognition, "Sherlock Holmes" is more likely than "Sherlock Homes"
- Limitations
  - Cannot model long-range dependencies
  - Predictability saturates at about N = 5 or 6 for English words [1]
- Datasets
  - Google N-Grams: ~500 billion words from ~5.2 million books, multi-lingual
  - http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
- References:
  - http://www.cs.utexas.edu/~mooney/cs388/slides/ngrams.ppt
  - http://en.wikipedia.org/wiki/N-gram

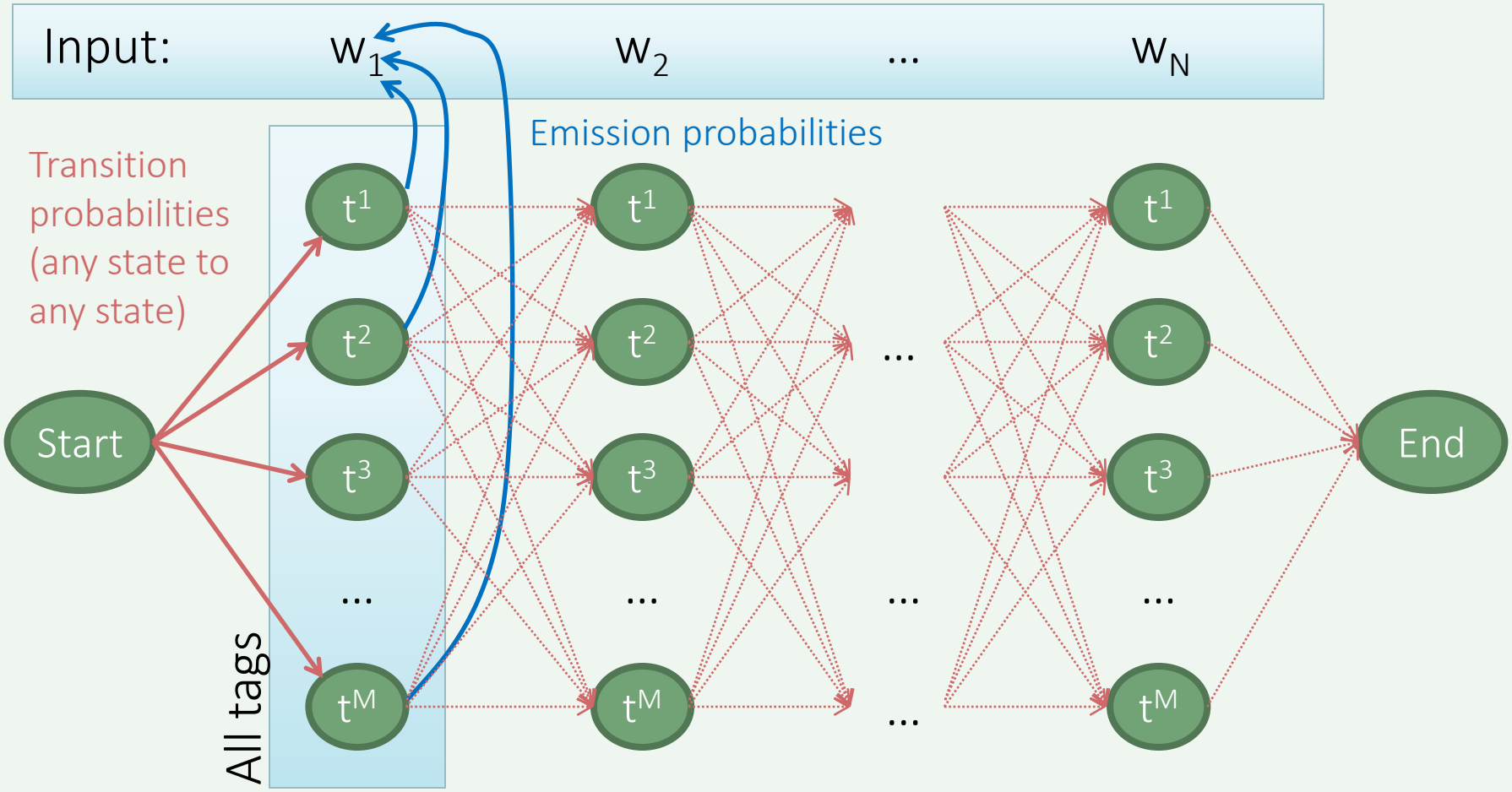
[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 4.9.4

# Sequence Labeling

- Example problems
  - Given words in a sentence (tokens) $w_1 \ldots w_N$, label the POS of each token, $t_1 \ldots t_N$.
  - Given sounds (phonemes), label the works spoken.
- Insight
  - With N-grams, we can estimate probability of Nth tag *given* previous N-1 tags. For sequence labeling, the previous N-1 tags would also be probabilistically known!
  - Goal: Maximize *overall* probability of the whole sequence.
- Hidden Markov Models [1]
  - Choose $t_1 t_2 \ldots t_N$ to maximize $P(t_1 \ldots t_N \mid w_1 \ldots w_N)$
  - Using Bayes Theorem, maximize: $P(w_1 \ldots w_N \mid t_1 \ldots t_N) * P(t_1 \ldots t_N)$
  - Approximate $P(t_1 \ldots t_N)$ using N-grams
  - Assume all words to be independent: $P(w_1 \ldots w_N \mid t_1 \ldots t_N) = \Pi\, P(w_k \mid t_k)$
- Training:
  - Given labeled data, calculate N-grams for output tags and emission probabilities $P(w_k \mid t_k)$.

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 5.5.2

# Hidden Markov Models




[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 5.5.3

# POS Tagging Accuracy [1]

- HMM Taggers (trained with labeled data) achieve ~97% accuracy
- Humans annotators themselves match only ~97% of the time!
  - Though agree 100% upon discussion
- Baseline: Treat each word independently; use most probable tag. Achieves ~94% [1]

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Sections 5.5 and 5.7

# Formal Language Theory (1)

- Formal language
  - Set of symbols (e.g. ASCII) and strings (e.g. sentences)
- Define patterns to divide all possible strings into two categories: Those that ____ and those that don't
  - Start with upper-case
  - Have matching parenthesis
  - Represent floating point numbers
  - Have all spellings correct
  - Form correct English grammar
  - That are *correct statements*
    - E.g.: Identify if the string encodes a correct proof of P≠NP [1].

[1] P≠NP: http://en.wikipedia.org/wiki/P_versus_NP_problem

# Chomsky's Grammar Hierarchy [1]

- Types of "productions" and their expressive powers
  - Type 4: Finite choice grammars. Only listed strings pass.
  - Type 3: Finite-state or regular grammars. Finite memory [2]
    - Cannot recognize recursive/nested structures in the input
  - Type 2: Context-Free Grammars (CFG)
    - Each rule describes a portion of input string independently of what its neighbors (describing other portions) do.
    - Uses single stack of memory of infinite length
    - Can handle some long-range dependencies, just leads to expansion of the number of rules
  - Type 1: Context-Sensitive or Monotonic Grammars
    - Can pass parameters across long-range dependencies
  - Type 0: Phase Structure Grammar. Unrestricted production rules
    - Impossible to have generic parser

[1] Dick Grune, Ceriel Jacobs, Parsing Techniques, A Practical Guide
[2] The input string may be of arbitrary length.

# Chomsky's Grammar Hierarchy (2)

- Dick Grune, Ceriel Jacobs, Parsing Techniques, A Practical Guide, 2nd Ed, Page 57

| Grammar | Expressive Power |
|---|---|
| Regular | Match lexically correct Java programs. E.g., No newlines inside strings, etc. |
| Context-Free | Match syntactically correct programs |
| Context-Sensitive | Match semantically correct programs. Those that will compile. |
| Phase Structure | Match programs that will finish in finite time |
| (Undecidable) | Match programs that will solve a given problem |

- Note: Cannot *correctly* implement beyond Regular on a computer with finite memory! [1] (Since out-of-memory error may happen)
  - Cannot even store a single integer correctly since it may be arbitrarily big!

[1] Note: Input string is allowed to be arbitrarily long (possibly infinite).

# Formal Language Grammars (3)

- Non-Chomskian Grammars [1]
  - Attribute and Affix Grammars — Most practical
  - Van Wijngaarden (VW) Grammars[2] — Most elegant
  - Boolean Grammars like PEG — Most promising
  - Tree-Adjoining Grammars
  - Etc.
- Grammars for natural languages are mostly Context-Free [3]
  - Verb-agreement handled using expanded set of rules or attaching parameters to non-terminals [4]

[1] Dick Grune, Ceriel Jacobs, Parsing Techniques, A Practical Guide, 2nd Ed, Chapter 15
[2] Cleaveland and Uzgalis, Grammars for Programming Languages
[3] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 16.3
[4] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 12.3.4

# Context-Free Parsers

CYK Parser

It’s Big-O Complexity

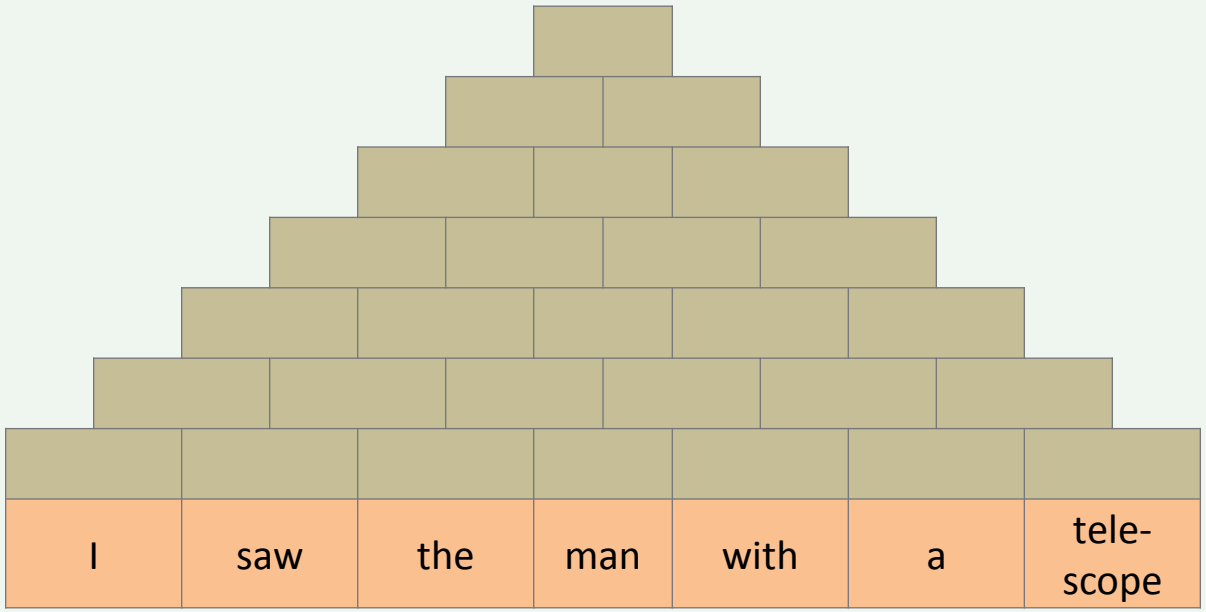
Online and offline parsing

Chart Parser

Semantic Grammars and Parsing

# Cocke-Younger-Kasami (CYK) Algorithm

- Dynamic programming algorithm; allocates a table

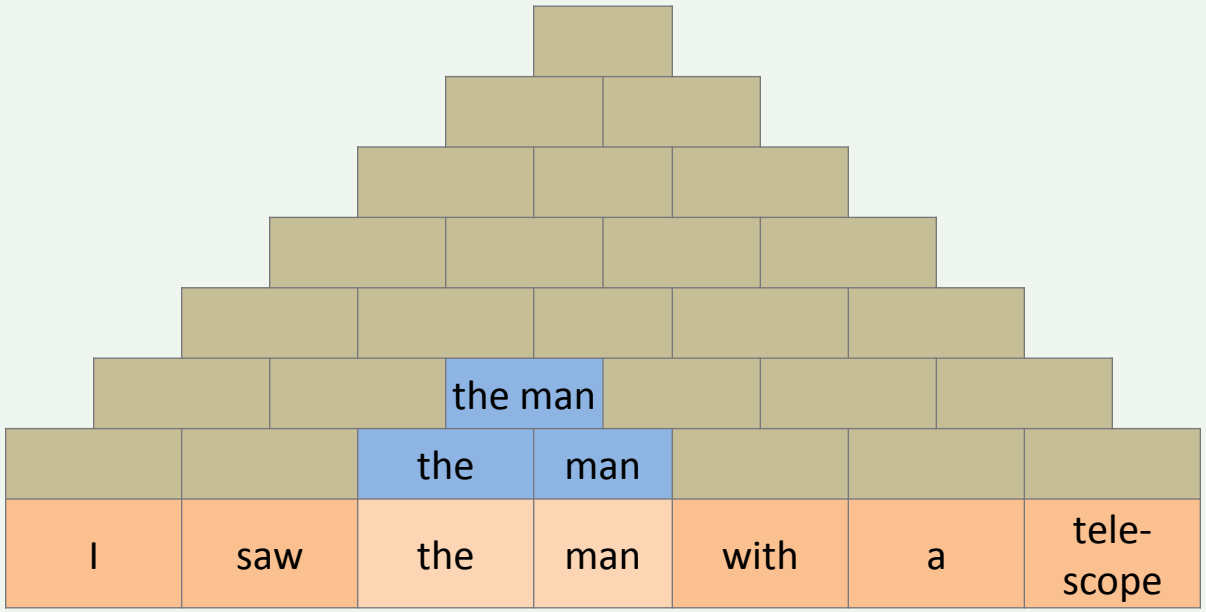



  - Note: The example is a bit simplified for demonstration.

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- What the boxes mean?

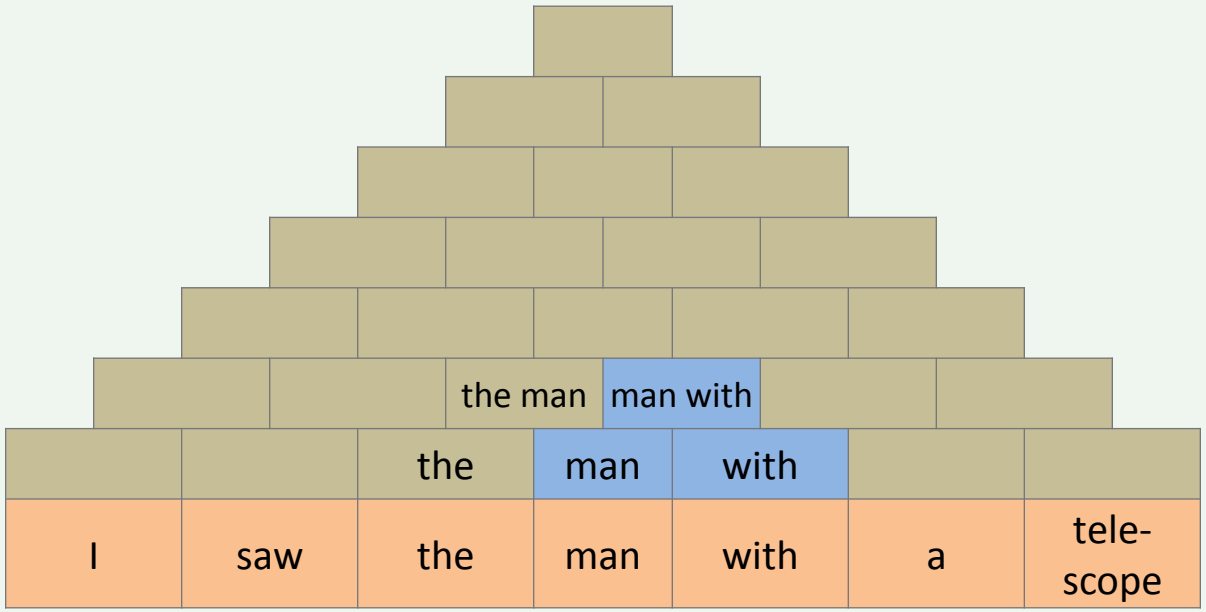



  - Each box covers portion of the input string diagonally under it.

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- Another example:

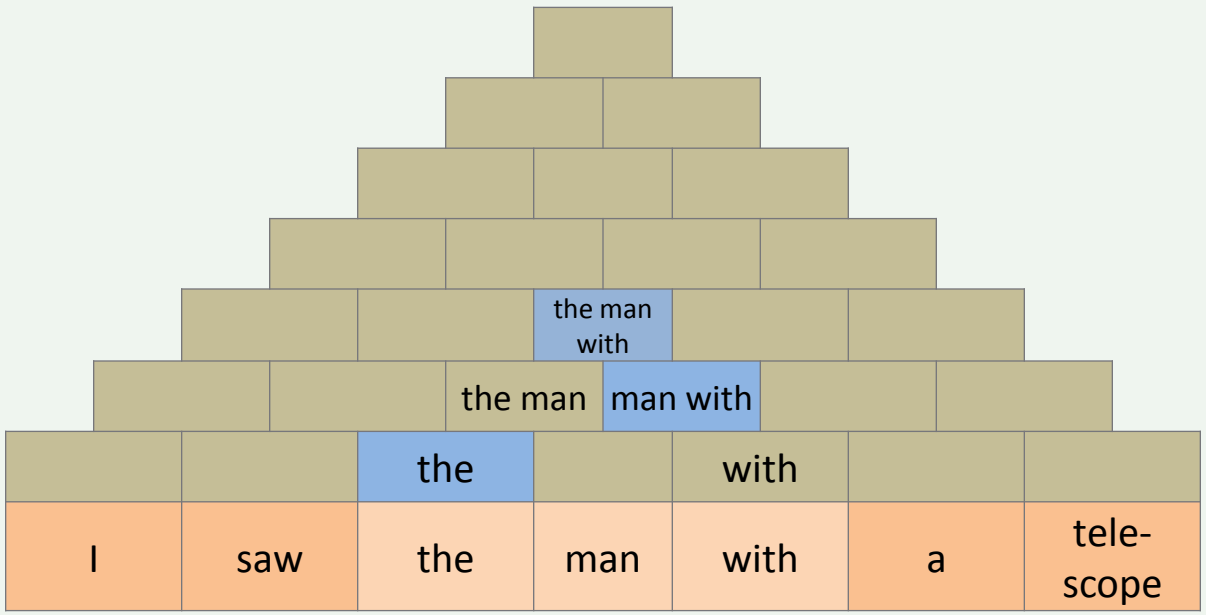



  - "man with" here is meaningless, would not actually fit any rule

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- How the boxes combine:

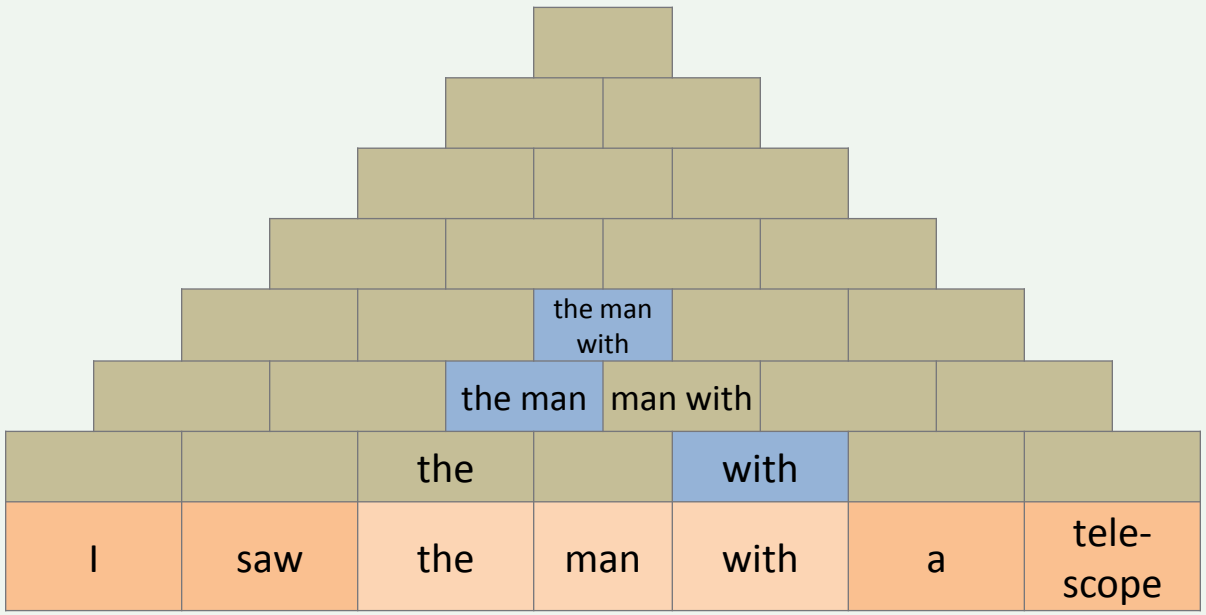



  - Pick adjacent, non-overlapping, input segments from below. “the” + “man with” in this example.

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- How the boxes combine:

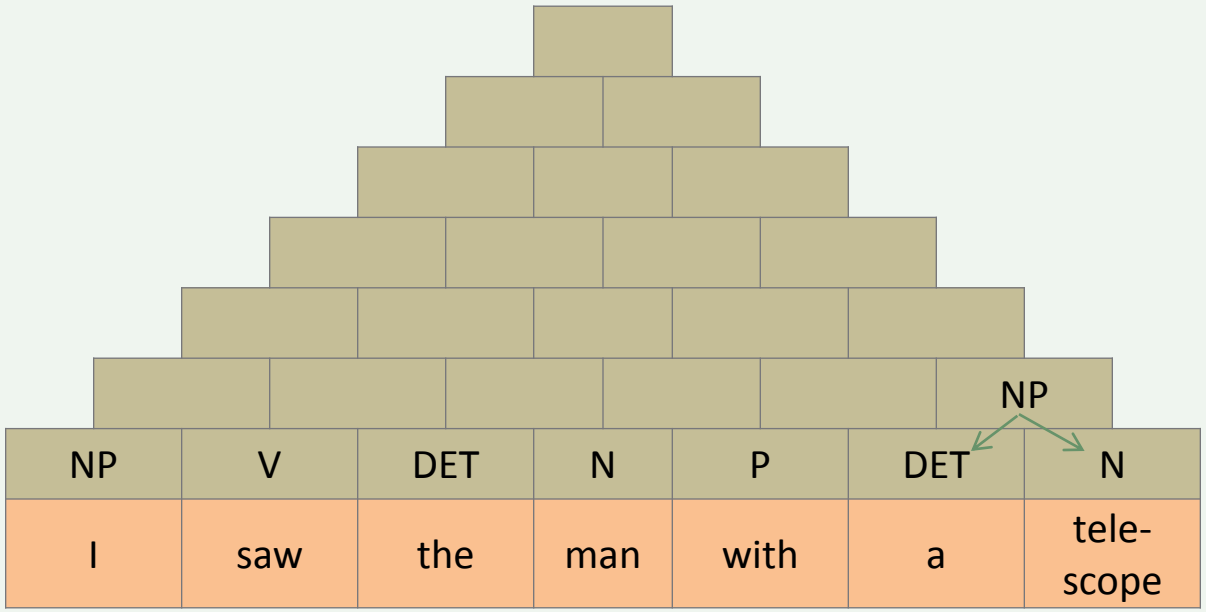



- Same box can also hold “the man” + “with”. Each box may thus have multiple entries. Scan is 3-D.

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- How are rules applied:

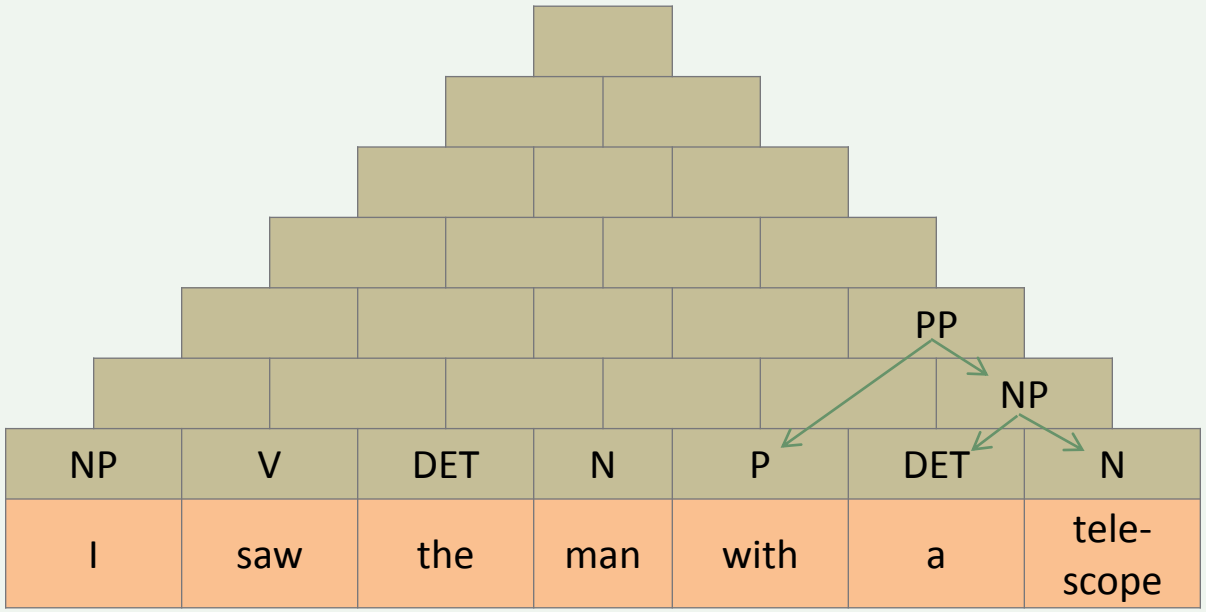



- All input combination possibilities for a box are tried. Here rule NP → DET N fits. Can index rules by RHS.

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- How are rules applied: Example 2

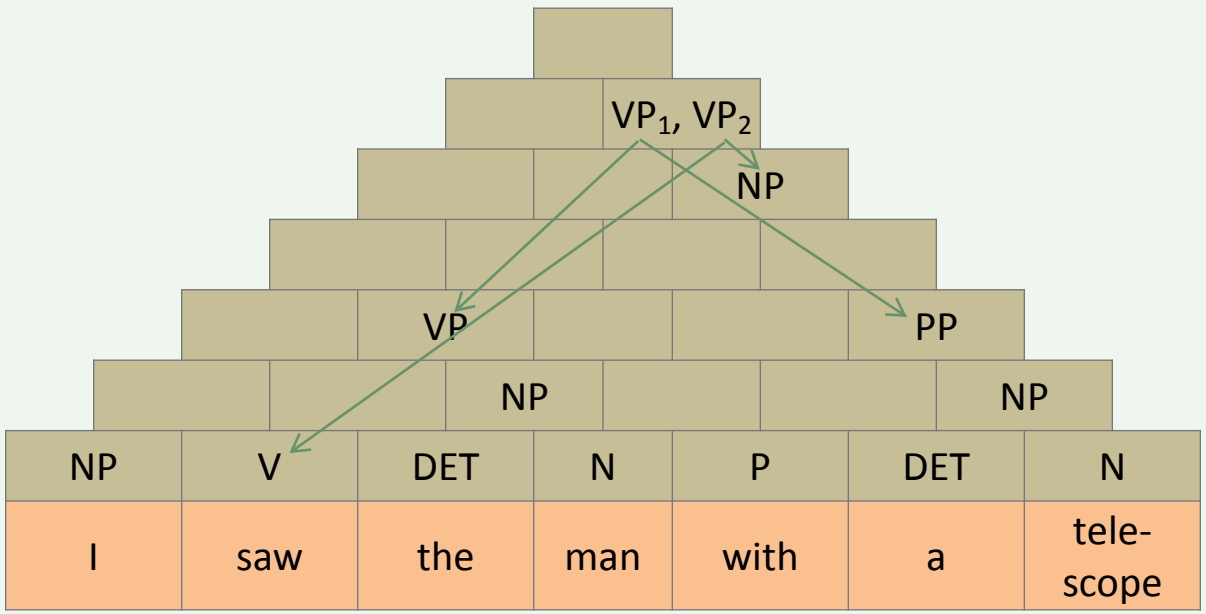



- Rule PP $\rightarrow$ P NP fits

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- Multiple rules may apply for the same 2-D box

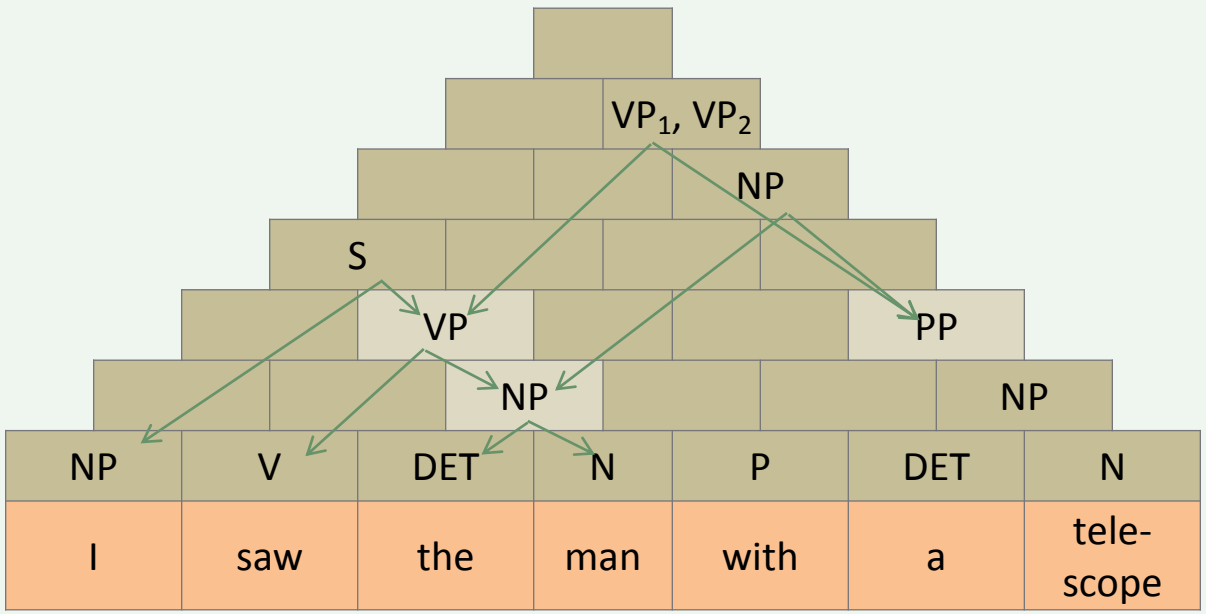



- Rule $VP_1 \rightarrow$ V NP fits: (saw) (the man with a telescope)
- Rule $VP_2 \rightarrow$ VP PP also fits: (saw the man) (with a telescope)

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

# Cocke-Younger-Kasami (CYK) Algorithm

- Multiple rules may *point to* same box

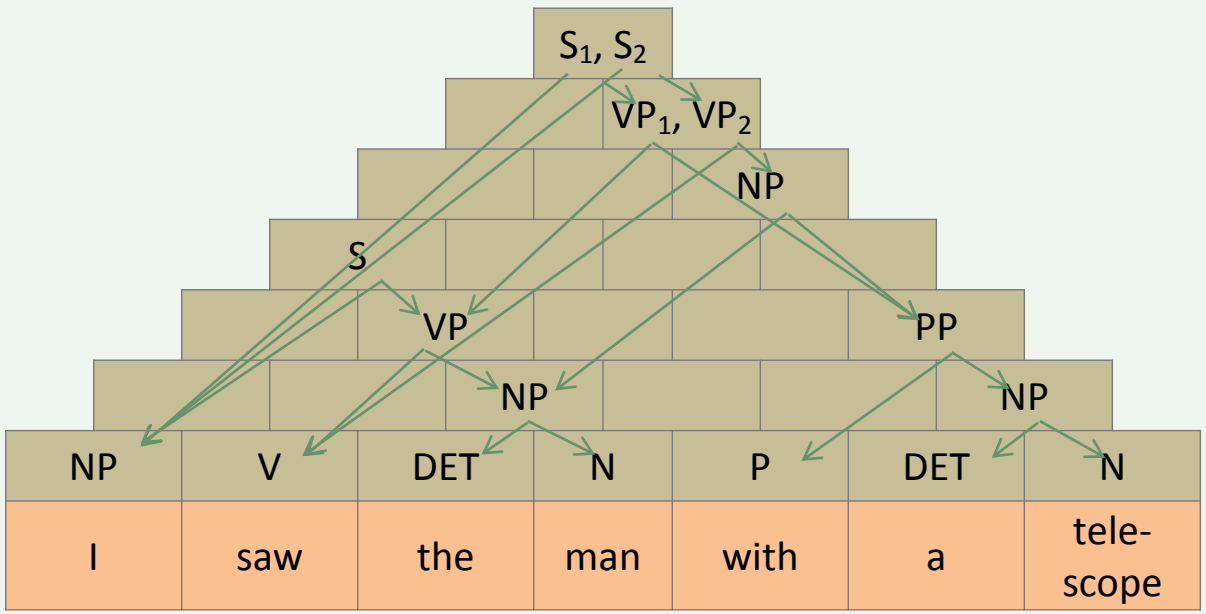



  - Must not duplicate the entry, else the algorithm becomes exponential! Referring boxes pointing to a box cannot peek underneath [2].

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html
[2] Additional or augmented tags can however be used to differentiate the constituents underneath.

# Cocke-Younger-Kasami (CYK) Algorithm

- Full parse forest (overlapping parse trees)




- The syntax-based algorithm could not distinguish which parse tree is correct. *Saw* with a telescope, or *the man with* the telescope.
- Cannot separate the trees! (Else the algorithm becomes exponential.)

[1] Example sentence taken from: http://www.diotavelli.net/people/void/demos/cky.html

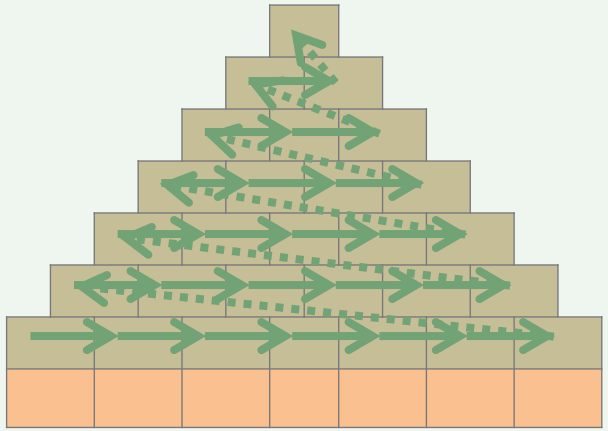
# Cocke-Younger-Kasami (CYK) Algorithm

- Big-O Complexity: **O**[$N^3G$] in the worst case
  - For sentence of length N => **O**[$N^3$] for the 3-D scan
  - Each box may fit any of the G binary rules
  - Lookup of rules is **O**[1] with a reverse index
- Context-Free parsing shown to be equivalent to Boolean matrix multiplication [1]
  - **O**[$N^3$] for naïve algorithm
  - **O**[$N^{2.81\ldots}$] for Strassen algorithm
  - **O**[$N^{2.37\ldots}$] for best-known algorithm [2]
  - However, the constant factor is very large, so not practical

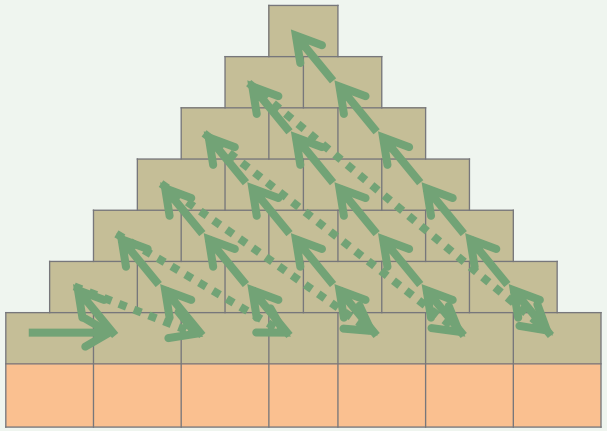
[1] Dick Grune, Ceriel Jacobs, Parsing Techniques, A Practical Guide, 2nd Ed, Section 3.10
[2] http://en.wikipedia.org/wiki/Coppersmith%E2%80%93Winograd_algorithm

# Chart Parser

- Online and offline variations:



Offline



Online

  - Offline: Whole sentence is available before parsing starts.
  - Online: Process the parse as far as possible reading just one token at a time.
- Chart parser: Use a separate plan for ordering

# Chomsky Normal Form (CNF)

- CYK parser requires the rules to be binary
  - S → NP VP .  Three symbols on the right, not OK.
  - VP → VBZ NP  Two symbols on the right, OK.
- Conversion of a general CFG grammar to one with binary rules*:
  - S → NP VP .  S → *NPVP* .
  - *NPVP* → NP VP
  - Must retain a backward index to convert the resulting parse tree to the original grammar.

* See Dick Grune, Ceriel Jacobs, Parsing Techniques, A Practical Guide, 2nd Ed, for other issues like ε-rules.

# Probabilistic and Lexicalized Parsers

- A purely syntax-driven parser cannot disambiguate between the many possible parse-trees
- Probabilistic CFG (PCFG)
  - Assigns a probability to each CFG grammar rule. Conversion to CNF adjusted accordingly.
  - Probability of a parse tree is product of all the rules used
  - Limitations:
    - Independence assumption. E.g. Subject NP is much more likely to use pronouns than Object NP.
    - Lack of lexical conditioning. E.g. Whether a prepositional phrase should attach to the noun phrase or verb phrase depends on the actual words involved (not just syntax)
  - Approach:
    - Split (and merge) non-terminals. E.g. Parent annotation [1]
    - Lexicalized Probabilistic CFGs: Include headwords into rules [2].

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Chapter 14
[2] Headword for "the tallest man on earth" would be "man".

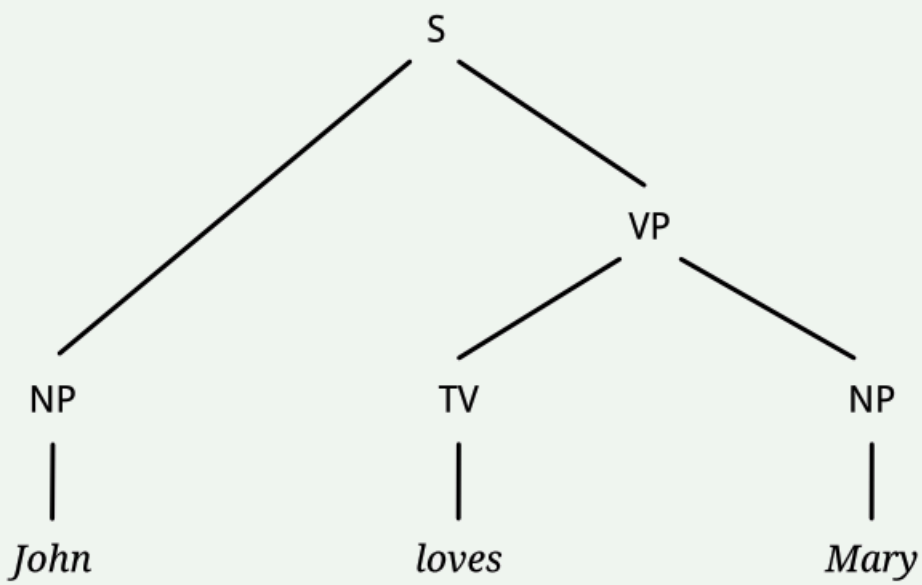
# Semantics and Inference (1)

- Meaning representation suitable for computational algorithms
- Predicate calculus (First-Order Logic) [1]
  - $\exists x$ . man (x) — There exists x such that x is a man
  - $\exists x$ . man (x) $\wedge$ walk (x) — -do-, and x walks
  - Or… A man walks
  - $\forall x$ . man (x) $\rightarrow$ walk (x) — For all x, if x is a man, then x walks
  - Or… Every man walks
  - walk (John) $\wedge$ walk (Mary) — John walks and Mary walks
  - Or… John and Mary walk
- Representing time, intervals, events [2]
  - Last Tuesday, John ate bread for lunch at my desk
    - $\exists e$ . Eating (e, John, bread, lunch, his desk) $\wedge$ Time (e, Tuesday)
  - Precedes (e, Now), etc.

[1] Examples from http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec09-intro-semparse.pdf, slides 23-4
[2] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Section 17.4

# Semantic Grammars and Parsing

- Syntactic parse:



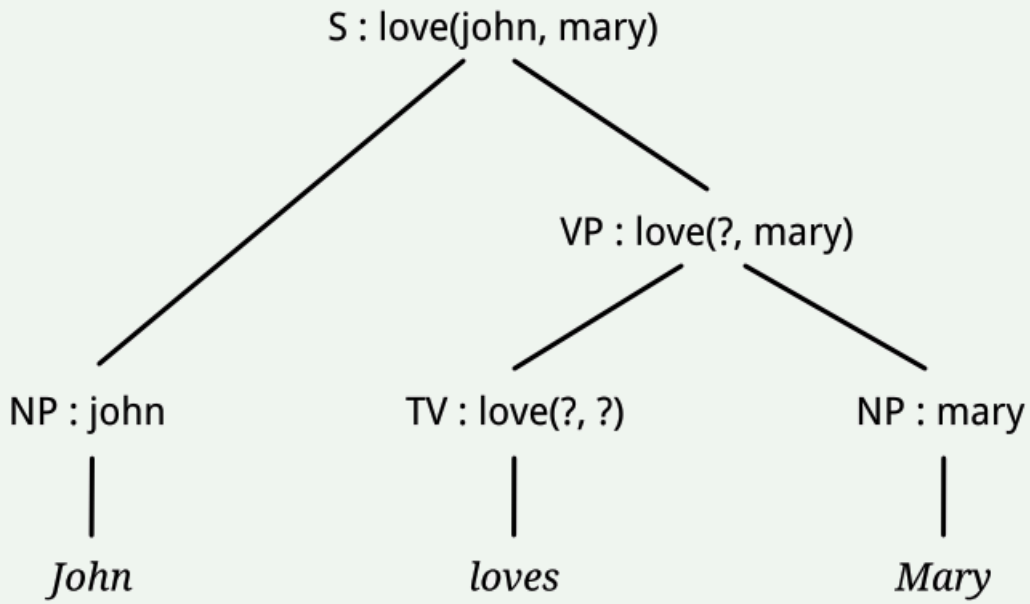

Thanks to Prof. Bill MacCartney [1] for the slide [2].

[1] http://nlp.stanford.edu/~wcmac/

[2] http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec09-intro-semparse.pdf

# Semantic Grammars and Parsing

- Semantic Annotations




Thanks to Prof. Bill MacCartney [1] for the slide [2].

[1] http://nlp.stanford.edu/~wcmac/

[2] http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec09-intro-semparse.pdf

# Semantic Grammars and Parsing

- Need semantic attachments with CFG rules
  - Lexicon
    - John ← NP: john
    - Mary ← NP: mary
    - loves ← TV: λy . λx . love (x, y)
  - Composition rules
    - VP: f(a) → TV: f NP: a
    - S: f(a) → NP: a VP: f

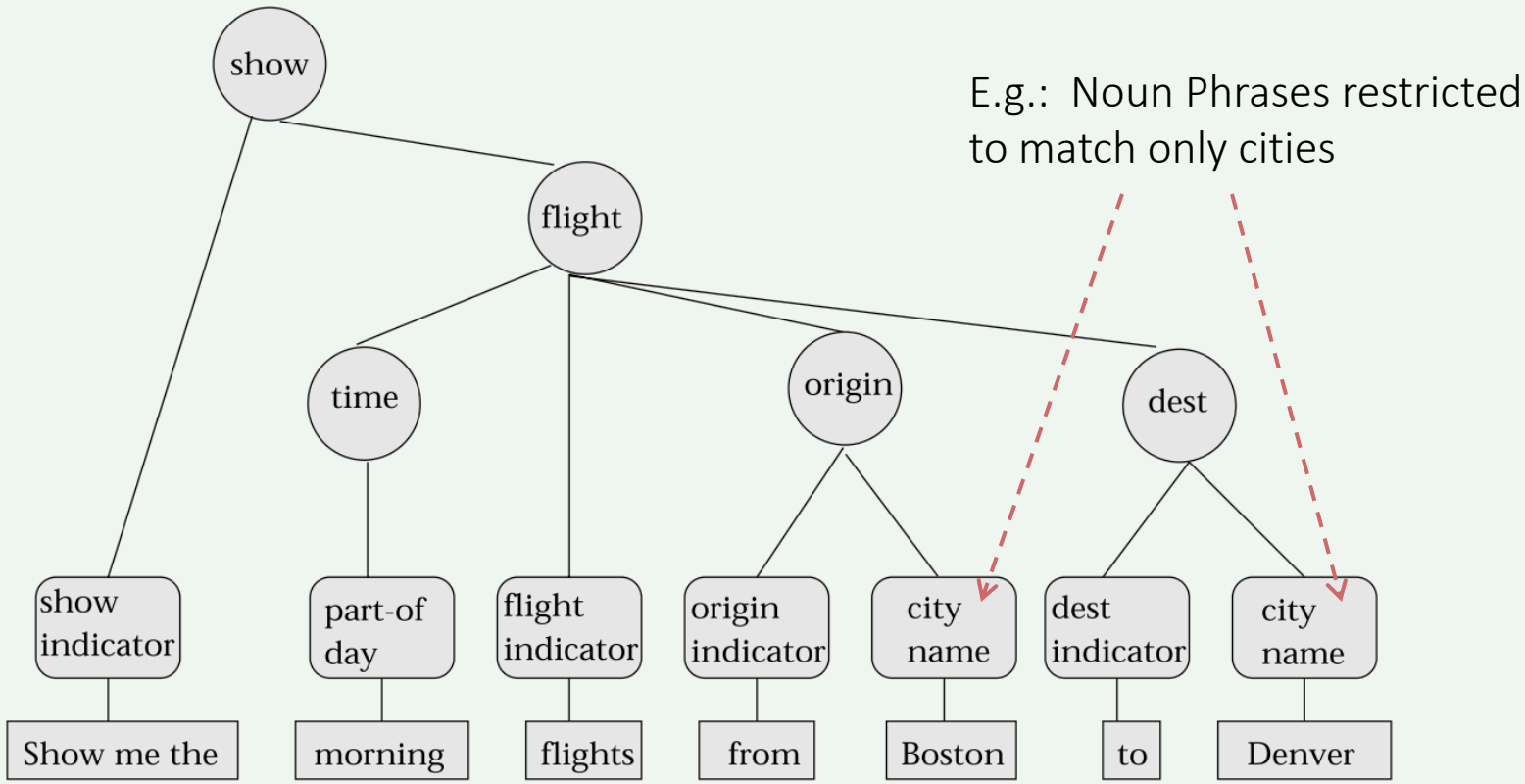
Thanks to Prof. Bill MacCartney [1] for the slide [2].

[1] http://nlp.stanford.edu/~wcmac/

[2] http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec09-intro-semparse.pdf

# Semantic Grammars and Parsing

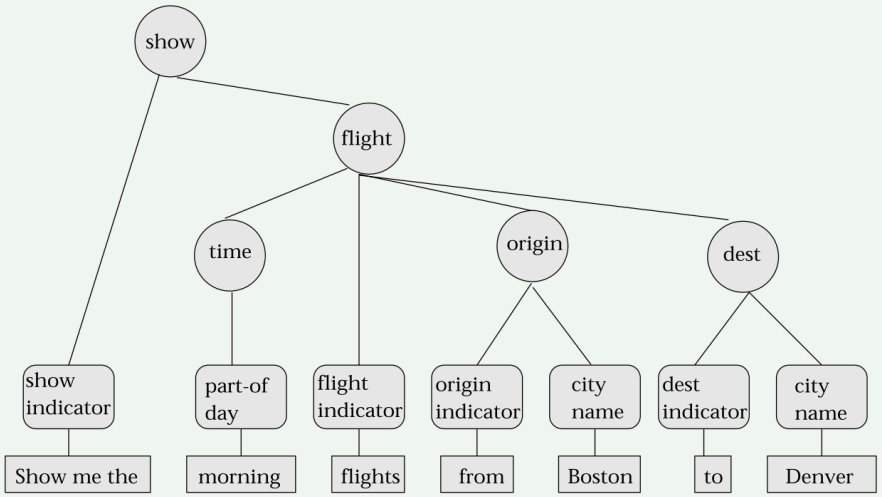
- With selectional restrictions:




[1] Corpus-Based Approaches to Semantic Interpretation in NLP, Hwee Tou Ng, John Zelle, http://www.aaai.org/ojs/index.php/aimagazine/article/view/1321, Figure 7

# Semantic Grammars and Parsing

- Conversion to database query or logic statements




FRAME: Air-Transportation
SHOW: (flight-information)
ORIGIN: (City ‘‘Boston”)
DEST: (City ‘‘Denver”)
TIME: (part-of-day ‘‘MORNING”)

[1] Corpus-Based Approaches to Semantic Interpretation in NLP, Hwee Tou Ng, John Zelle, http://www.aaai.org/ojs/index.php/aimagazine/article/view/1321, Figure 7 and 8

# Semantics and Inference (2)

- Inference [1]
  - $\forall x$ . smoke (x) $\rightarrow$ snore (x), smoke (John)
  - $\Rightarrow$ snore (John)
- Automated Theorem Proving
  - See excellent review by David A. Plaisted:
    - http://onlinelibrary.wiley.com/doi/10.1002/wcs.1269/pdf

[1] Examples from http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec09-intro-semparse.pdf, slides 23-4

# Word Sense Disambiguation (WSD)

- Baseline:
  - Assume most common sense of each word (available from training data)
- Simplified Lesk Algorithm [1]:
  - Consider surrounding words from the sentence: Context-window
    - Remove stop words (common words like “is”, “the”, etc.)
  - For all possible senses, consider words from their dictionary definitions
  - Sense with largest number of overlapping words is accepted
  - Variations:
    - Context-window [2]:
      - Ordered and unordered, with and without part-of-speech and morphology, with and without syntactic relations (subject-verb, etc.)
    - Senses:
      - Broaden using related senses (e.g. hypernyms from WordNet)
      - Include context words from training data (**Corpus Lesk** method)
    - Matching:
      - Weight overlapping words with their importance (e.g. TF/IDF scores [3])

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Sections 20.4
[2] Corpus-Based Approaches to Semantic Interpretation in NLP, Hwee Tou Ng, John Zelle, http://www.aaai.org/ojs/index.php/aimagazine/article/view/1321
[3] http://en.wikipedia.org/wiki/Tf%E2%80%93idf

# Word Sense Disambiguation (WSD)

- Yarowsky's method [2, 3]: Used boot-strapping to extend available training data
- Remarks:
  - Many other methods have been tried [2, 3]. E.g.: Train a separate classifier for each word, graph-based methods [4], etc.
  - If a whole article is to be disambiguated, using same sense for all occurrences of the same word increases accuracy.

[1] Speech and Language Processing, Jurafsky and Martin, 2nd Ed, Sections 20.4
[2] Corpus-Based Approaches to Semantic Interpretation in NLP, Hwee Tou Ng, John Zelle, http://www.aaai.org/ojs/index.php/aimagazine/article/view/1321
[3] Manning and Schutze, Foundations of Statistical Natural Language Processing, Chapter 7
[4] Eneko Agirre, David Martinez, Two graph-based algorithms for state-of-the-art WSD, http://hal.archives-ouvertes.fr/docs/00/08/05/12/PDF/emnlp.pdf

# WSD Performance

- SENSEVAL-3 [1]
  - Baseline: 55.2% (fine-grained), 64.5% (coarse grained)
  - The best system 72.9% (79.3%) for fine-grained (coarse-grained) scoring.
- Challenges [2]
  - A sense inventory cannot be task-independent
    - Note: Evaluation often performed at single sentence-level
  - Word meaning does not divide up into discrete senses
    - WordNet sometimes considered too fine-grained, even for humans

[1] Rada Mihalcea, et al, The SENSEVAL–3 English Lexical Sample Task, http://www.aclweb.org/anthology/W04-0807
[2] http://www.scholarpedia.org/article/Word_sense_disambiguation

# Natural Language Synthesis/Generation

- The inverse problem
  - Note: I currently have limited understanding in NLG
- Example from SimpleNLG Java library [1]

```
SPhraseSpec p = nlgFactory.createClause ();
p.setSubject ("Mary");
p.setVerb ("chase");
p.setObject ("the monkey");
System.out.println (realiser.realiseSentence (p)); // Prints "Mary chases the monkey."
p.setFeature (Feature.TENSE, Tense.FUTURE);
System.out.println (realiser.realiseSentence (p)); // Prints "Mary will chase the monkey."
```

  - Lexicon/morphology system: Computes inflected forms (morphological)
  - Realiser: Generates texts from a syntactic form
  - Microplanning: Currently just simple aggregation

[1] https://code.google.com/p/simplenlg/

# Natural Language Synthesis/Generation

- References:
  - Natural Language Generation: an introduction and open-ended review of the state of the art
    - http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/info-pages/nlg/ATG01/ATG01.html

# Datasets

Dictionaries

Parsing (Treebanks)

Word-Sense Disambiguation

Semantics

Knowledge

# Select Datasets

- Dictionaries
  - WordNet: Lexical database of open-class words, ~155K words in ~117K synsets
    - Synset: Set of one or more synonyms that are semantically equivalent
      - E.g.: {detect, observe, find, discover, notice} "She detected high levels of …"
      - Many-to-many mappings from words to synsets
    - Relationships between synsets: Antonyms, Hypernyms, Meronyms, etc.
      - E.g.: {detect, observe, find, discover, notice} → … → {perceive, comprehend}
    - http://wordnet.princeton.edu/
    - Extensions: BabelNet, SentiWordNet, etc.
  - Wiktionary: Wiki + Dictionary, ~3.8 million entries for English
- Treebanks for Parsing: Parse-trees
  - http://en.wikipedia.org/wiki/Treebank
  - PennTreeBank (PTB) III: ~1 million words, ~17K grammar rules
- Word Sense Disambiguation: SemCor 3.0
  - 352 articles, 37176 sentences, 676546 tokens, tagged 226036
- Semantics: FrameNet, VerbNet, Propbank, Cyc, ConceptNet
- Facts: Dbpedia, Freebase, et al
- See also: Prof. Rada Mihalcea's website:
  - http://web.eecs.umich.edu/~mihalcea/downloads.html

# Some Modern Approaches, State of the Art and Challenges

Select Modern Approaches

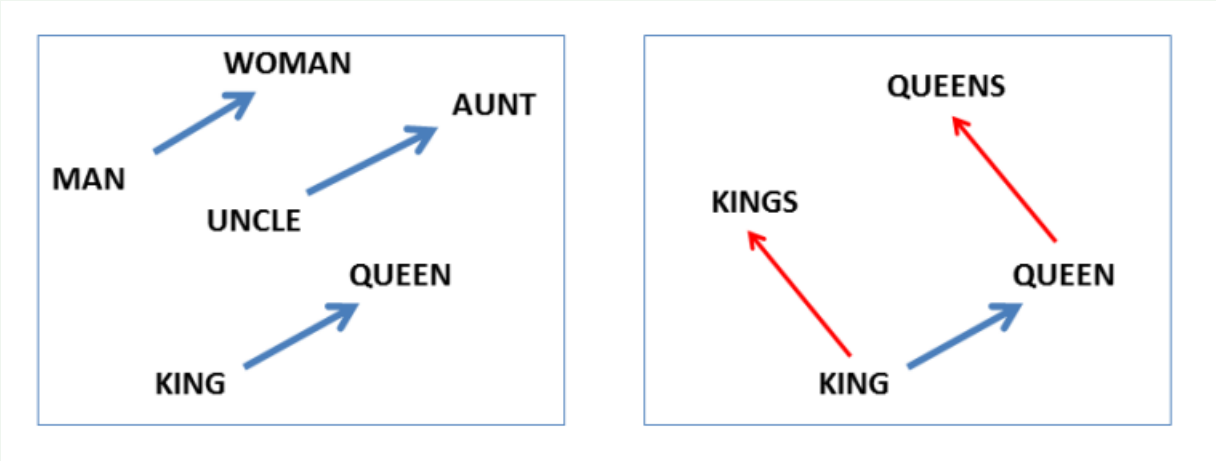
State of the Art

Language abilities in humans

Fundamental Challenges for NLP/NLU

# Select Modern Approaches

- Distributed Vector-Space Models
  - Word to vector: https://code.google.com/p/word2vec/



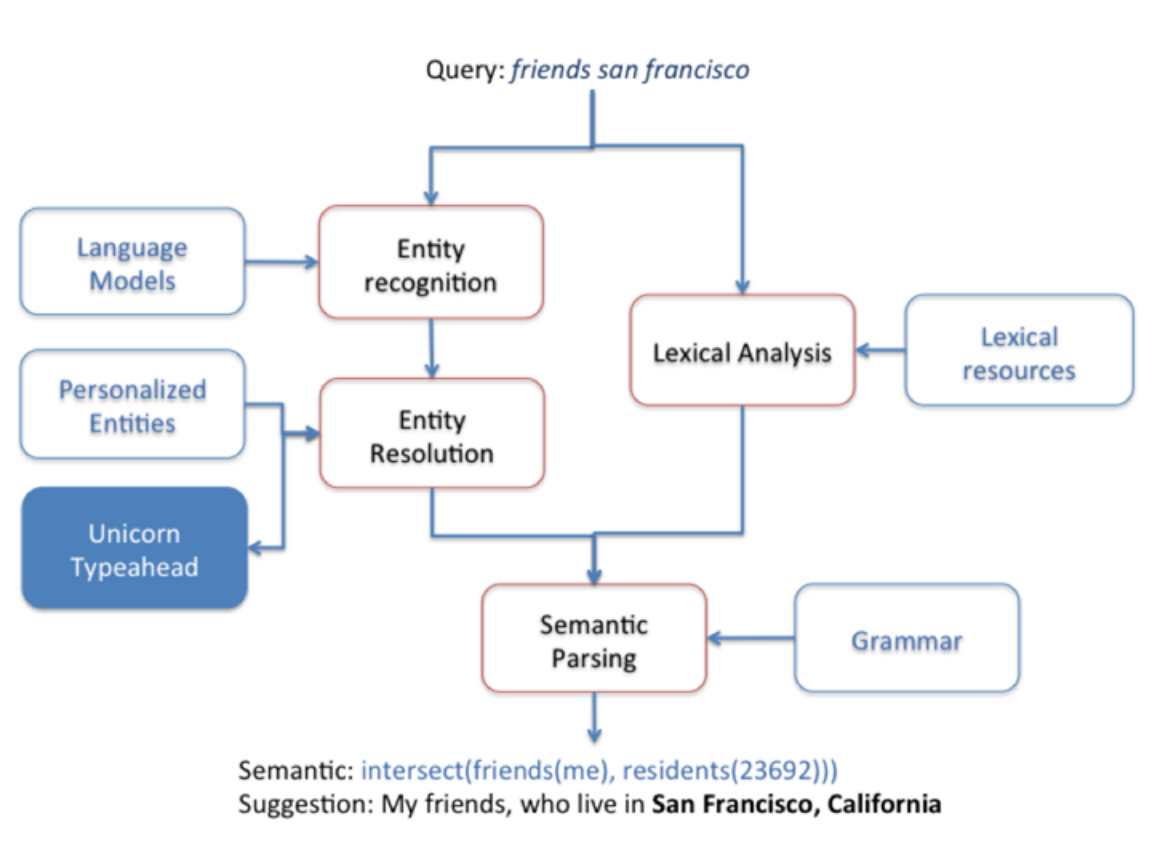

[1]

  - Excellent overview of why it works:
    - Christopher Olah, Deep Learning, NLP, and Representations:
      - http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/
  - GloVe: http://nlp.stanford.edu/projects/glove/

[1] Tomas Mikolov et al, Linguistic Regularities in Continuous Space Word Representations, http://research.microsoft.com/pubs/189726/rvecs.pdf

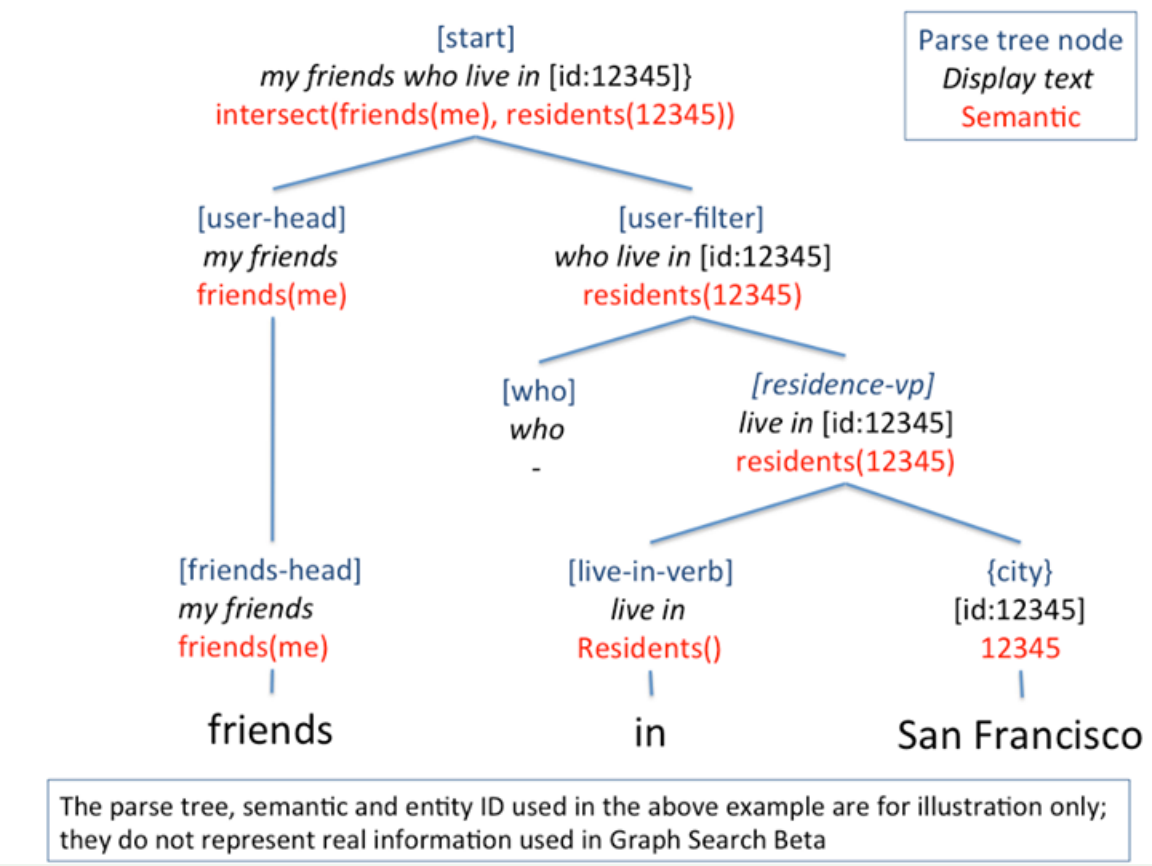
# State of the Art: Examples

- Facebook Graph Search
  - https://www.facebook.com/notes/facebook-engineering/under-the-hood-the-natural-language-interface-of-graph-search/10151432733048920



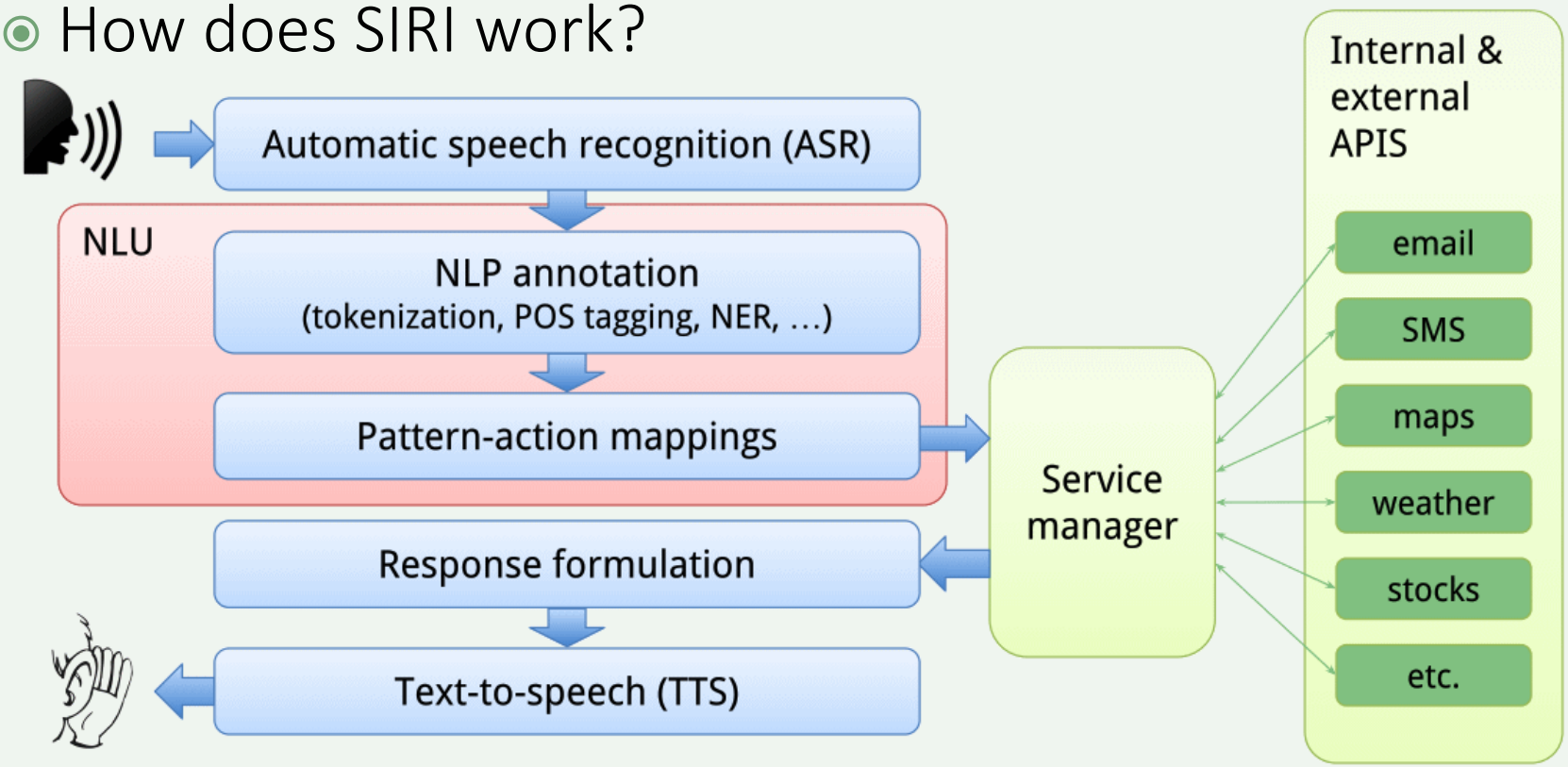
# State of the Art: Examples

- Facebook Graph Search
  - https://www.facebook.com/notes/facebook-engineering/under-the-hood-the-natural-language-interface-of-graph-search/10151432733048920



# State of the Art: Examples

- How does SIRI work?

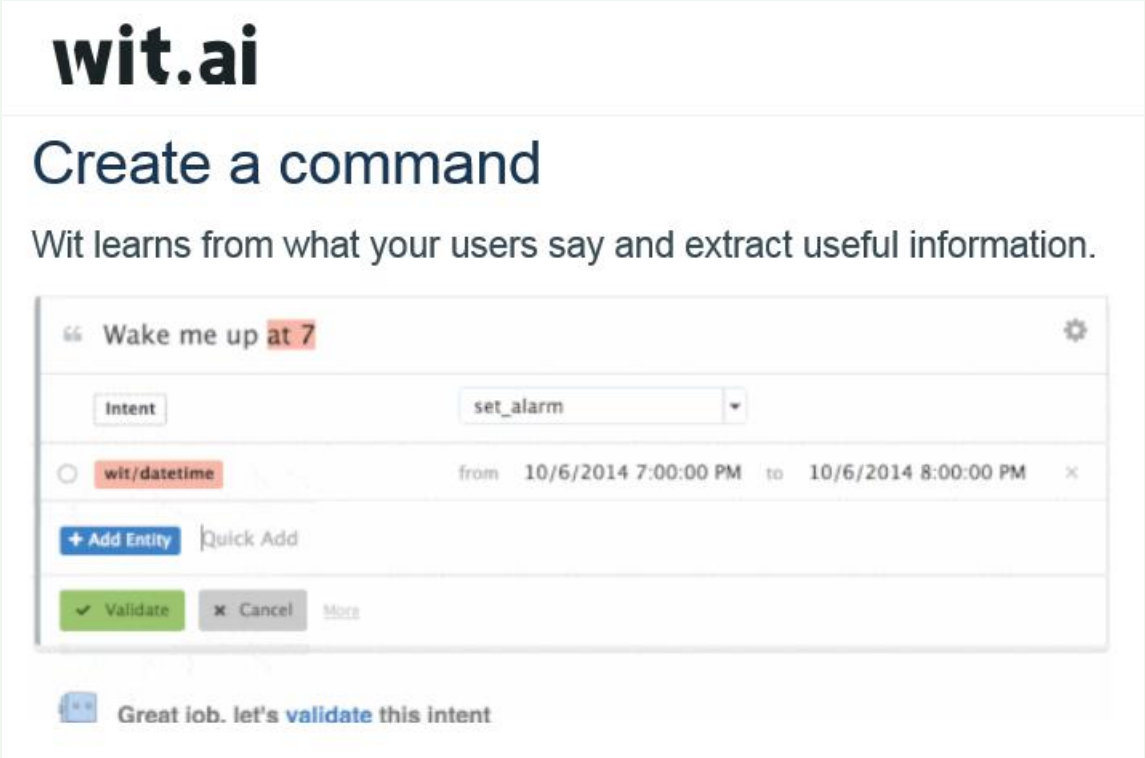



  - See also: http://www.novaspivack.com/technology/how-hisiri-works-interview-with-tom-gruber-cto-of-siri

Thanks to Prof. Bill MacCartney [1] for the slide [2]. Reproduced with permission.
[1] http://nlp.stanford.edu/~wcmac/
[2] http://web.stanford.edu/class/cs224u/slides/cs224u-2014-lec01-intro.pdf

# State of the Art: Examples

- Wit.ai: https://wit.ai/getting-started



# State of the Art: Examples

- How IBM Watson works?
  - www.cs.cornell.edu/courses/CS6700/2013sp/readings/01-presentation-watson_v2.pptx
- Wolfram Alpha
  - Little is publicly known. Some weakly relevant comments on Hacker News: https://news.ycombinator.com/item?id=7148121

# Language in Humans - 1

- Evolution
  - Vision/Sound for longest, then talking / listening, then reading and writing
  - Pictures are still worth a thousand words
    - E.g.: We remember faces more easily than names
    - Less visual cognition cycles for pictures than for reading?
- Child development
  - Vision/Sound, then talking / listening, then reading and writing
    - Noteworthy parallels to evolution!
    - Proceeding the same way for computers??
      - Speech/Vision → Language
- Action/perception cycle
  - Importance of context / doing while learning a language

# Language in Humans - 2

- How did languages originate? [1, 2]
  - Pidgins: Rudimentary language with a small vocabulary
  - Creole: Pidgins transform into native language
  - Bickerton's hypothesis:
    - No languages observed that lie between pidgins and real languages
    - Suggests language develops very rapidly during the transformation
- How/Why do languages change over time?
  - Evolving needs leading to learning and forgetting
  - Lossy language acquisition during childhood and across geography/cultures
  - Language changes spread like ideas, some survive some die (meme propagation)

[1] Steven Pinker, The Language Instinct
[2] Introducing Linguistics, Trask and Mayblin

# Fundamental Challenges in NLP/NLU

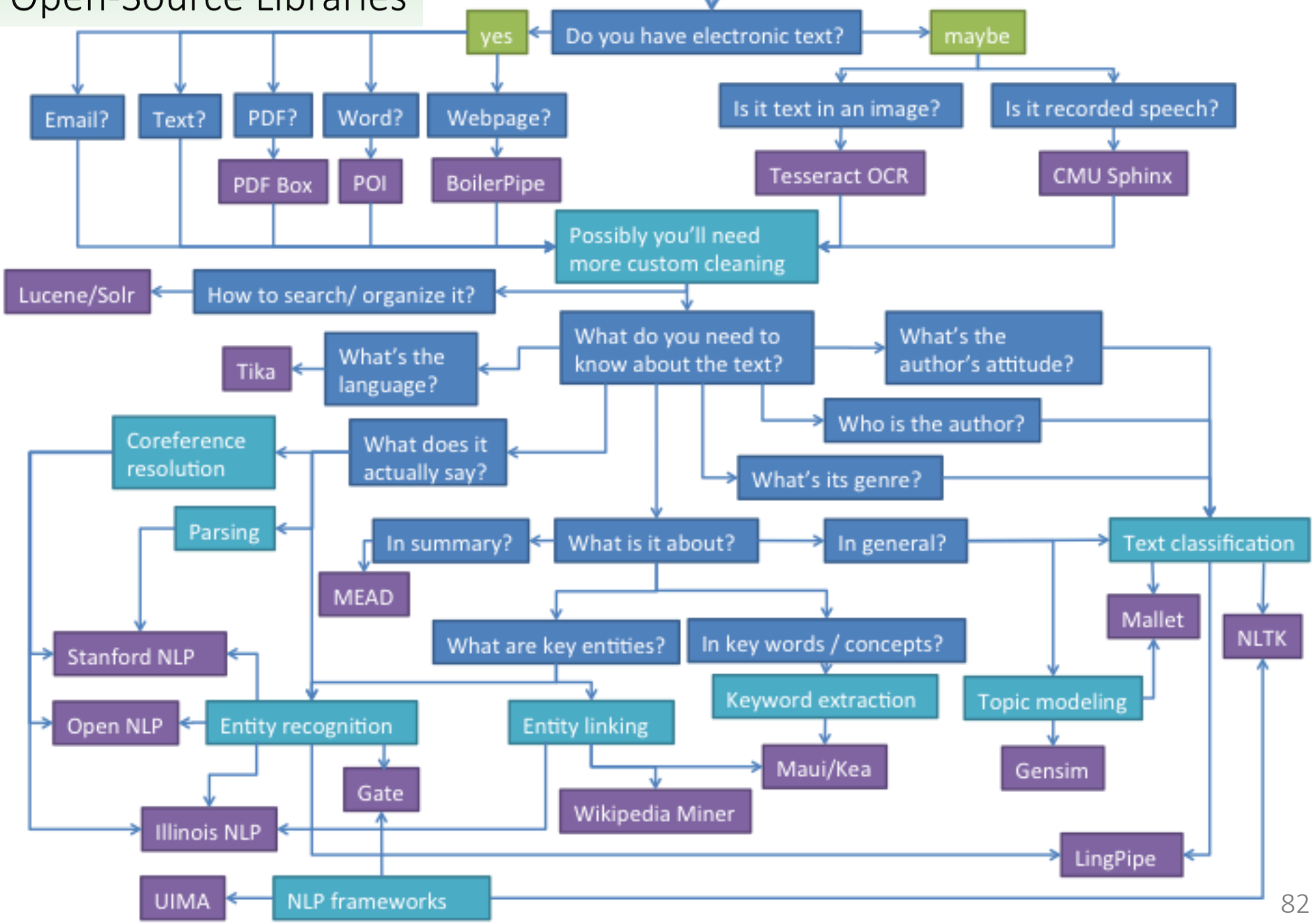
- Ill-posed problem?
- Number of thoughts / sentences possible is huge
  - Average length of a written sentence: 27-31 words. Say 30.
  - Vocabulary: No established upper-bound (Heap's law), say 30,000
  - Number of random sentences: $10^{\sim 134}$
  - Crude estimate for number of meaningful sentences: $10^{\sim 30}$ [1]
  - The same problem applies to vision though (still larger space), yet human-level performance on the horizon [2]
- Limited common-sense datasets or context
  - Children develop mental models of the world first before language
  - Computer vision and robotics not currently at par with humans for learning via action perception cycles (though maybe soon!)
  - Limited supervised training datasets in the interim
- The "algorithms" human brain uses not well understood
  - Humans often exhibit single-example learning
  - Finding correct language representation could be the key

[1] Steven Pinker, The Language Instinct, page 77
[2] http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

http://entopix.com/so-you-need-to-understand-language-data-open-source-nlp-software-can-help/

# Open-Source Libraries

Thanks to Alyona Medelyan, Reproduced with permission



# Suggested References

- Books:
  - Introductory:
    - Introducing Linguistics, Trask and Mayblin
    - The Language Instinct, Steven Pinker
    - Words and Rules, Steven Pinker
  - Natural Language Processing:
    - Speech and Language Processing, Jurafsky and Martin
    - Foundations of Statistical Natural Language Processing, Manning and Schutze
  - Parsing
    - Parsing Techniques, Dick Grune and Ceriel Jacobs
    - Grammars for Programming Languages, Cleaveland and Uzgalis
  - Grammar:
    - A Student's Introduction to English Grammar, Huddleston and Pullum
  - Tumultuous history of the Search for Artificial Intelligence, Daniel Crevier
- Websites and online courses
  - Coursera, Natural Language Processing: https://www.coursera.org/course/nlp
  - Natural Language Understanding Course website: http://web.stanford.edu/class/cs224u/

# My Contact

**Alok Govil**
Technology Architect (Anything EE + CS)

alokgovil@hotmail.com
1-408-533-2749
www.linkedin.com/in/alokgovil/

# APPENDICES

# Session Description

- Natural Language Processing (NLP) and Understanding (NLU) aim to make machines process human languages like English. This session will provide a complete overview of the field from the basic structure of human languages to the state of the art.
- The session will focus on (a) deep theoretical understanding instead of mere use of pre-existing NLP libraries, (b) natural language understanding aspects instead of keywords-based analysis or text classification, (c) processing of 'English' 'text' and not speech recognition/synthesis or language translation.
- We will discuss NLP applications and challenges, language components, Chomsky's hierarchy of grammars, parsing algorithms, word-sense disambiguation, logic and inference, language synthesis, and available test/training datasets.

# About Me

## PROFESSIONAL SUMMARY

- Qualcomm, Lead Technology Architect and Project Engineer
- Philips Research, Researcher
- MSEE, University of Southern California
- Research Assistant (USC), Computer Vision and Augmented Reality
- River Run Software Group

## AREAS OF INTEREST

- Natural Language Processing
- Artificial Intelligence
- Computer Vision
- Software development (PC/Embedded)
- System architecture and design
- Electronic displays (LCD/MEMS)
- Analog and digital circuit design