Natural Language Processing

Péter Molnár

pmolnar@rentpath.com/pmolnar@gsu.edu

JESAL
@JesalTV

Follow

Jeff Bezos: "Alexa, buy me something from Whole Foods."

Alexa: "Sure, Jeff. Buying Whole Foods now."

Jeff Bezos: "WHA- ahh go ahead."

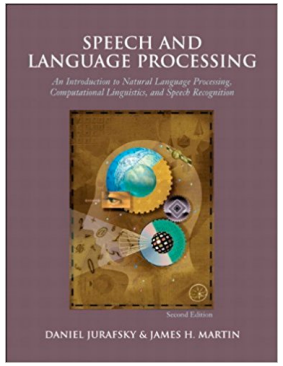
9:23 AM - 16 Jun 2017

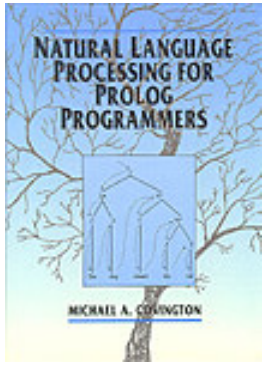
19,839 35,724

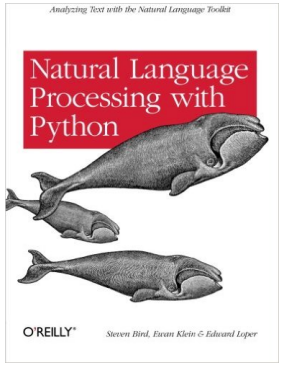
Toolkits

- Natural Language Toolkit http://www.nltk.org/
  - Comprehensive Python programming platform to work with human language data.
  - Corpora and lexical resources such as WordNet. Text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning
- Stanford CoreNLP https://stanfordnlp.github.io/CoreNLP/
  - Statistical parser to produce syntax tree (Java)
  - Available languages: Arabic, Chinese, English, French, German, and Spanish
- OpenNLP http://opennlp.apache.org/
  - Sentence detector, tokenizer, name finder, document categorizer, part-of-speech tagger, chunker, parser, coreference resolution (Java)
- Prolog Natuaral Language Tool (ProNTo) http://ai1.ai.uga.edu/mc/pronto/
- Facebook AI Research Sequence-to-Sequence Toolkit https://github.com/facebookresearch/fairseq
  - Sequence-to-sequence learning toolkit for Torch for Neural Machine Translation (NMT).
  - Pre-trained models for English to French, English to German and English to Romanian translation.
- SyntaxNet: https://github.com/tensorflow/models/tree/master/syntaxnet
  - TensorFlow toolkit for deep learning powered natural language understanding (NLU).

Books







- Dan Jurafsky & James H. Martin, *Speech and Language Processing* (3rd ed. draft). https://web.stanford.edu/~jurafsky/slp3/
- Chris Manning & Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA: May 1999. https://nlp.stanford.edu/fsnlp/ (PDF)
- Michael A. Covington, Natural Language Processing for Prolog Programmers, Prentice-Hall, 1994. http://www.covingtoninnovations.com/books.html#nlp
- Steven Bird, Ewan Klein, & Edward Loper, *Natural Language Processing with Python,* O'Reilly, 2009. http://www.nltk.org/book/

PROCEDURES AS A REPRESENTATION FOR DATA
IN A COMPUTER PROGRAM FOR UNDERSTANDING
NATURAL LANGUAGE

by

Terry Winograd

Massachusetts Institute of Technology

January 1971

http://hci.stanford.edu/~winograd/shrdlu/

ABSTRACT

This paper describes a system for the computer understanding of English. The system answers questions, executes commands, and accepts information in normal English dialog. It uses semantic information and context to understand discourse and to disambiguate sentences. It combines a complete syntactic analysis of each sentence with a "heuristic understander" which uses different kinds of information about a sentence, other parts of the discourse, and general information about the world in deciding what the sentence means.

It is based on the belief that a computer cannot deal reasonably with language unless it can "understand" the subject it is discussing. The program is given a detailed model of the knowledge needed by a simple robot having only a hand and an eye. We can give it instructions to manipulate toy objects, interrogate it about the scene, and give it information it will use in deduction. In addition to knowing the properties of toy objects, the program has a simple model of its own mentality. It can remember and discuss its plans and actions as well as carry them out. It enters into a dialog with a person, responding to English sentences with actions and English replies, and asking for clarification when its heuristic programs cannot understand a sentence through use of context and physical knowledge.

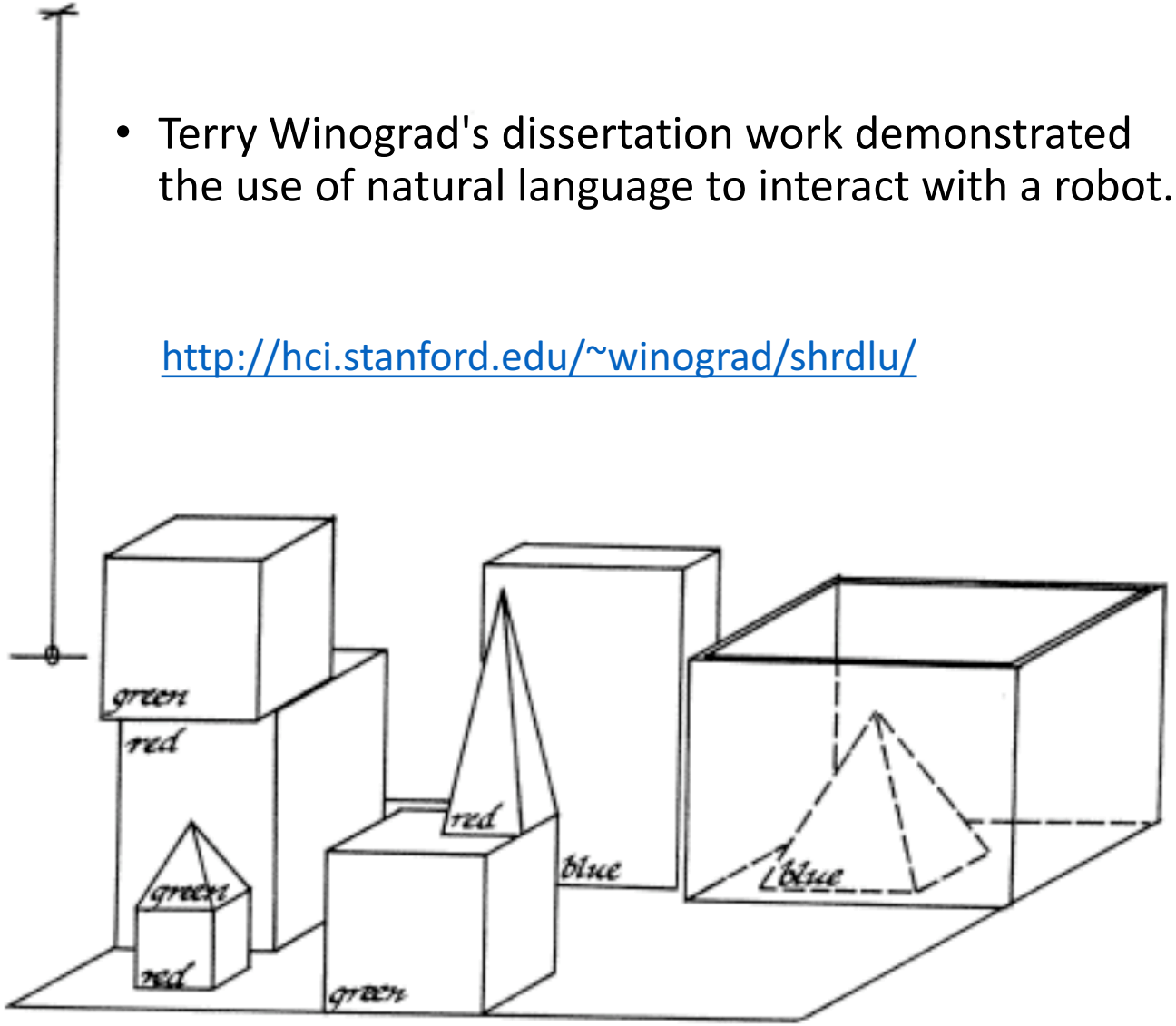
In the programs, syntax, semantics and inference are integrated in a "vertical" system in which each part is

The Robot's World

SHRDLU – Blocks World

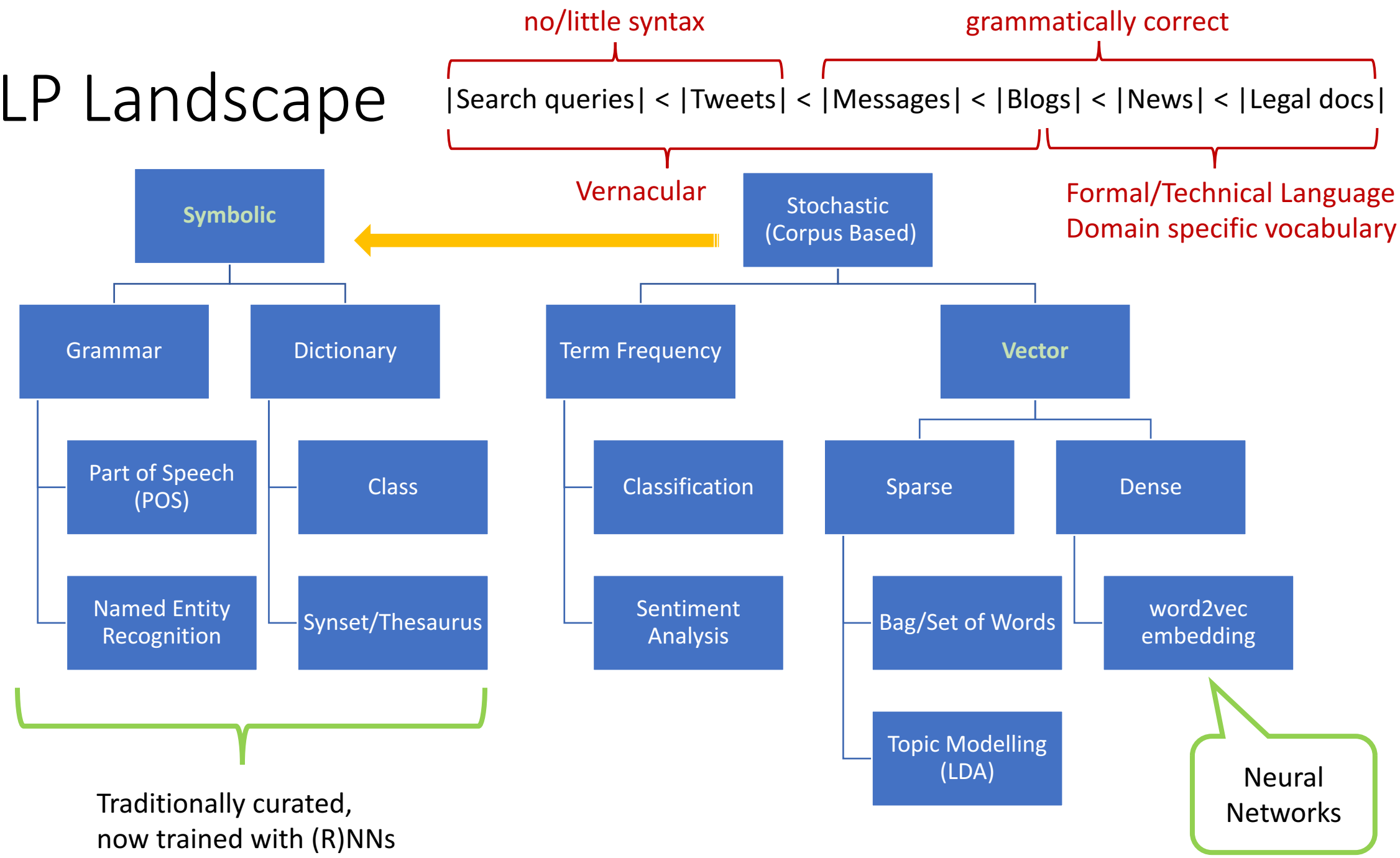
- Terry Winograd's dissertation work demonstrated the use of natural language to interact with a robot.

http://hci.stanford.edu/~winograd/shrdlu/



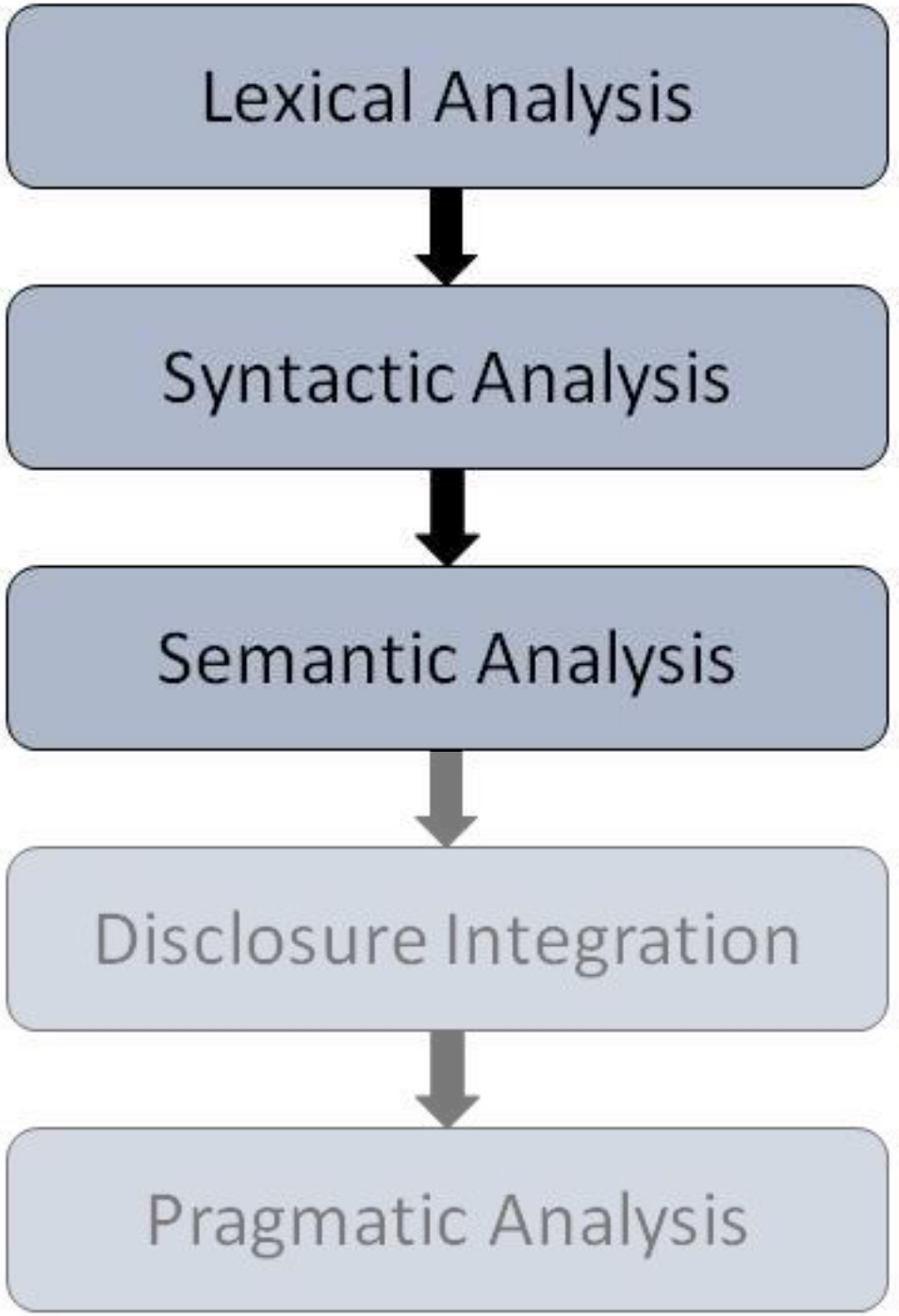
Person: PICK UP A BIG RED BLOCK.
Computer: OK. (does it)
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
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: IS AT LEAST ONE OF THEM NARROWER THAN THE ONE WHICH I TOLD YOU TO PICK UP?
Computer: YES, THE RED CUBE.

NLP Landscape



NLP Pipeline

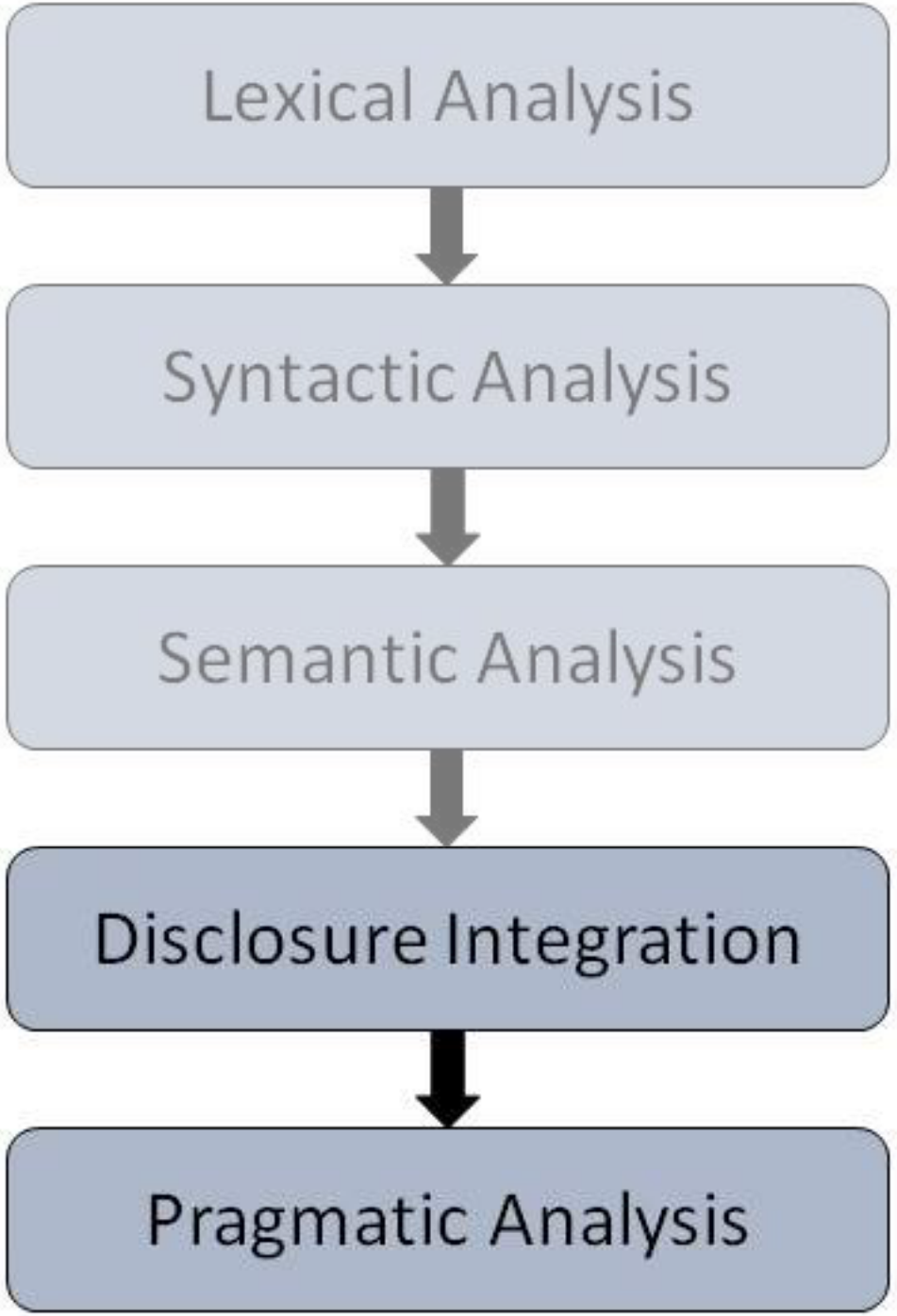
- **Lexical Analysis** involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of text into paragraphs, sentences, and words.
- **Syntactic Analysis** involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.
- **Semantic Analysis** draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.



NLP Pipeline (2)

- **Disclosure Integration**: The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
- **Pragmatic Analysis:** During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

- Source: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm



Sentiment Analysis

- Classifying the **polarity** of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is **positive**, **negative**, or **neutral**.
- Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "**angry**", "**sad**", and "**happy**".

- Dataset https://github.com/jperla/sentiment-data

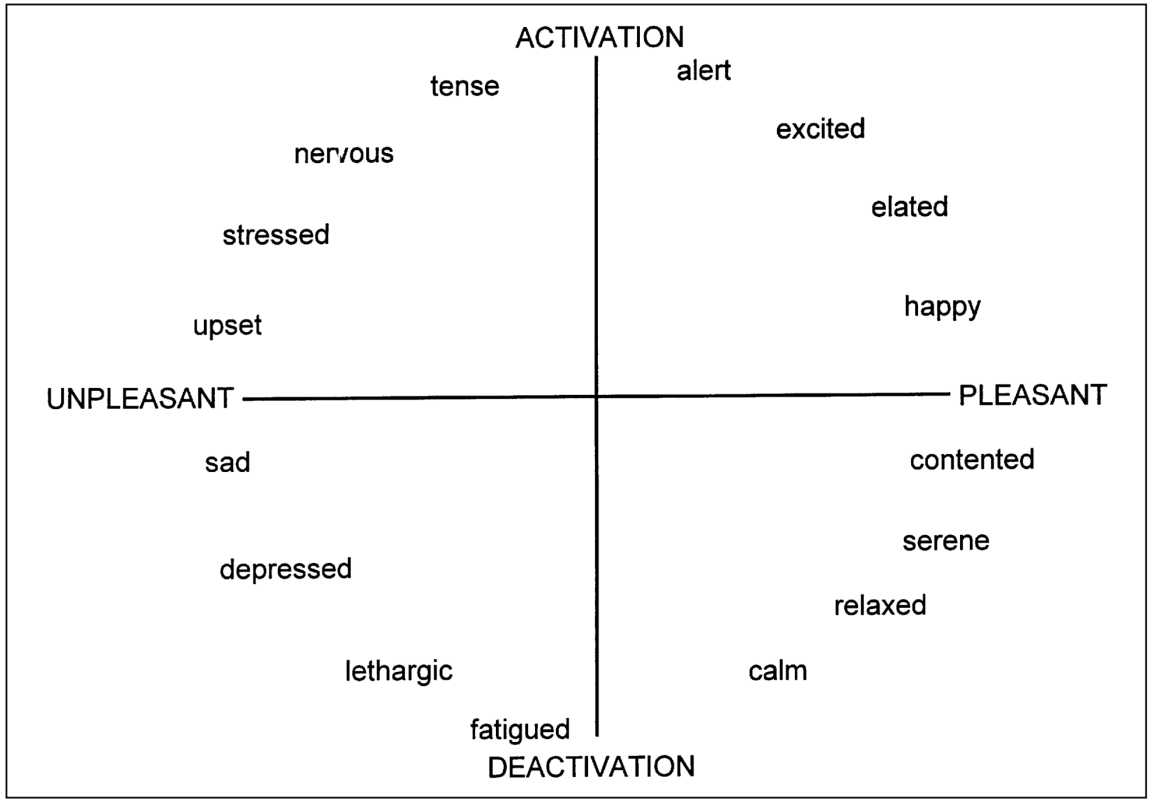


Fig. 2. A schematic for the two-dimensional structure of affect. Adapted from Feldman Barrett and Russell (1998).

Challenges in Sentiment Analysis

- **Simple cases → determine polarity based on trigger words might be sufficient**
  - *Coronet has the best lines of all day cruisers.*
  - *Bertram has a deep V hull and runs easily through seas.*
  - *Pastel-colored 1980s day cruisers from Florida are ugly.*
  - *I dislike old cabin cruisers*.
- **More challenging examples → trigger words alone would fail, require syntax analysis**
  - *I do not dislike cabin cruisers.* (Negation handling)
  - *Disliking watercraft is not really my thing.* (Negation, inverted word order)
  - *Sometimes I really hate RIBs.* (Adverbial modifies the sentiment)
  - *I'd really truly love going out in this weather!* (Possibly sarcastic)
  - *Chris Craft is better looking than Limestone.* (Two brand names, identifying target of attitude is difficult).
  - *Chris Craft is better looking than Limestone, but Limestone projects seaworthiness and reliability.* (Two attitudes, two brand names).
  - The movie is surprising with plenty of unsettling plot twists. (Negative term used in positive sense).
  - *I love my mobile but would not recommend it to any of my colleagues*. (Qualified positive sentiment, difficult to categorize)

Sentiment Treebank

- Challenge to express the meaning of longer phrases in a principled way.
- Accurately capture the effects of negation and its scope at various tree levels for both positive and negative phrases
- Based on Stanford Sentiment Treebank
- Ref: Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. 2013. *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.*

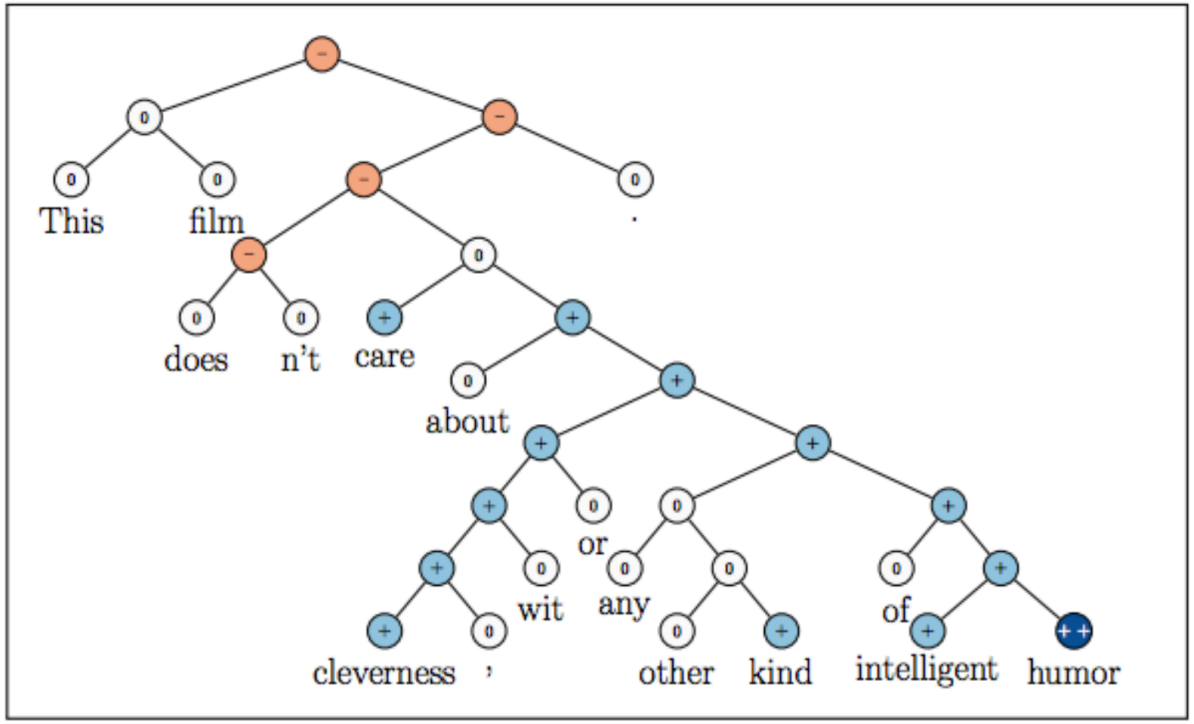


Figure 1: Example of the Recursive Neural Tensor Network accurately predicting 5 sentiment classes, very negative to very positive (– –, –, 0, +, + +), at every node of a parse tree and capturing the negation and its scope in this sentence.

https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf

Formal Languages/Grammars

A formal grammar consists of a finite set of **production rules** (*left-hand side* → *right-hand side*), where each side consists of a finite sequence of the following symbols:

- a finite set of **nonterminal symbols** (indicating that some production rule can yet be applied)
- a finite set of **terminal symbols** (indicating that no production rule can be applied)
- a **start symbol** (a distinguished nonterminal symbol)

Nonterminals are often represented by uppercase letters, terminals by lowercase letters, and the start symbol by S.

For example, the grammar with terminals {*a, b*}, nonterminals {*S, A, B*}, production rules

$S \rightarrow AB$
$S \rightarrow \varepsilon$ (where ε is the empty string)
$A \rightarrow aS$
$B \rightarrow b$

and start symbol S, defines the language of all words of the form:

ab
aabb
aaabbb
…

Chomsky, Noam (1956). "Three models for the description of language" . IRE Transactions on Information Theory (2): 113–124. doi:10.1109/TIT.1956.1056813

(Toy) example of English language

- **Terminals** {generate, hate, great, green, ideas, linguists}
- **Nonterminals** {*SENTENCE, NOUNPHRASE, VERBPHRASE, NOUN, VERB, ADJ*}
- **Production Rules**
  SENTENCE → *NOUNPHRASE VERBPHRASE*
  NOUNPHRASE → *ADJ NOUNPHRASE*
  NOUNPHRASE → *NOUN*
  VERBPHRASE → *VERB NOUNPHRASE*
  VERBPHRASE → *VERB*
  NOUN → *ideas*
  NOUN → *linguists*
  VERB → *generate*
  VERB → *hate*
  ADJ → *great*
  ADJ → *green*
- **Start Symbol** *SENTENCE*

green ideas generate great linguists

.

Example Grammar

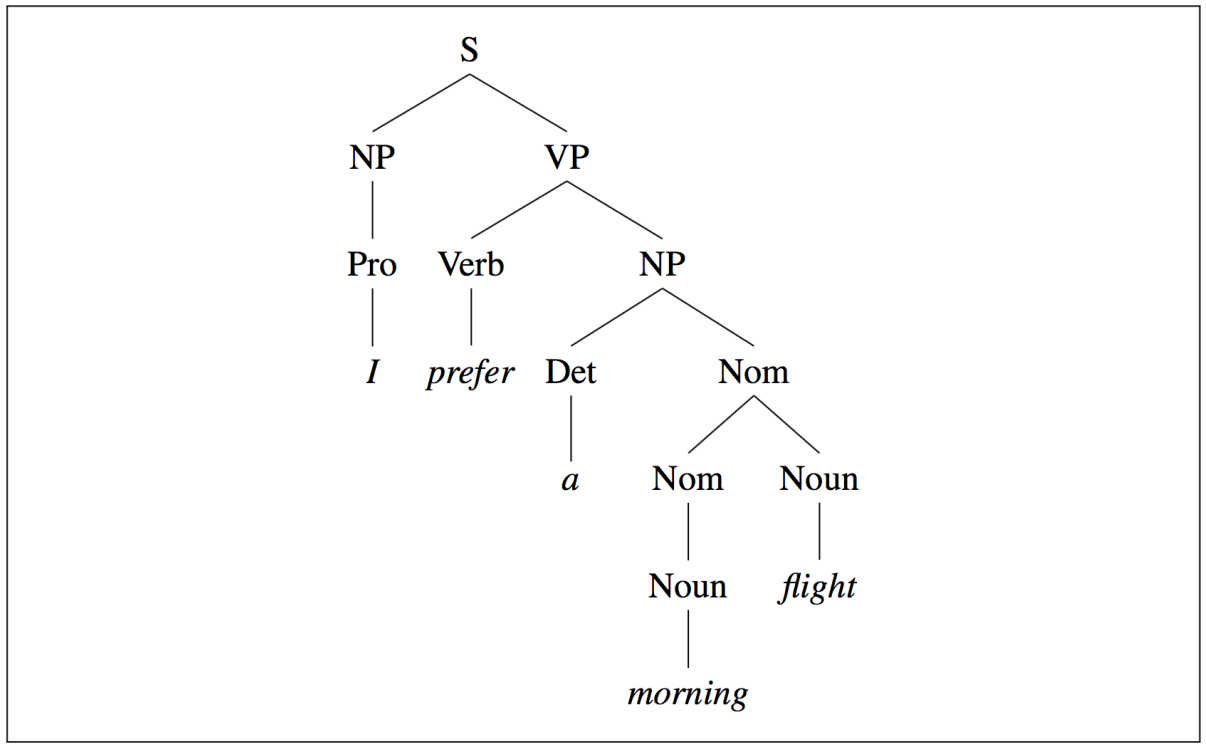


Figure 11.4 The parse tree for "I prefer a morning flight" according to grammar $\mathscr{L}_0$.

Noun	→	*flights* | *breeze* | *trip* | *morning*
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	→	*cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun	→	*me* | *I* | *you* | *it*
Proper-Noun	→	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner	→	*the* | *a* | *an* | *this* | *these* | *that*
Preposition	→	*from* | *to* | *on* | *near*
Conjunction	→	*and* | *or* | *but*

Figure 11.2 The lexicon for $\mathscr{L}_0$.

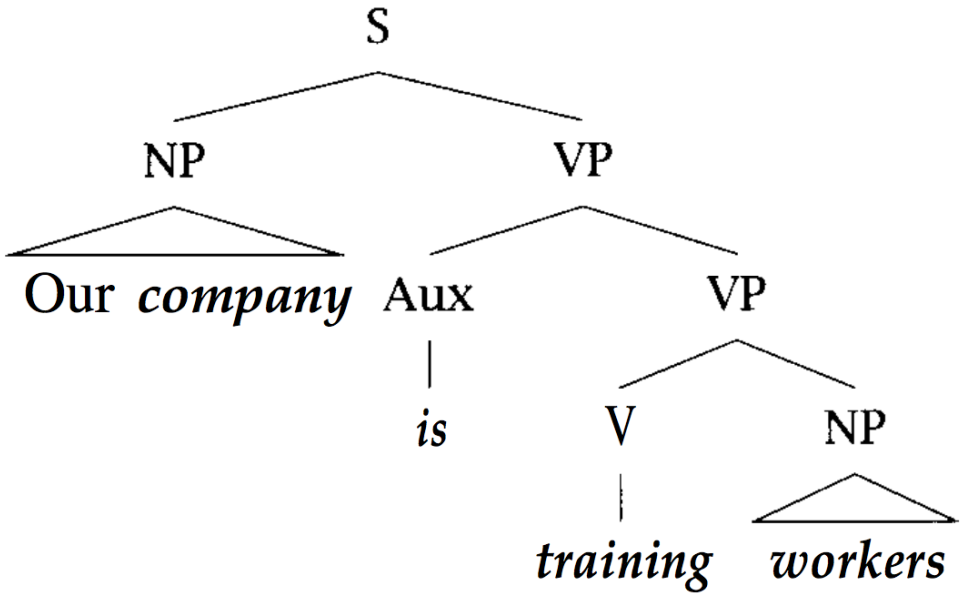
Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

Figure 11.3 The grammar for $\mathscr{L}_0$, with example phrases for each rule.

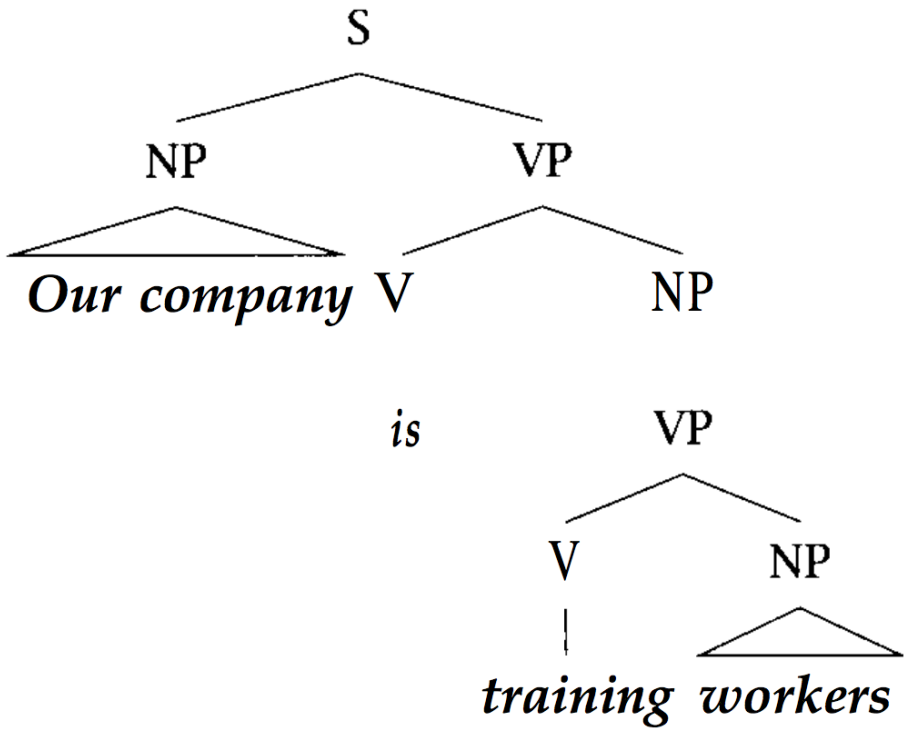
https://web.stanford.edu/~jurafsky/slp3/11.pdf

Ambiguity of Language

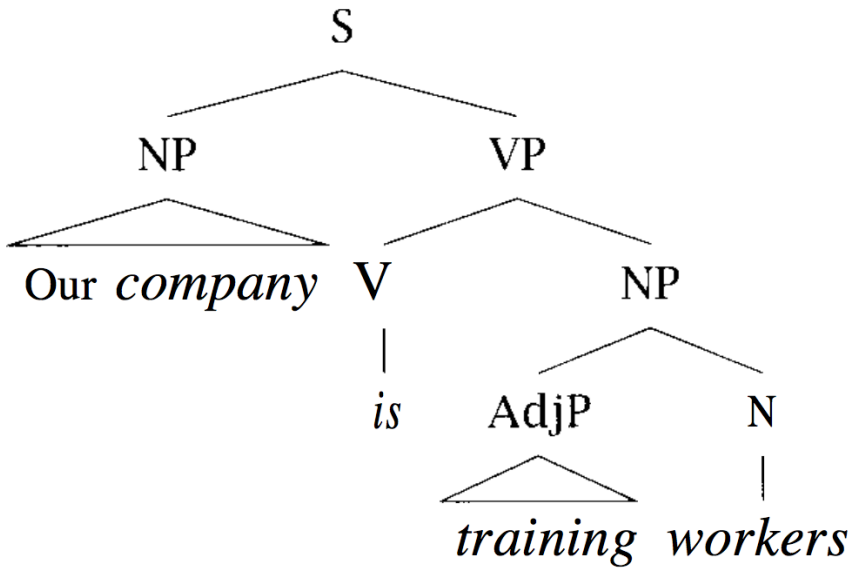
a.



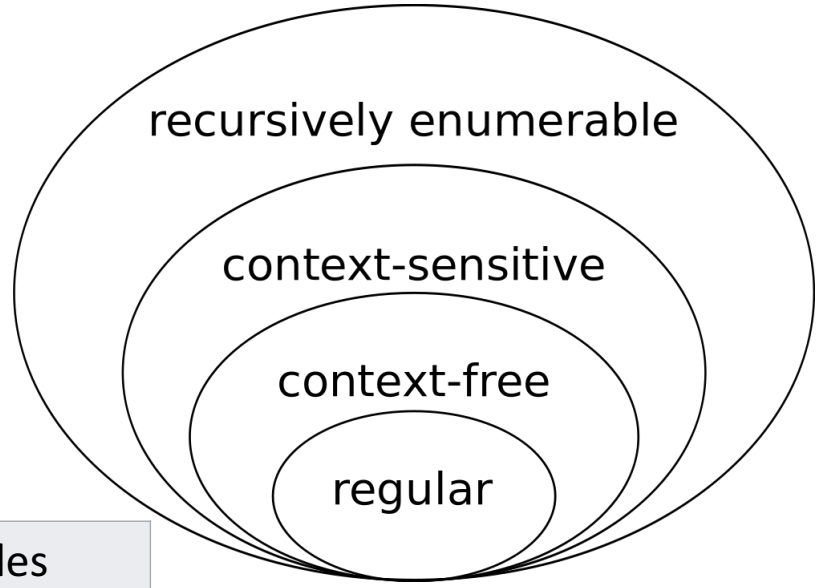
b.



c.



Chomsky's four types of grammars



Grammar	Languages	Automaton	Production rules (constraints)
Type-0	Recursively enumerable	Turing machine	$\alpha \to \beta$ (no restrictions)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \to \alpha \gamma \beta$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \to \gamma$
Type-3	Regular	Finite state automaton	$A \to a$ and $A \to aB$

Demos

- Introduce Prolog: kinship.pl
- Simple grammar for writing out numbers: numbers.pl
- Simple English grammar with concept representation: naturallanguage.pl
- Free-word dependency parser (Latin): dparse.pl

- Presentation and examples are posted at https://github.com/kingmolnar/natural_language