

Natural Language Processing

Chapter 15: Rich & knight

Dr. Suthikshn Kumar

Created with



nitro^{PDF} professional

download the free trial online at nitropdf.com/professional

NLP Intro

- Language is meant for Communicating about the world.
- By studying language, we can come to understand more about the world.
- If we can succeed at building computational mode of language, we will have a powerful tool for communicating about the world.
- We look at how we can exploit knowledge about the world, in combination with linguistic facts, to build computational natural language systems.
- NLP problem can be divided into two tasks:
 - Processing written text, using lexical, syntactic and semantic knowledge of the language as well as the required real world information.
 - Processing spoken language, using all the information needed above plus additional knowledge about phonology as well as enough added information to handle the further ambiguities that arise in speech.

Steps in NLP

- **Morphological Analysis:** Individual words are analyzed into their components and nonword tokens such as punctuation are separated from the words.
- **Syntactic Analysis:** Linear sequences of words are transformed into structures that show how the words relate to each other.
- **Semantic Analysis:** The structures created by the syntactic analyzer are assigned meanings.
- **Discourse integration:** The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that follow it.
- **Pragmatic Analysis:** The structure representing what was said is reinterpreted to determine what was actually meant.

Morphological Analysis

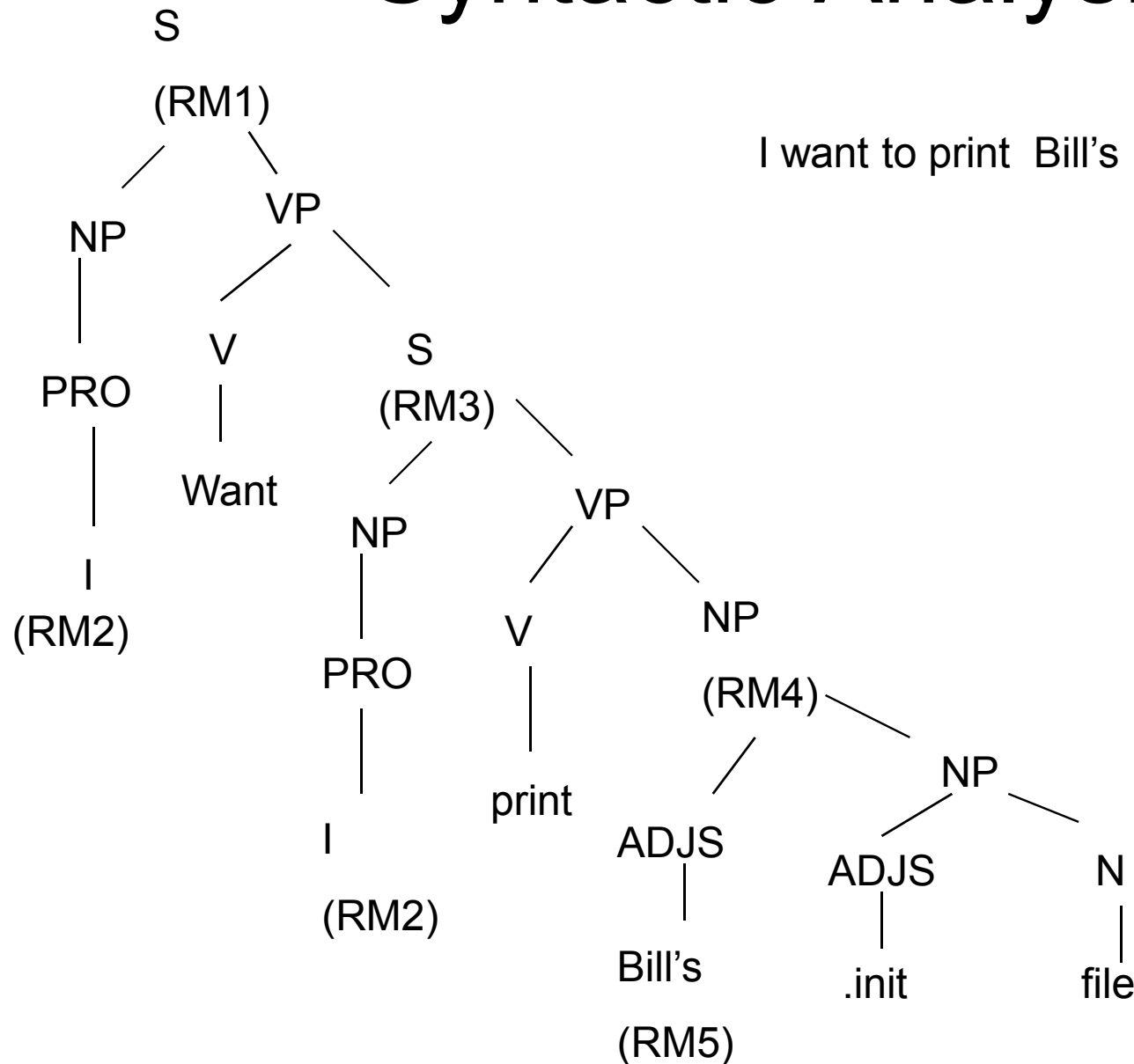
- Suppose we have an english interface to an operating system and the following sentence is typed:
 - I want to print Bill's .init file.
- Morphological analysis must do the following things:
 - Pull apart the word “Bill's” into proper noun “Bill” and the possessive suffix “s”
 - Recognize the sequence “.init” as a file extension that is functioning as an adjective in the sentence.
- This process will usually assign syntactic categories to all the words in the sentece.
- Consider the word “prints”. This word is either a pulral noun or a third person singular verb (he prints).

Syntactic Analysis

- Syntactic analysis must exploit the results of morphological analysis to build a structural description of the sentence.
- The goal of this process, called parsing, is to convert the flat list of words that forms the sentence into a structure that defines the units that are represented by that flat list.
- The important thing here is that a flat sentence has been converted into a hierarchical structure and that the structure correspond to meaning units when semantic analysis is performed.
- Reference markers are shown in the parenthesis in the parse tree
- Each one corresponds to some entity that has been mentioned in the sentence.

Syntactic Analysis

I want to print Bill's .init file.



Created with

 **nitro**^{PDF} professional

download the free trial online at nitropdf.com/professional

Semantic Analysis

- Semantic analysis must do two important things:
 - It must map individual words into appropriate objects in the knowledge base or database
 - It must create the correct structures to correspond to the way the meanings of the individual words combine with each other.

Discourse Integration

- Specifically we do not know whom the pronoun “I” or the proper noun “Bill” refers to.
- To pin down these references requires an appeal to a model of the current discourse context, from which we can learn that the current user is USER068 and that the only person named “Bill” about whom we could be talking is USER073.
- Once the correct referent for Bill is known, we can also determine exactly which file is being referred to.

Pragmatic Analysis

- The final step toward effective understanding is to decide what to do as a results.
- One possible thing to do is to record what was said as a fact and be done with it.
- For some sentences, whose intended effect is clearly declarative, that is precisely correct thing to do.
- But for other sentences, including ths one, the intended effect is different.
- We can discover this intended effect by applying a set of rules that characterize cooperative dialogues.
- The final step in pragmatic processing is to translate, from the knowledge based representation to a command to be executed by the system.
- The results of the understanding process is
- Lpr /wsmith/stuff.init

Summary

- Results of each of the main processes combine to form a natural language system.
- All of the processes are important in a complete natural language understanding system.
- Not all programs are written with exactly these components.
- Sometimes two or more of them are collapsed.
- Doing that usually results in a system that is easier to build for restricted subsets of English but one that is harder to extend to wider coverage.

Syntactic Processing

- Syntactic Processing is the step in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence.
- This process is called parsing.
- It plays an important role in natural language understanding systems for two reasons:
 - Semantic processing must operate on sentence constituents. If there is no syntactic parsing step, then the semantics system must decide on its own constituents. If parsing is done, on the other hand, it constrains the number of constituents that semantics can consider. Syntactic parsing is computationally less expensive than is semantic processing. Thus it can play a significant role in reducing overall system complexity.
 - Although it is often possible to extract the meaning of a sentence without using grammatical facts, it is not always possible to do so. Consider the examples:
 - The satellite orbited Mars
 - Mars orbited the satellite
 - In the second sentence, syntactic facts demand an interpretation in which a planet revolves around a satellite, despite the apparent improbability of such a scenerio.

Syntactic Processing

- Almost all the systems that are actually used have two main components:
 - A declarative representation, called a grammar, of the syntactic facts about the language.
 - A procedure, called parser, that compares the grammar against input sentences to produce parsed structures.

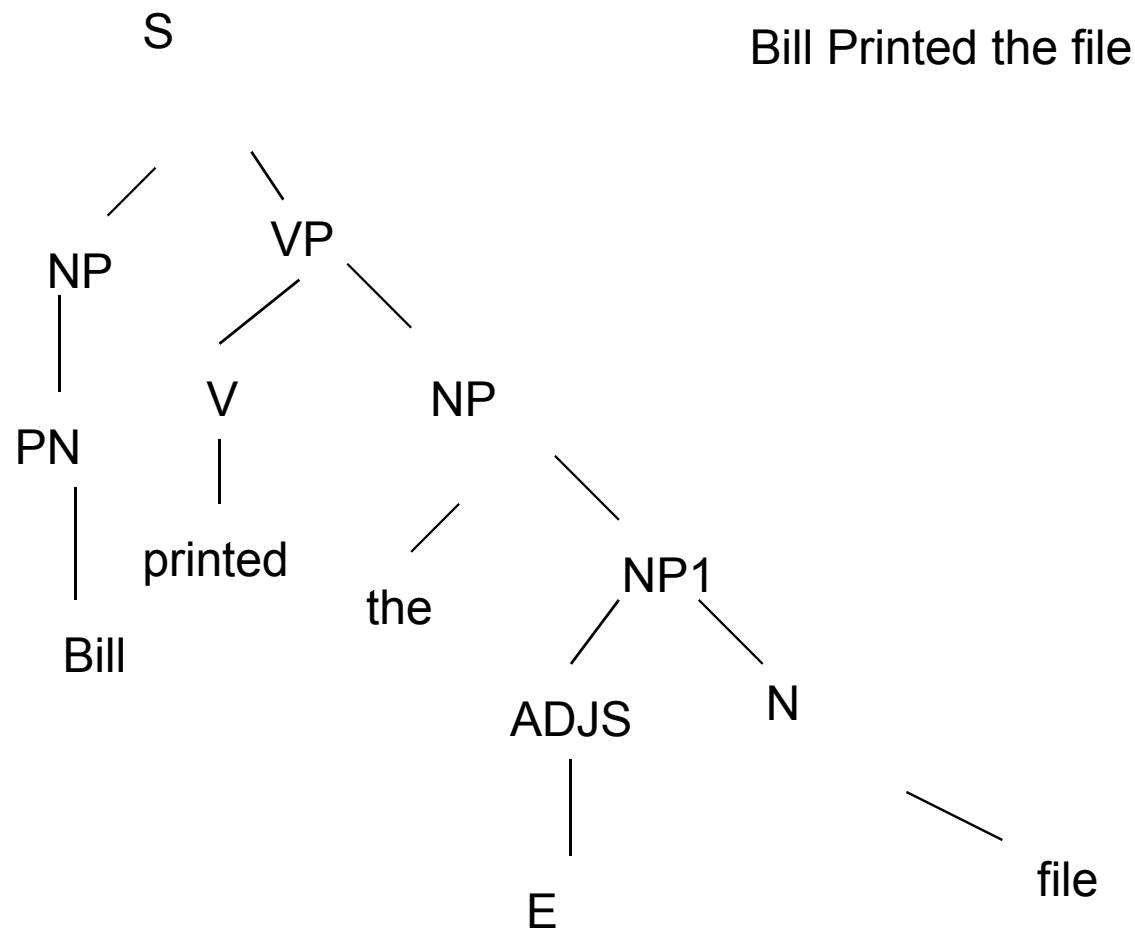
Grammars and Parsers

- The most common way to represent grammars is as a set of production rules.
- A simple Context-free phrase structure grammar for English:
 - $S \rightarrow NP VP$
 - $NP \rightarrow the NP1$
 - $NP \rightarrow PRO$
 - $NP \rightarrow PN$
 - $NP \rightarrow NP1$
 - $NP1 \rightarrow ADJS N$
 - $ADJS \rightarrow \epsilon \mid ADJ ADJS$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$
 - $N \rightarrow file \mid printer$
 - $PN \rightarrow Bill$
 - $PRO \rightarrow I$
 - $ADJ \rightarrow short \mid long \mid fast$
 - $V \rightarrow printed \mid created \mid want$
- First rule can be read as “A sentence is composed of a noun phrase followed by Verb Phrase”; Vertical bar is OR ; ϵ represents empty string.
- Symbols that are further expanded by rules are called nonterminal symbols.
- Symbols that correspond directly to strings that must be found in an input sentence are called terminal symbols.

Grammars and Parsers

- Grammar formalism such as this one underlie many linguistic theories, which in turn provide the basis for many natural language understanding systems.
- Pure context free grammars are not effective for describing natural languages.
- NLPs have less in common with computer language processing systems such as compilers.
- Parsing process takes the rules of the grammar and compares them against the input sentence.
- The simplest structure to build is a Parse Tree, which simply records the rules and how they are matched.
- Every node of the parse tree corresponds either to an input word or to a nonterminal in our grammar.
- Each level in the parse tree corresponds to the application of one grammar rule.

A Parse tree for a sentence



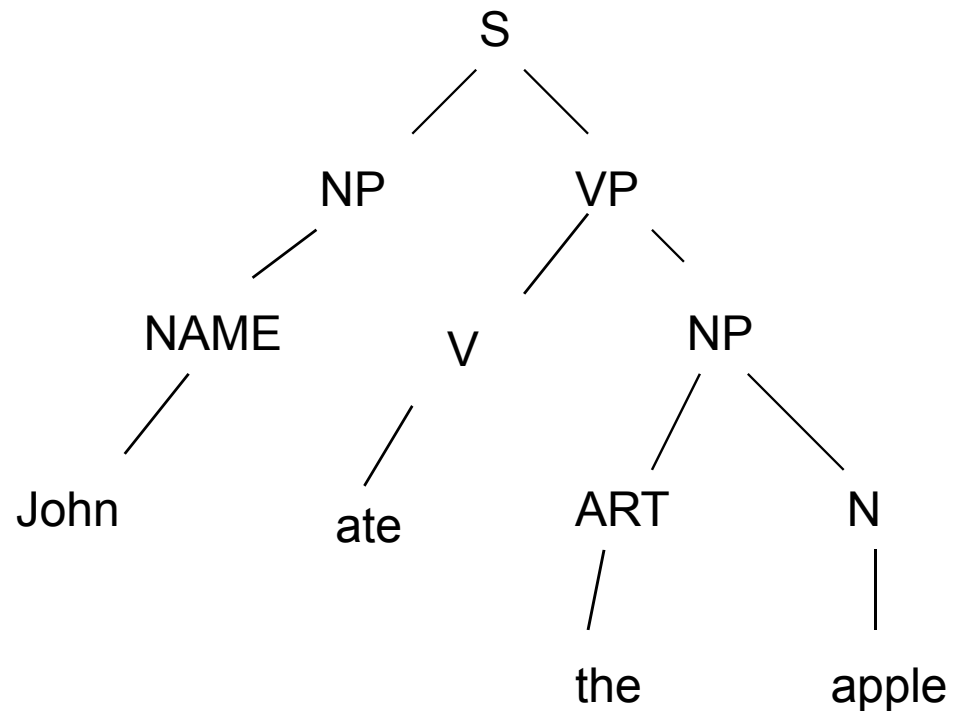
Created with

 **nitro**^{PDF} professional

download the free trial online at nitropdf.com/professional

A parse tree

- John ate the apple.
1. $S \rightarrow NP VP$
 2. $VP \rightarrow V NP$
 3. $NP \rightarrow NAME$
 4. $NP \rightarrow ART N$
 5. $NAME \rightarrow John$
 6. $V \rightarrow ate$
 7. $ART \rightarrow the$
 8. $N \rightarrow apple$



Exercise: For each of the following sentences, draw a parse tree

- John wanted to go to the movie with Sally
- I heard the story listening to the radio.
- All books and magazines that deal with controversial topics have been removed from the shelves.

What grammar specifies about language?

- Its weak generative capacity, by which we mean the set of sentences that are contained within the language. This set is made up of precisely those sentences that can be completely matched by a series of rules in the grammar.
- Its strong generative capacity, by which we mean the structure to be assigned to each grammatical sentence of the language.

Top-down versus Bottom-Up parsing

- To parse a sentence, it is necessary to find a way in which that sentence could have been generated from the start symbol. There are two ways this can be done:
 - Top-down Parsing: Begin with start symbol and apply the grammar rules forward until the symbols at the terminals of the tree correspond to the components of the sentence being parsed.
 - Bottom-up parsing: Begin with the sentence to be parsed and apply the grammar rules backward until a single tree whose terminals are the words of the sentence and whose top node is the start symbol has been produced.
- The choice between these two approaches is similar to the choice between forward and backward reasoning in other problem-solving tasks.
- The most important consideration is the branching factor. Is it greater going backward or forward?
- Sometimes these two approaches are combined to a single method called “bottom-up parsing with top-down filtering”.

Finding one interpretation or finding many

Created with

 **nitro**^{PDF} professional

download the free trial online at nitropdf.com/professional

Augmented Transition Networks

Created with



download the free trial online at nitropdf.com/professional

Unification Grammars

Created with

 **nitro**^{PDF} professional

download the free trial online at nitropdf.com/professional

Semantic Analysis

- Producing a syntactic parse of a sentence is only the first step toward understanding it.
- We must still produce a representation of the meaning of the sentence.
- Because understanding is a mapping process, we must first define the language into which we are trying to map.
- There is no single definitive language in which all sentence meaning can be described.
- The choice of a target language for any particular natural language understanding program must depend on what is to be done with the meanings once they are constructed.

Choice of target language in semantic Analysis

- There are two broad families of target languages that are used in NL systems, depending on the role that the natural language system is playing in a larger system:
 - When natural language is being considered as a phenomenon on its own, as for example when one builds a program whose goal is to read text and then answer questions about it, a target language can be designed specifically to support language processing.
 - When natural language is being used as an interface language to another program(such as a db query system or an expert system), then the target language must be legal input to that other program. Thus the design of the target language is driven by the backend program.

Lexical processing

- The first step in any semantic processing system is to look up the individual words in a dictionary (or lexicon) and extract their meanings.
- Many words have several meanings, and it may not be possible to choose the correct one just by looking at the word itself.
- The process of determining the correct meaning of an individual word is called word sense disambiguation or lexical disambiguation.
- It is done by associating, with each word in lexicon, information about the contexts in which each of the word's senses may appear.
- Sometimes only very straightforward info about each word sense is necessary. For example, baseball field interpretation of diamond could be marked as a LOCATION.
- Some useful semantic markers are :
 - PHYSICAL-OBJECT
 - ANIMATE-OBJECT
 - ABSTRACT-OBJECT

Sentence-Level Processing

- Several approaches to the problem of creating a semantic representation of a sentence have been developed, including the following:
 - Semantic grammars, which combine syntactic, semantic and pragmatic knowledge into a single set of rules in the form of grammar.
 - Case grammars, in which the structure that is built by the parser contains some semantic information, although further interpretation may also be necessary.
 - Conceptual parsing in which syntactic and semantic knowledge are combined into a single interpretation system that is driven by the semantic knowledge.
 - Approximately compositional semantic interpretation, in which semantic processing is applied to the result of performing a syntactic parse

Semantic Grammar

- A semantic grammar is a context-free grammar in which the choice of nonterminals and production rules is governed by semantic as well as syntactic function.
- There is usually a semantic action associated with each grammar rule.
- The result of parsing and applying all the associated semantic actions is the meaning of the sentence.

A semantic grammar

- S-> what is FILE-PROPERTY of FILE?
 - { query FILE.FILE-PROPERTY }
- S-> I want to ACTION
 - { command ACTION }
- FILE-PROPERTY -> the FILE-PROP
 - { FILE-PROP }
- FILE-PROP -> extension | protection | creation date | owner
 - { value }
- FILE -> FILE-NAME | FILE1
 - { value }
- FILE1 -> USER's FILE2
 - { FILE2.owner: USER }
- FILE1 -> FILE2
 - { FILE2 }
- FILE2 -> EXT file
 - { instance: file-struct extension: EXT }
- EXT -> .init | .txt | .lsp | .for | .ps | .mss
 - value
- ACTION -> print FILE
 - { instance: printing object : FILE }
- ACTION -> print FILE on PRINTER
 - { instance : printing object : FILE printer : PRINTER }
- USER -> Bill | susan
 - { value }

Advantages of Semantic grammars

- When the parse is complete, the result can be used immediately without the additional stage of processing that would be required if a semantic interpretation had not already been performed during the parse.
- My ambiguities that would arise during a strictly syntactic parse can be avoided since some of the interpretations do not make sense semantically and thus cannot be generated by a semantic grammar.
- Syntactic issues that do not affect the semantics can be ignored.
- The drawbacks of use of semantic grammars are:
 - The number of rules required can become very large since many syntactic generalizations are missed.
 - Because the number of grammar rules may be very large, the parsing process may be expensive.

Case grammars

- Case grammars provide a different approach to the problem of how syntactic and semantic interpretation can be combined.
- Grammar rules are written to describe syntactic rather than semantic regularities.
- But the structures the rules produce correspond to semantic relations rather than to strictly syntactic ones
- Consider two sentences:
 - Susan printed the file.
 - The file was printed by Susan.
- The case grammar interpretation of the two sentences would both be :
- (printed (agent Susan)
- (object File))

Conceptual Parsing

- Conceptual parsing is a strategy for finding both the structure and meaning of a sentence in one step.
- Conceptual parsing is driven by dictionary that describes the meaning of words in conceptual dependency (CD) structures.
- The parsing is similar to case grammar.
- CD usually provides a greater degree of predictive power.

Discourse and Pragmatic processing

- There are a number of important relationships that may hold between phrases and parts of their discourse contexts, including:
- Identical entities. Consider the text:
 - Bill had a red balloon.
 - John wanted it.
 - The word “it” should be identified as referring to red balloon. This type of references are called anaphora.
- Parts of entities. Consider the text:
 - Sue opened the book she just bought.
 - The title page was torn.
 - The phrase “title page” should be recognized as part of the book that was just bought.

Discourse and pragmatic processing

- Parts of actions. Consider the text:
 - John went on a business trip to New York.
 - He left on an early morning flight.
 - Taking a flight should be recognized as part of going on a trip.
- Entities involved in actions. Consider the text:
 - My house was broken into last week.
 - They took the TV and the stereo.
 - The pronoun “they” should be recognized as referring to the burglars who broke into the house.
- Elements of sets. Consider the text:
 - The decals we have in stock are stars, the moon, item and a flag.
 - I’ll take two moons.
 - Moons means moon decals

Discourse and Pragmatic processing

- Names of individuals:
 - Dave went to the movies.
- Causal chains
 - There was a big snow storm yesterday.
 - The schools were closed today.
- Planning sequences:
 - Sally wanted a new car
 - She decided to get a job.
- Illocutionary force:
 - It sure is cold in here.
- Implicit presuppositions:
 - Did Joe fail CS101?

Discourse and Pragmatic processing

- We focus on using following kinds of knowledge:
 - The current focus of the dialogue
 - A model of each participant's current beliefs
 - The goal-driven character of dialogue
 - The rules of conversation shared by all participants.