18CSE359T – Natural Language Processing

"Enabling Computers to Understand Natural Language like Humans"

School of computing

SRM Institute of Science and Technology

UNIT II Topics

- Syntax Parsing
- Dependency Parsing
- Semantics
- Semantic Parsing
- Word sense Disambiguation
- Lexical Disambiguation
- Structural Disambiguation
- Word, Context and Sentence-level Semantics
- Pronoun Solution
- Semantic Representation of text
- introduction to Semantic Relations

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S1-Syntax Parsing

Syntax parser

- **Syntax Parsing** is the process of analyzing a sentence, breaking it down into smaller components, and identifying the grammatical structure of the sentence.
- Parse is to "resolve a sentence into its component parts and describe their syntactic roles."
- The purpose of this phase is **to draw exact meaning**, or you can say dictionary meaning from the text.
- Syntax analysis checks the text for meaningfulness comparing to the rules of formal grammar.
- For example, the sentence like "hot ice-cream" would be rejected by semantic analyzer.
- In other words, parsing is the process of analyzing a sentence's syntax and its underlying structure to extract meaning from it.

Syntax Parsing

- Syntactic parsing deals with syntactic structure of a sentence.
- The word 'syntax' refers to the grammatical arrangement of words in a sentence and their relationship with each other.
- The objective of syntactic analysis is to find syntactic structure of a sentence which is usually depicted as a tree.
- Identifying the syntactic structure is useful in determining the meaning of a sentence.

Syntax parser

- **Example sentence with syntax error**
  - Her goes to the store yesterday
  - The car red
  - I like to dancing
  - She read the book quick
  - I goes to the park every day
  - The cat sleep on the couch
  - She run to the store yesterday

- A syntax error occurs when a sentence or phrase violates the grammatical rules and structure of the language.
- Examples of syntax errors in NLP include incorrect word order, missing punctuation, or improperly formed sentences

Functions of syntax parser

- **Grammatical Structure Identification:** The primary function of a syntax parser is to **identify the grammatical structure of a sentence.** It determines the roles of different words and phrases, such as subjects, verbs, objects, and modifiers, and how they are organized in relation to each other.
- **Dependency Relationships:** A syntax parser identifies the **dependency relationships** between words in a sentence. It determines which words are connected and how they depend on each other to form meaningful sentence.
- **Syntax Trees and Graphs:** It constructs syntax trees or dependency graphs that visually represent the hierarchical structure and relationships among the words in a sentence. These visual representations aid in understanding the sentence's composition.
- **Error Detection:** Syntax parsers can identify grammatical errors and inconsistencies in sentences, such as subject-verb agreement, incorrect word order, and missing or extra components.
- **Parsing Ambiguity Resolution:** Language often has ambiguities, where a sentence can be interpreted in multiple ways. A syntax parser helps resolve such ambiguities by selecting the most likely grammatical interpretation based on the given context.
- **Language Understanding**: Its a fundamental step in language understanding. It provides a foundation for higher-level language tasks such as sentiment analysis, named entity recognition, and information extraction by determining how words interact in sentences.
- **Machine Translation:** it ensures that the translated text retains the intended meaning.

Grammer Rewrite Rules

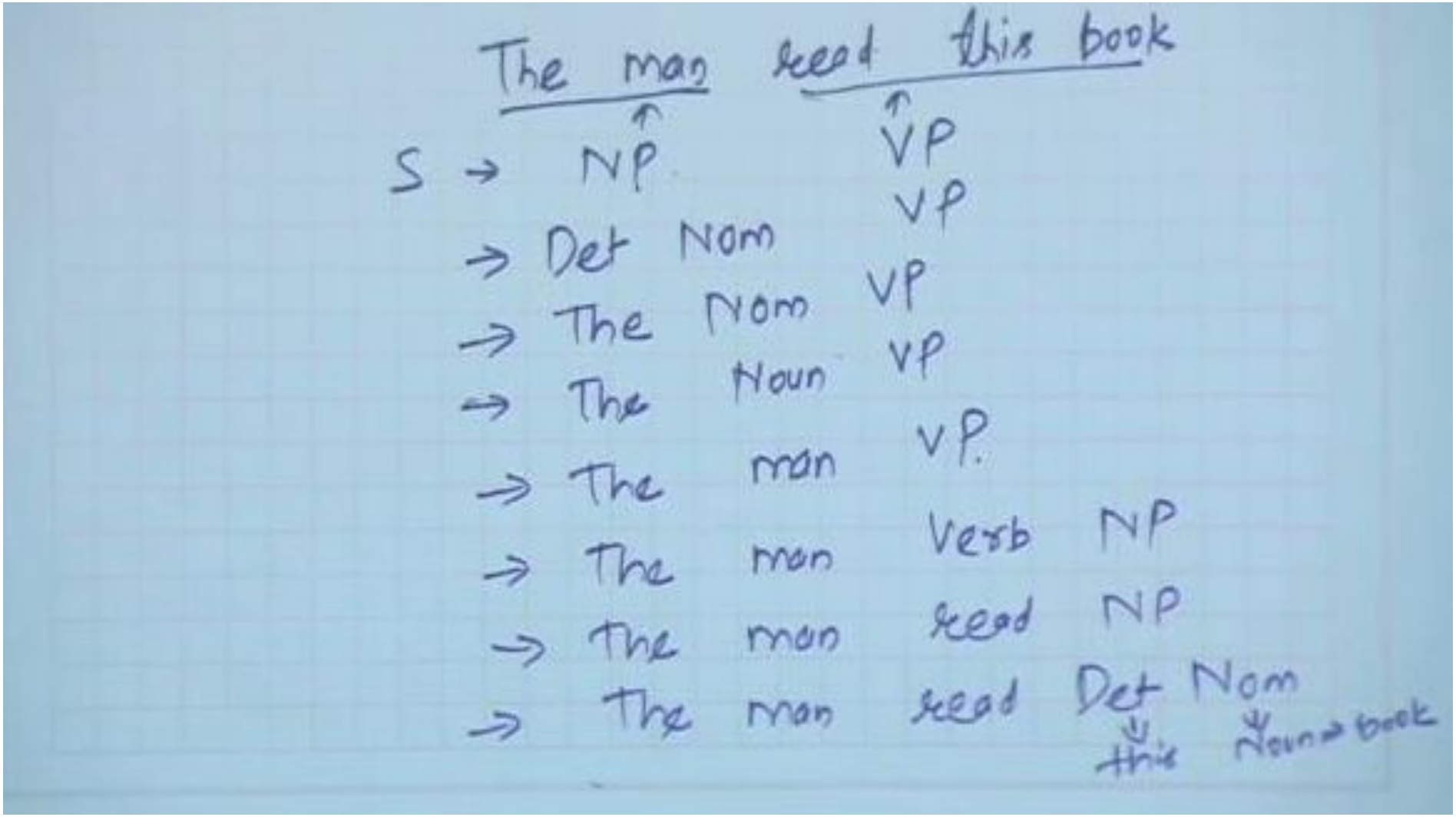
Grammar Rewrite Rules

S → NP VP
S → Aux NP VP
S → VP
NP → Det NOM
NOM → Noun
NOM → Noun NOM
VP → Verb
VP → Verb NP

Det → *that* | *this* | *a* | *the*
Noun → *book* | *flight* | *meal* | *man*
Verb → *book* | *include* | *read*
Aux → *does*

S → NP VP
→ Det NOM VP
→ *The* NOM VP
→ *The* Noun VP
→ *The man* VP
→ *The man* Verb NP
→ *The man read* NP
→ *The man read* Det NOM
→ *The man read this* NOM
→ *The man read this* Noun
→ *The man read this book*

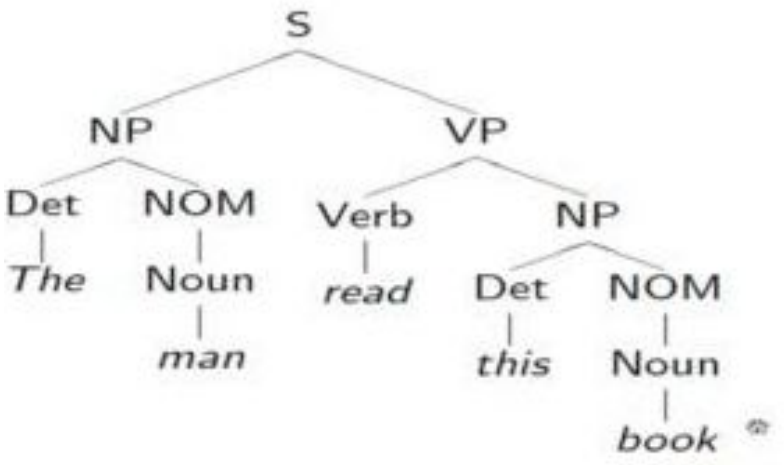
Example


The man read this book
S → NP VP
→ Det Nom VP
→ The Nom VP
→ The Noun VP
→ The man VP
→ The man Verb NP
→ The man read NP
→ The man read Det Nom
this
Noun → book

Parse Trees

Parse Tree

S → NP VP
→ Det NOM VP
→ *The* NOM VP
→ *The* Noun VP
→ *The man* VP
→ *The man* Verb NP
→ *The man read* NP
→ *The man read* Det NOM
→ *The man read this* NOM
→ *The man read this* Noun
→ *The man read this book*



Syntax parser

- The purpose of this phase is to **draw exact meaning**, or you can say dictionary meaning from the text.
- Syntax analysis checks the **text for meaningfulness comparing to the rules of formal grammar**.
- For example, the sentence like “Give me hot ice-cream” would be rejected by semantic analyzer.
- In other words, parsing is the process of analyzing a sentence’s syntax and its underlying structure to extract meaning from it.

Syntax parser

- The **goal** of syntax parsing is to break down a sentence into its constituent parts (such as words, phrases, and clauses) and
- Establish the hierarchical relationships between these parts.
- This process helps computers understand the underlying grammatical structure of the text,
- This is important for various NLP applications such as
  - Machine translation,
  - Information retrieval,
  - Sentiment analysis,
  - Question answering.

Parsing

What is Parsing?

- The process of taking a string and a grammar and returning all possible parse trees for that string
- That is, find all trees, whose root is the start symbol S, which cover exactly the words in the input

What are the constraints? "book that flight"

- There must be three leaves, *book*, *that* and *flight*
- The tree must have one root, the start symbol S
- Give rise to two search strategies: *top-down* (goal-oriented) and *bottom-up* (data-directed)

Example

Parsing

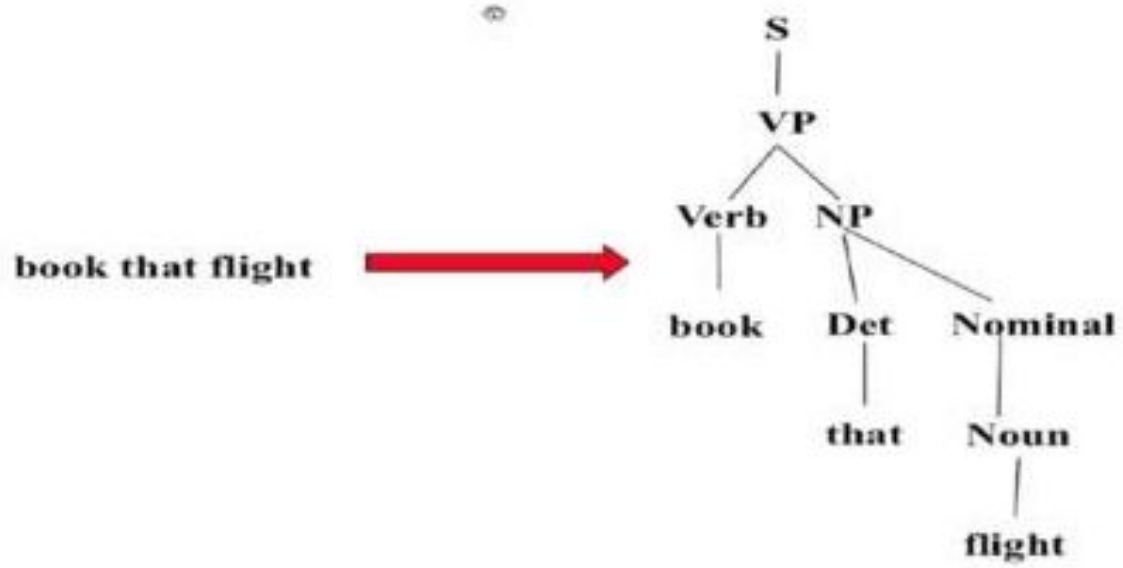
Grammar

S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → VP PP
PP → Prep NP

Lexicon

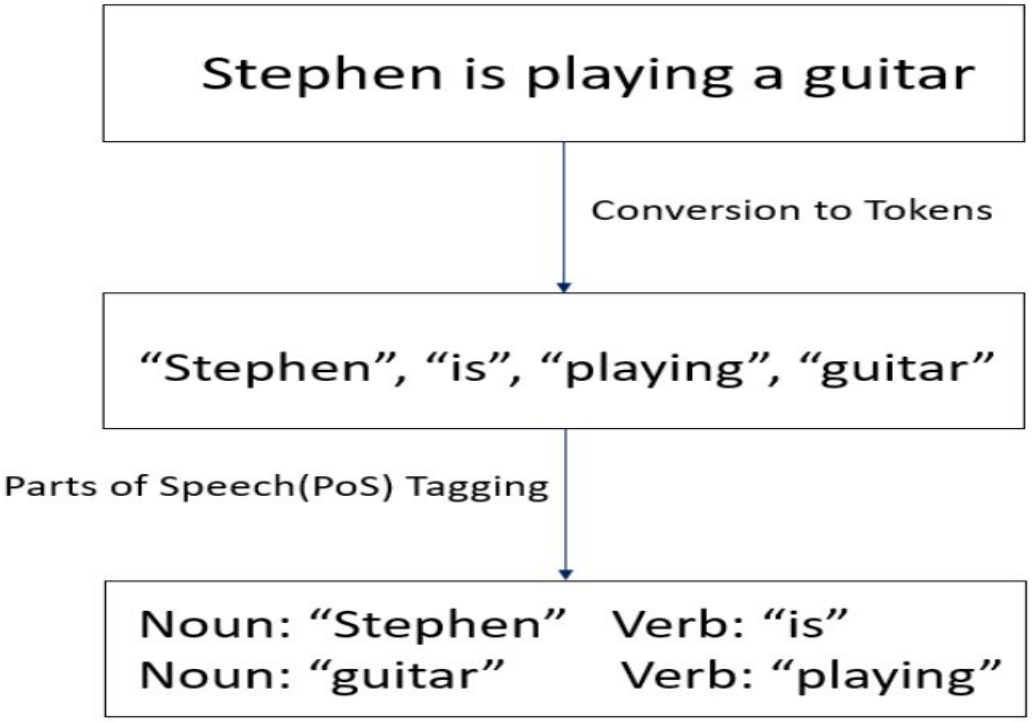
Det → the | a | that | this
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | he | she | me
Proper-Noun → Houston | NWA
Aux → does
Prep → from | to | on | near | through

Parsing


book that flight
S
VP
Verb
NP
book
Det
Nominal
that
Noun
flight

Example of syntax parsing

- The sentence “Stephen is playing guitar”.
- Once we parse this sentence, it will be stated into individual constituents as “Stephen”, “is”, “playing”, and “guitar”.



Example of syntax parsing

- In the above sentence, parsing works by first breaking it down to individual tokens
- i.e., “Stephen”, “is”, “playing”, and “guitar” which are nothing but individual words making up the sentence.
- In the next steps, part of speech is tagged like a Noun tagged to “Stephen” and “guitar”
- Whereas Verb is tagged to “is” and “playing”.
- As per the above example, it is evident that parsing a natural language sentence involves analyzing the input sentence by breaking it down into its grammatical constituents,
- Identifying the parts of speech, and syntactic relations.

Challenges in Syntax Parsing

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

- Syntax parsing is a challenging task due to the ambiguity and complexity of natural language.
- Many sentences can have multiple valid parse trees or dependency structures.
- Determining the correct one requires understanding the **context and semantics** of the sentence.

Types of Syntax Parsing

Derivation divides parsing into the followings two types −

- **Top-down Parsing**
  - In this kind of parsing, the parser starts constructing the parse tree from the start symbol and then tries to transform the start symbol to the input.
  - The most common form of top down parsing uses recursive procedure to process the input.
  - The main disadvantage of recursive descent parsing is **backtracking.**
- **Bottom-up Parsing**
  - In this kind of parsing, the parser starts with the input symbol and tries to construct the parser tree up to the start symbol.

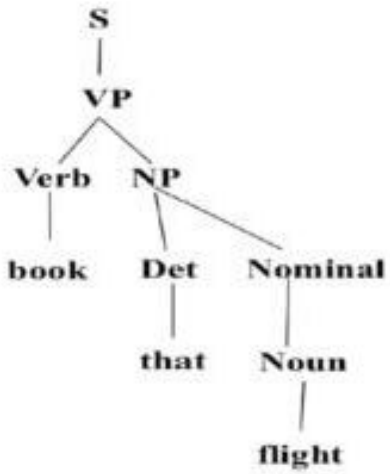
Top Down Parsing

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Top-Down Parsing

- Searches for a parse tree by trying to build upon the root node S down to the leaves
- Start by assuming that the input can be derived by the designated start symbol S
- Find all trees that can start with S, by looking at the grammar rules with S on the left-hand side
- Trees are grown downward until they eventually reach the POS categories at the bottom
- Trees whose leaves fail to match the words in the input can be rejected

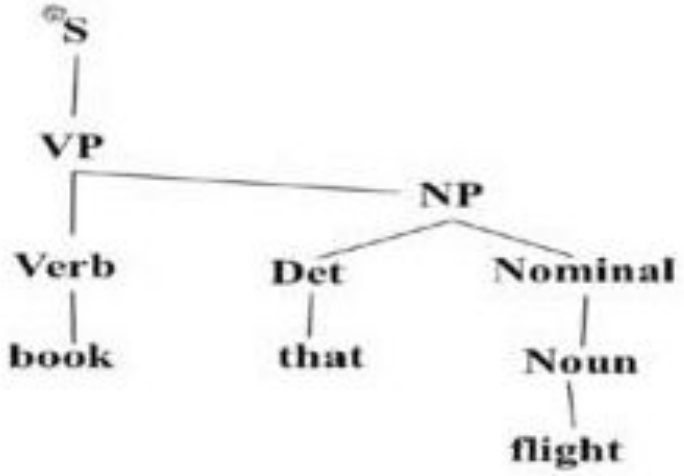
Top-Down Parsing



Bottom-Up Parsing

- The parser starts with the words of the input, and tries to build trees from the words up, by applying rules from the grammar one at a time
- Parser looks for the places in the parse-in-progress where the right-hand-side of some rule might fit.

Bottom-Up Parsing



Top-Down vs. Bottom-Up

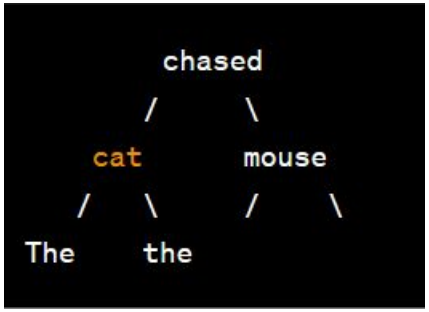
- Top down never explores options that will not lead to a full parse, but can explore many options that never connect to the actual sentence.
- Bottom up never explores options that do not connect to the actual sentence but can explore options that can never lead to a full parse.
- Relative amounts of wasted search depend on how much the grammar branches in each direction.

Types of syntax parser

(Approaches to analyzing the syntactic structure of sentences in natural language processing)

1. Constituency Parsing

- The output of constituency parsing is a **syntax tree, also known as a parse tree or constituent tree.**
- In this tree, the sentence is broken down into hierarchical constituents (phrases) and their sub constituents, with non-terminal nodes representing phrases and terminal nodes representing individual words.
- The syntax tree visually represents the hierarchical structure of the sentence in terms of phrases and their relationships.

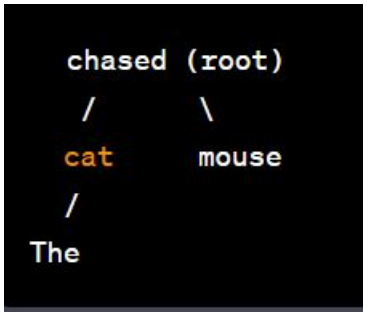


Types of syntax parser

(Approaches to analyzing the syntactic structure of sentences in natural language processing)

2. Dependency Parsing

- The output of dependency parsing is a dependency tree, where **each word in the sentence is a node**, and the relationships (dependencies) between words are represented by directed edges.
- Each edge typically has a label that indicates the grammatical relationship (dependencies) between the words.
- The dependency tree provides insights into how words are connected and the roles they play in the sentence's structure.



Concept of Derivation

- In order to get the input string, we need a sequence of production rules. **Derivation is a set of production rules.** During parsing, we need to decide the non-terminal, which is to be replaced along with deciding the production rule with the help of which the non-terminal will be replaced.
- **Types of Derivation**
  - There are two types of derivations, which can be used to decide which non-terminal to be replaced with production rule −
- **Left-most Derivation**
  - In the left-most derivation, the sentential form of an input is scanned and replaced from the left to the right. The sentential form in this case is called the left-sentential form.
- **Right-most Derivation**
  - In the left-most derivation, the sentential form of an input is scanned and replaced from right to left. The sentential form in this case is called the right-sentential form.

Concept of Parse Tree

- Parse tree is a graphical depiction of a derivation. The start symbol of derivation serves as the root of the parse tree. In every parse tree, the leaf nodes are terminals and interior nodes are non-terminals. A property of parse tree is that in-order traversal will produce the original input string.

- Syntax parsing aims to create a parse tree or syntactic tree, which is a hierarchical representation of the sentence's grammatical structure.

- This tree depicts the relationships between words, phrases, and clauses in the sentence, showing how they are connected through various grammatical rules and syntactic patterns in visual form.

Grammer of syntax parser

- A sentence is structured as follows:
- **Sentence = S = Noun Phrase + Verb Phrase + Preposition Phrase**
- S = NP + VP + PP

- The different word groups that exist according to English grammar rules are:
- **Noun Phrase(NP):** Determiner + Nominal Nouns = DET + Nominal

- **Verb Phrase (VP):** Verb + range of combinations

- **Prepositional Phrase (PP):** Preposition + Noun Phrase = P + NP

- We can make different forms and structures versions of the noun phrase, verb phrase, and prepositional phrase and join in a sentence.

Example of syntax parsing

- Below is a parse tree for the sentence “The thief robbed the apartment.” Included is a description of the three different information types conveyed by the sentence.

1) Part of speech

N = noun

V = verb

DT = determiner

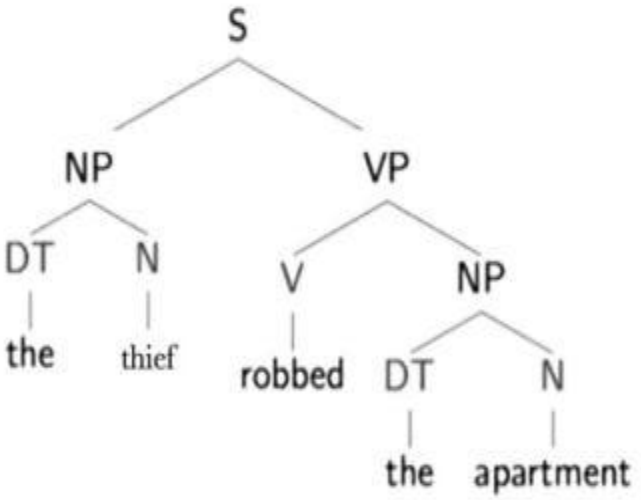
2) Phrases

Noun Phrases: : “the thief”, “the apartment”

Verb Phrases: “robbed the apartment”

Sentence: “the burglar robbed the apartment”

3) Relationships



Example of syntax parsing

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

- The letters directly above the **single words show the parts of speech** for each word (noun, verb and determiner).
- One level higher is some **hierarchical grouping of words into phrases**.
- For example, “the thief” is a **noun phrase**, “robbed the apartment” is a **verb phrase** and when put together the two phrases form a **sentence**, which is marked one level higher.
- But what is actually meant by a noun or verb phrase?
  - Noun phrases are one or more words that contain a noun and maybe some descriptors, verbs or adverbs.
  - The idea is to group nouns with words that are in relation to them.

Example of syntax parsing

- A parse tree also provides us with information about the **grammatical relationships of the words** due to the structure of their representation. For example, we can see in the structure that “the thief” is the subject of “robbed.”

- With structure I mean that we have the verb (“robbed”), which is marked with a “V” above it and a “VP” above that, which is linked with a “S” to the subject (“the thief”), which has a “NP” above it. This is like a template for a subject-verb relationship and there are many others for other types of relationships.

Types of syntax parser

1. Constituency parsing

2. Dependency parsing.

Syntax parsing involves several key concepts

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

1. Dependency Parsing: Dependency parsing focuses on representing the relationships between words in a sentence by creating a tree-like structure where each word is a node, and the relationships between words are represented as directed edges (dependencies). Each word has a specific grammatical role and relationship with other words in the sentence.
2. Constituency Parsing: Constituency parsing involves identifying the grammatical constituents (such as noun phrases, verb phrases, and clauses) that make up a sentence. The parsing process produces a tree structure where each node represents a constituent, and the tree's leaves correspond to individual words.

Output of Syntax parser

Aspect	Syntax Tree	Dependency Graph
Representation	Hierarchical structure of phrases and words	Grammatical relationships and dependencies between words
Structure	Highlights hierarchical arrangement of phrases and words	Shows links between words based on grammatical roles and relationships
Visualization	Upside-down tree with branching branches	Arrows connecting words to indicate dependency or relationship
Usage	Formal linguistic analysis of phrase structure	Computational linguistics, parsing, machine translation, sentiment analysis
Focus	Structure and organization of sentence elements	Function of words in relation to each other
Analytical Insights	Syntactic structure, constituents' grouping	Grammatical roles, word dependencies
Common Representation	Tree-like diagram with branching nodes	Graph with nodes and directed edges

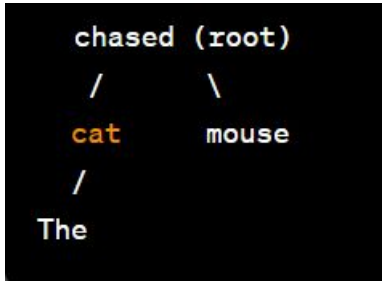
2-Dependency parsing

Dependency Parsing

- The Dependency Parsing refers to the process of examining the dependencies between the phrases of a sentence to determine its grammatical structure.
- It aims to determine the **syntactic dependencies** between words in a sentence, representing how they are connected based on their grammatical roles.
- **Structure of dependency parser:**
  - In a dependency parse, each word in the sentence is considered a **node**, and the relationships between words are represented as **directed edges** between these nodes.
  - Each relationship has one **head** and a **dependent** that modifies the head.
  - Each edge is **labeled** with a grammatical relationship or dependency type that describes the syntactic connection between the dependent and the head.

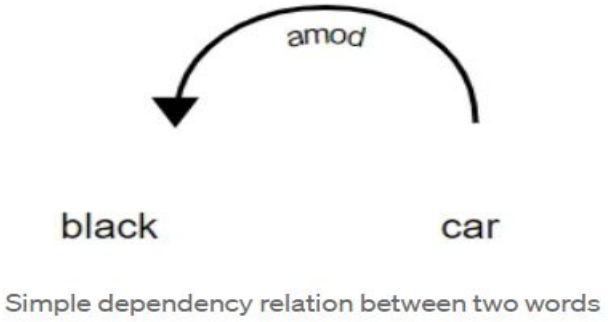
Example of Dependency Parsing

- **Sentence: "The cat chased the mouse“**



- "chased" is the root of the parse tree since it's the main verb of the sentence.
- "cat" is a dependent of "chased" and is connected by the "subject" relationship. This means "cat" is the subject of the verb "chased."
- "mouse" is also a dependent of "chased" and is connected by the "object" relationship. This means "mouse" is the object of the verb "chased."
- "The" is a dependent of "cat" and is connected by the "determiner" relationship. This means "The" is modifying the noun "cat."

Example of Dependency Parsing

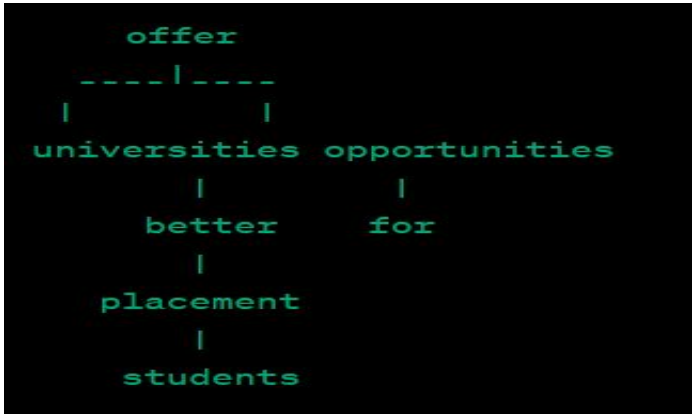


Simple dependency relation between two words

- In the above diagram, there exists a relationship between **car and black** because black modifies the meaning of car.
- Here, car acts as the head and black is a dependent of the head.
- The nature of the relationship here is amod which stands for “Adjectival Modifier”.
- It is an adjective or an adjective phrase that modifies a noun.

Example of Dependency Parsing

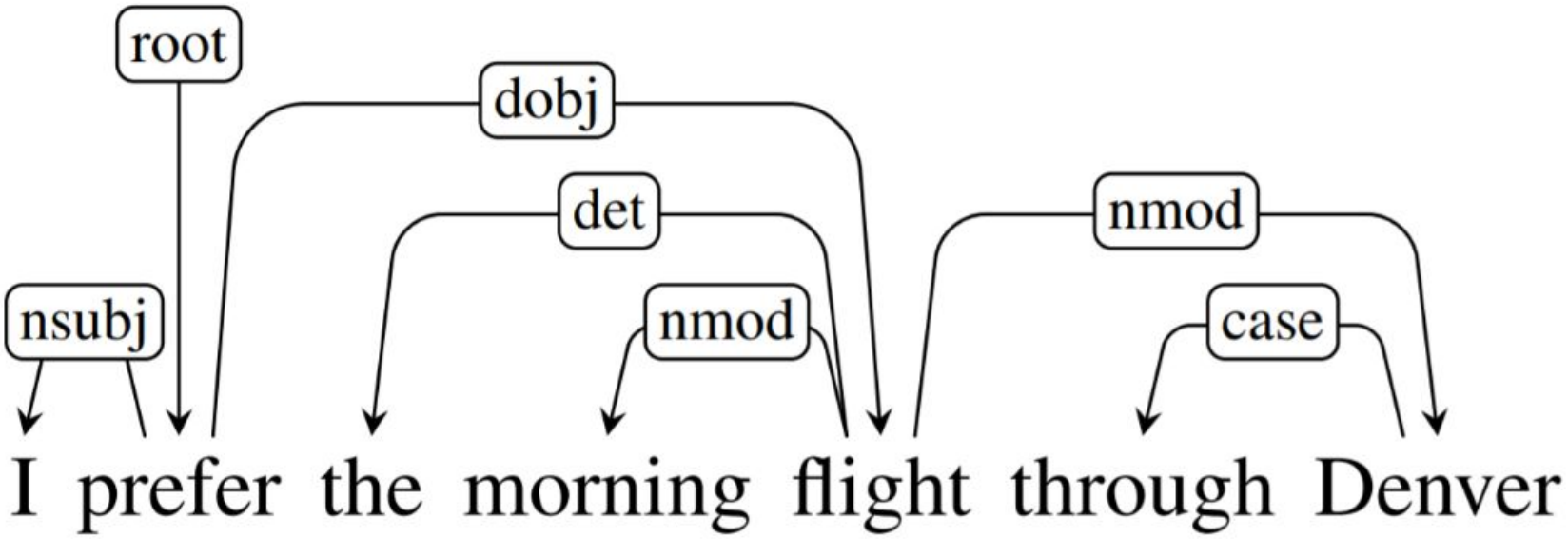
- **Ex: Universities offer better placement opportunities for students**



Word	Dependency Relationship	Dependent
offer	ROOT	
universities	nsubj	offer
opportunities	dobj	offer
better	amod	opportunities
for	prep	opportunities
placement	compound	opportunities
students	pobj	for

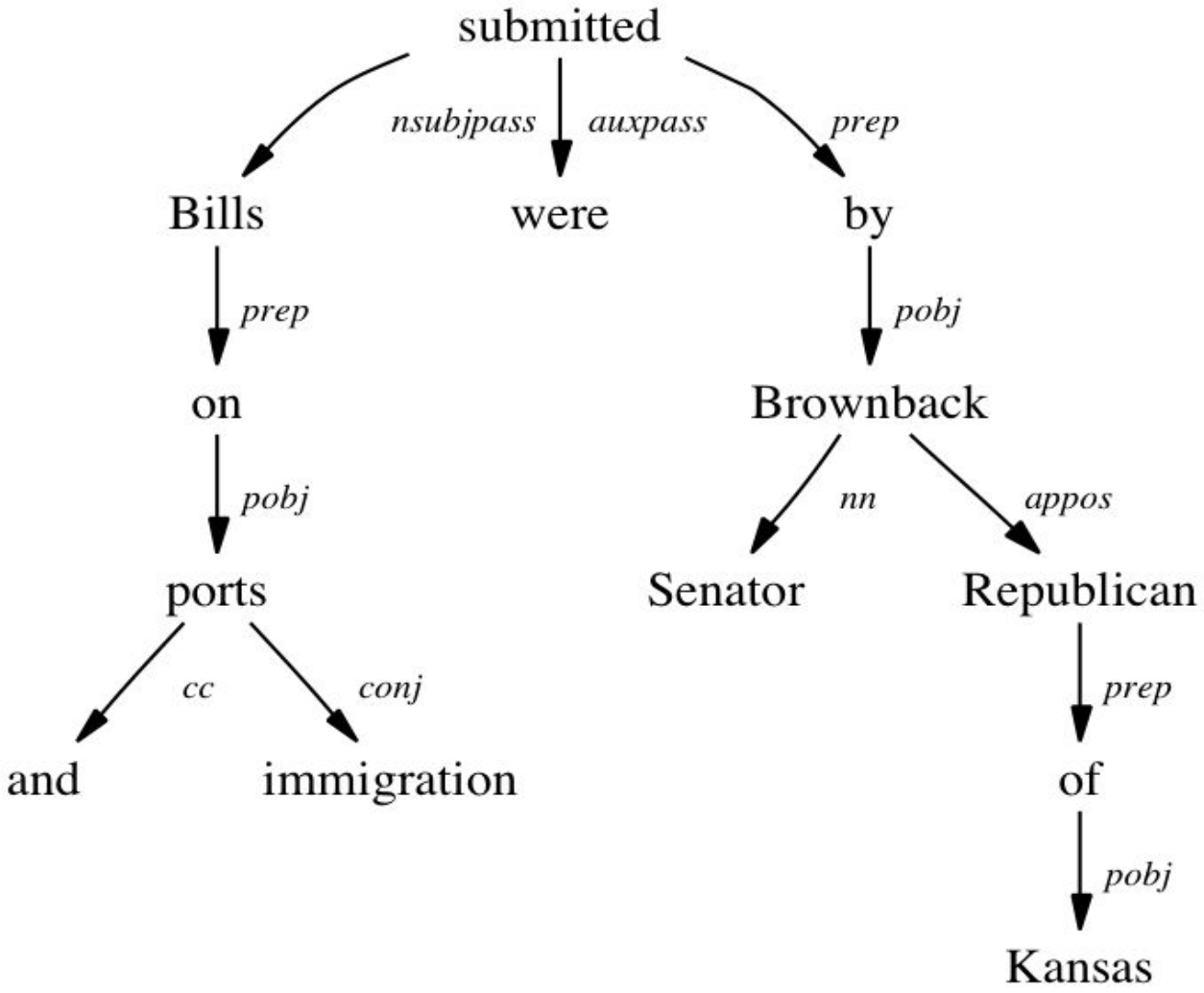
These labels can be found at Universal Dependency Relations.

Example of Dependency Parsing



Example of Dependency Parsing

- **Sentence:** Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas



Implementation of Dependency Parsing

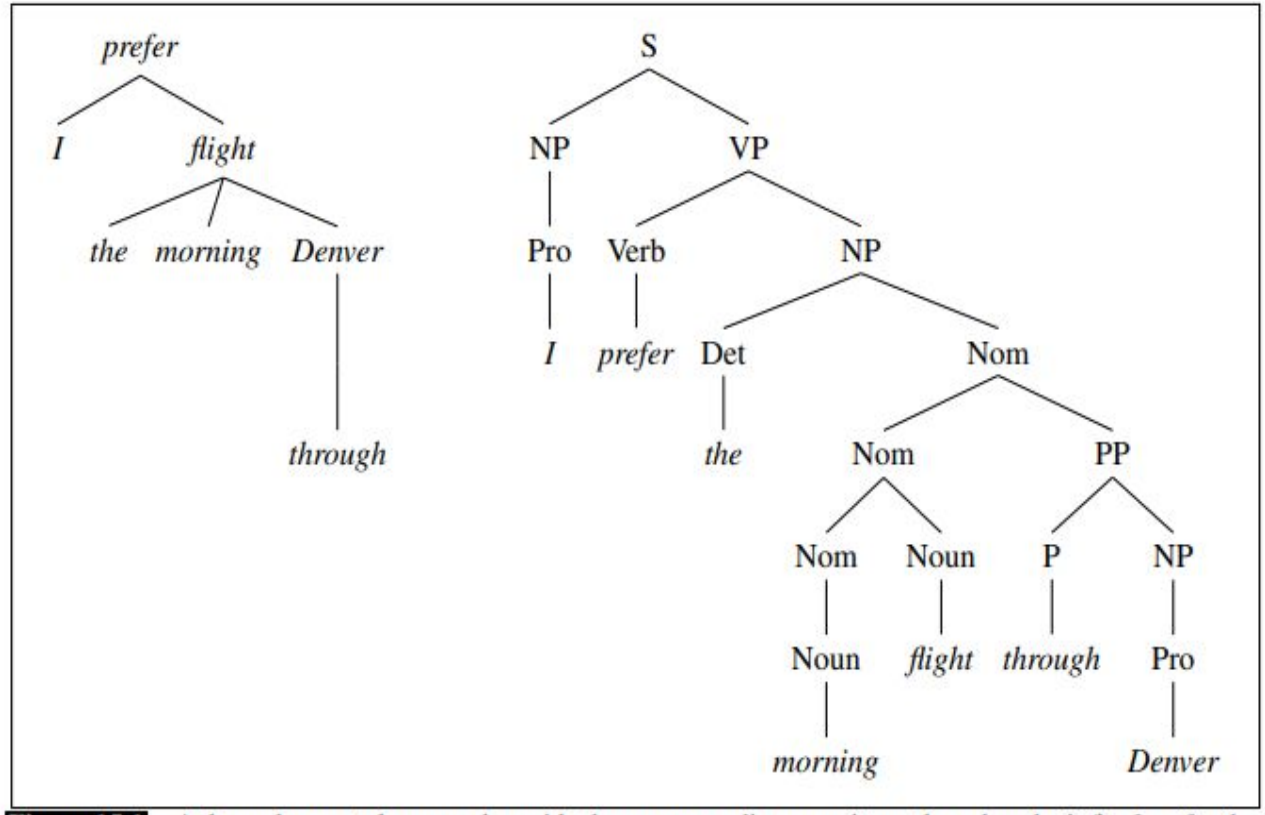


Figure 15.1 A dependency-style parse alongside the corresponding constituent-based analysis for *I prefer the morning flight through Denver.*

Dependency Parsing Relations

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 15.2 Selected dependency relations from the Universal Dependency set. (de Marneffe et al., 2014)

Currently, the Common Dependency V2 taxonomy consists of 37 common syntactic relationships, as shown in the table below:

https://www.analyticsvidhya.com/blog/2021/12/dependency-parsing-in-natural-language-processing-with-examples/

Implementation of Dependency Parsing

- Method 1: Using spaCy
- Method 2: Using NLTK with Stanford CoreNLP
  - Visualize using NetworkX
- Method 3: Using Stanza

UPenn TreeBank POS tag set

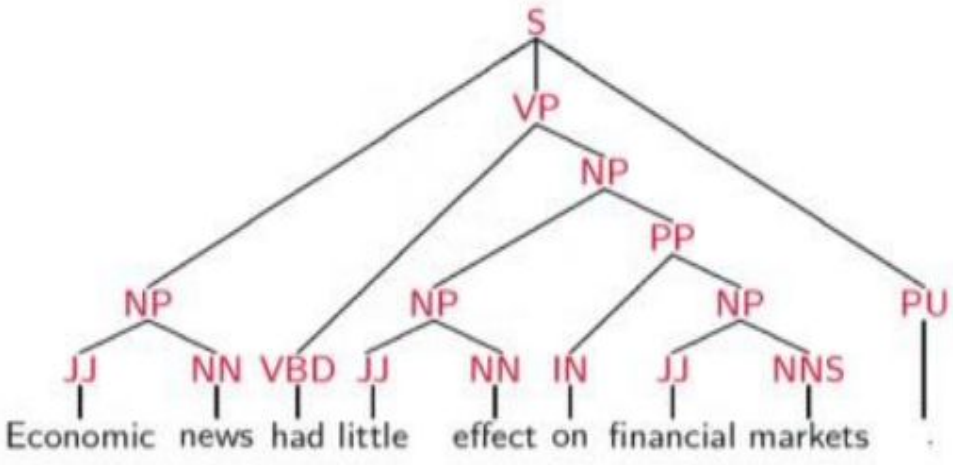
Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Dependency Grammars and Parsing

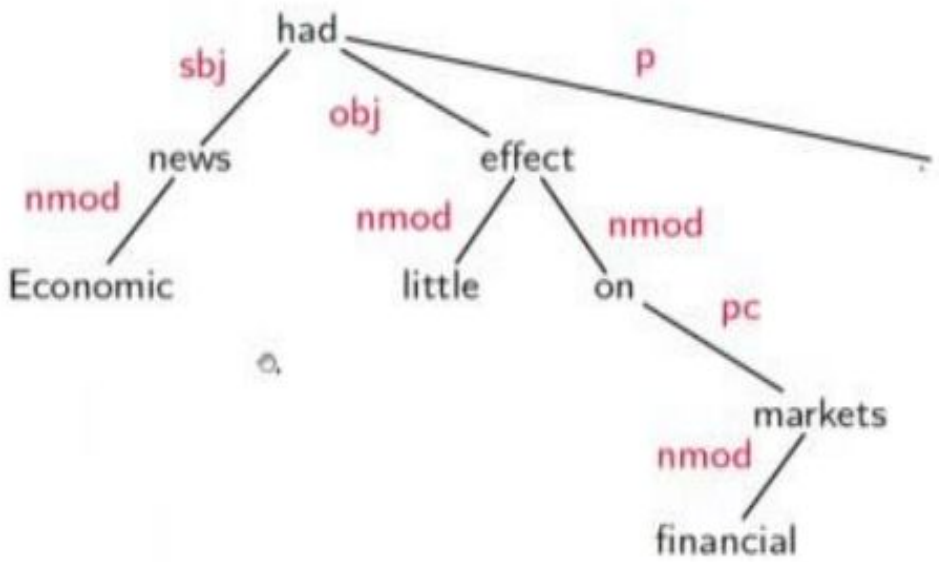
- 2 different words are connected with some relationship called dependency structure.

Phrase Structure Representation

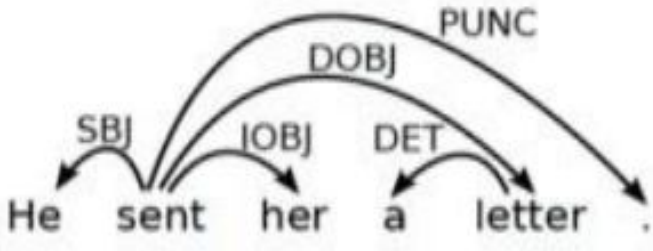
Phrase Structure



Dependency Structure Representation



Dependency Structure

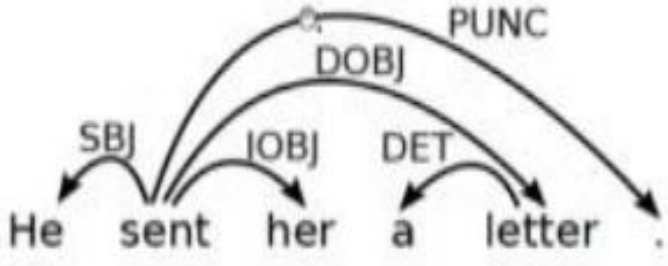


- Connects the words in a sentence by putting arrows between the words.
- Arrows show relations between the words and are typed by some grammatical relations.
- Arrows connect a head (governor, superior, regent) with a dependent

(modifier, inferior, subordinate).

- Usually dependencies form a tree.

Criteria for Heads and Dependents

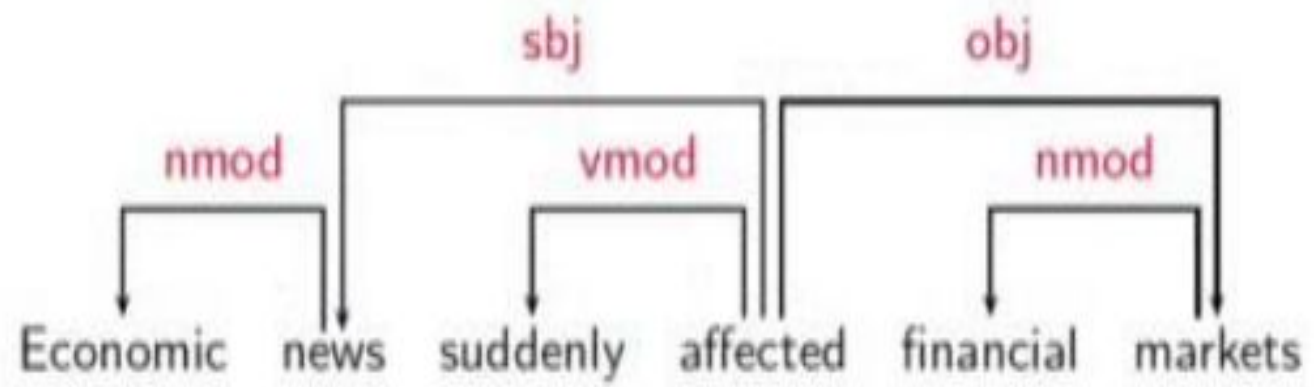


Criteria for a syntactic relation between a head H and a dependent D in a construction C

- H determines the syntactic category of C; H can replace C.
- D specifies H.
- H is obligatory; D may be optional.
- H selects D and determines whether D is obligatory.
- The form of D depends on H (agreement or government).
- The linear position of D is specified with reference to H.

Some Clear Cases

Construction	Head	Dependent
Exocentric	Verb	Subject (sbj)
	Verb	Object (obj)
Endocentric	Verb	Adverbial (vmod)
	Noun	Attribute (nmod)



Comparison

Phrase structures explicitly represent

- Phrases (nonterminal nodes)
- Structural categories (nonterminal labels)

Dependency structures explicitly represent

- Head-dependent relations (directed arcs)
- Functional categories (arc labels)

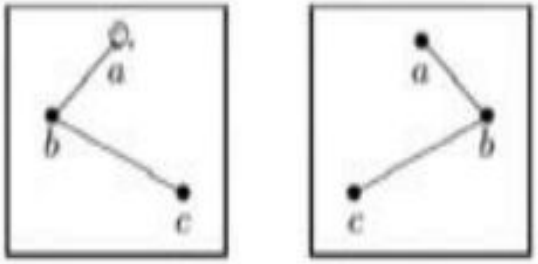
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Dependency Graphs

- A dependency structure can be defined as a directed graph G, of
  - a set V of nodes,
  - a set A of arcs (edges),
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in A are labeled with dependency types.
- Notational convention:
  - Arc (w_i, d, w_j) links head w_i to dependent w_j with label d
  - $w_i \xrightarrow{d} w_j \Leftrightarrow (w_i, d, w_j) \in A$
  - $i \rightarrow j \equiv (i,j) \in A$
  - $i \rightarrow^* j \equiv i = j \vee \exists k : i \rightarrow k, k \rightarrow^* j$

Formal conditions on Dependency Graphs

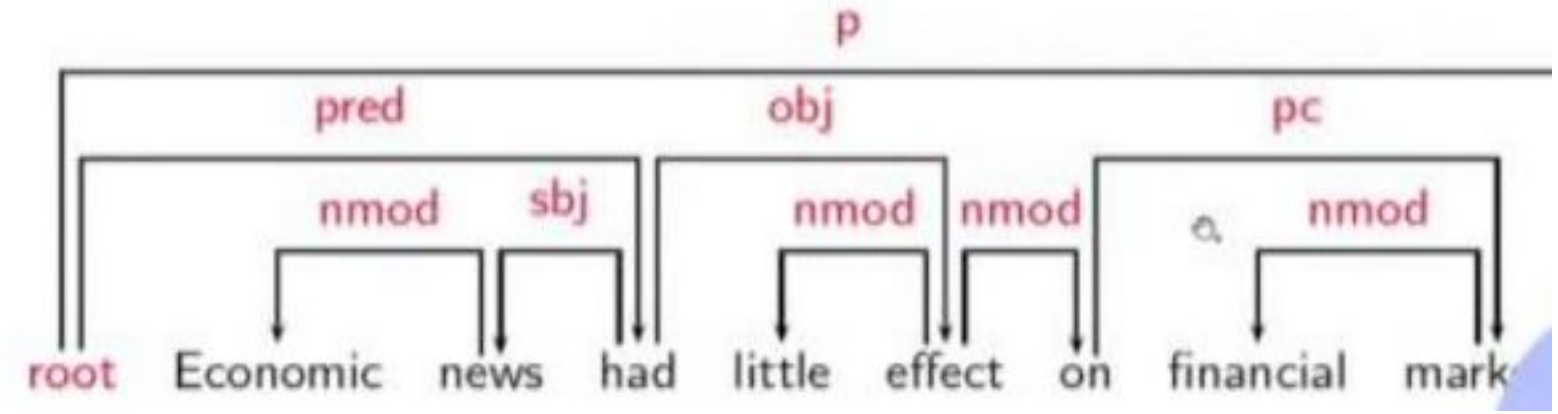
- G is connected:
  - For every node i there is a node j such that $i \rightarrow j$ or $j \rightarrow i$.
- G is acyclic:
  - if $i \rightarrow j$ then not $j \rightarrow^* i$.
- G obeys the single head constraint:
  - if $i \rightarrow j$ then not $k \rightarrow j$, for any $k \neq i$.
- G is projective:
  - if $i \rightarrow j$ then $j \rightarrow^* k$, for any k such that both j and k lie on the same side of i.



Formal Conditions: Basic Intuitions

Connectedness, Acyclicity and Single-Head

- **Connectedness:** Syntactic structure is complete.
- **Acyclicity:** Syntactic structure is hierarchical.
- **Single-Head:** Every word has at most one syntactic head.
- **Projectivity:** No crossing of dependencies.



Dependency Parsing

Dependency Parsing

- **Input:** Sentence $x = w_1, \ldots, w_n$
- **Output:** Dependency graph G

Parsing Methods

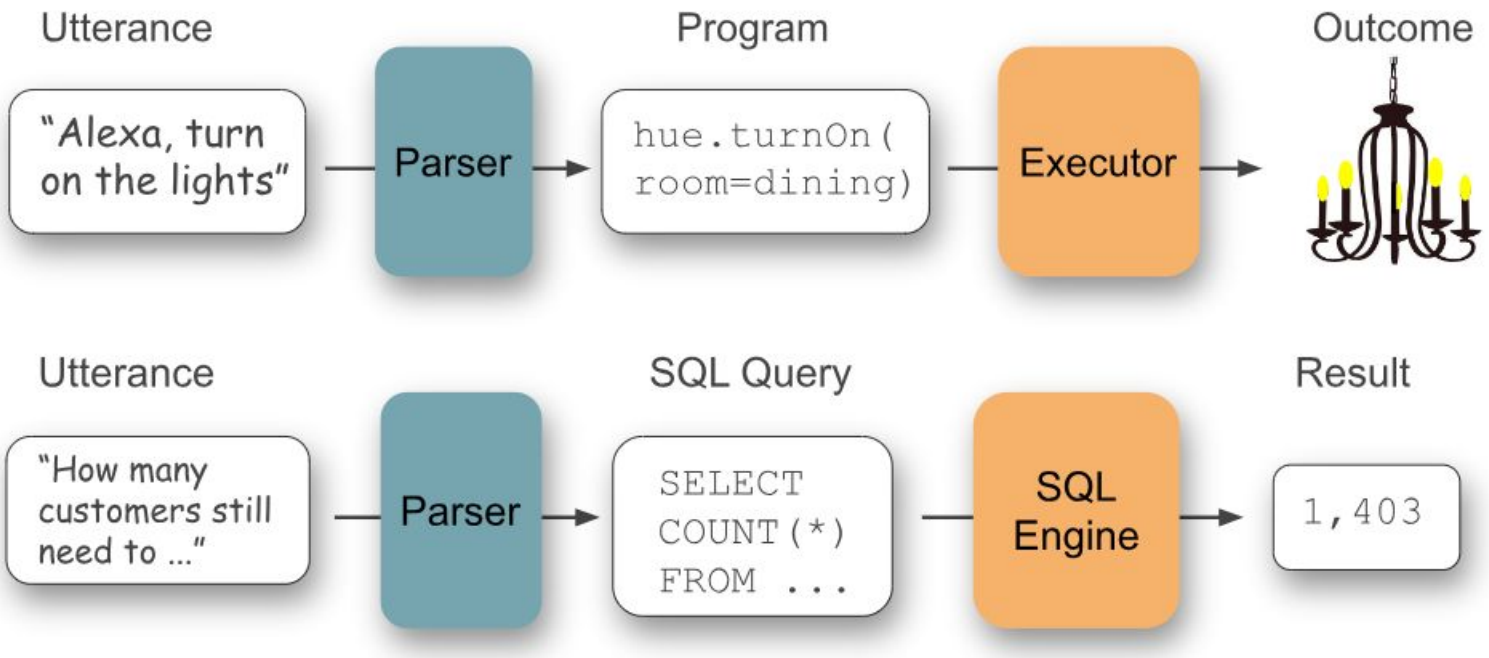
- Deterministic Parsing
- Maximum Spanning Tree Based
- Constraint Propagation Based

S3-S4-Semantic

Semantic

- **Semantic- Meaning**
- Semantic parsing analyze meaning of the text from sentences, phrases, or complete texts.
- It goes beyond grammatical structure and aims to capture the semantics or the actual message that the language conveys.
- The role of this is to detect all the subjective elements in an exchange: approach, positive feeling, dissatisfaction, impatience, etc.
- Example of sentence with semantic error:
  - The cat is a type of dog
  - The sun rises in the west
  - He drank the entire ocean
  - The car sings beautifully

Semantic



- Semantic parsing translating natural language statements into some executable meaning representation.

- Semantic parsers form the backbone of voice assistants, as shown above, or they can be used to answer questions or give natural language interfaces to databases.

How Does Semantic Analysis Work?

- Semantic analysis starts with lexical semantics, which studies individual words' meanings (i.e., dictionary definitions).
- Semantic analysis then examines relationships between individual words and analyzes the meaning of words that come together to form a sentence.
- This analysis provides a clear understanding of words in context.
- Example:
- "The boy ate the apple" defines an apple as a fruit.
- "The boy went to Apple" defines Apple as a brand or store.

Elements of Semantic Analysis

- A semantic system brings entities, concepts, relations and predicates together to provide more context to language so machines can understand text data with more accuracy. To better understand this, consider the following elements of semantic analysis that help support language understanding:
- Hyponymy: It is an instance of a generic term
  - For example: 'Color' is a hypernymy while 'grey', 'blue', 'red', etc, are its hyponyms.
- Homonymy: Two or more lexical terms with the same spelling and different meanings.
  - For example: 'Rose' might mean 'the past form of rise' or 'a flower', – same spelling but different meanings; hence, 'rose' is a homonymy.
- Polysemy: Two or more terms that have the same spelling but multiple closely related meanings.
  - It differs from homonymy because the meanings of the terms need not be closely related in the case of homonymy. For example: 'man' may mean 'the human species' or 'a male human' or 'an adult male human' – since all these different

Elements of Semantic Analysis

- Synonymy: Two or more lexical terms with different spellings and similar meanings.
  - For example: (Job, Occupation), (Large, Big), (Stop, Halt)
- Antonymy: A pair of lexical terms with contrasting meanings.
  - they are symmetric to a semantic axis. For example: (Day, Night), (Hot, Cold), (Large, Small)
- Meronomy: A relationship between a lexical term and a larger entity.
  - Meronomy refers to a relationship wherein one lexical term is a constituent of some larger entity. For example: 'Wheel' is a meronym of 'Automobile'

Part of Semantic Analysis

- Lexical analysis is the process of reading a stream of characters, identifying the lexemes and converting them into tokens that machines can read.

- Grammatical analysis correlates the sequence of lexemes (words) and applies formal grammar to them so part-of-speech tagging can occur.

- Syntactical analysis analyzes or parses the syntax and applies grammar rules to provide context to meaning at the word and sentence level.

- Semantic analysis uses all of the above to understand the meaning of words and interpret sentence structure so machines can understand language as humans do.

Syntax vs Semantic

- Semantic" refers to meaning.
- Syntactic analysis focuses on "form" and syntax, meaning the relationships between words in a sentence.
- Semantic analysis focuses on "meaning," or the meaning of words together and not just a single word.
- "parsing" means resolving a sentence into its component parts.

For example, we can build a parser that converts the natural language query "*Who was the first person to walk on the moon?*" to an equivalent (although complex!) SQL query such as "SELECT name FROM Person WHERE moon_walk = true ORDER BY moon_walk_date FETCH first 1 rows only."

Semantic parsing is inherently more complicated than syntactic parsing because it requires understanding concepts from different word phrases. For instance, the following sentences (adapted from [4]) should ideally map to the same formal representation.

Semantics

What is Semantics?

The study of meaning: Relation between symbols and their denotata.

John told Mary that the train moved out of the station at 3 o'clock.

Computational Semantics

Computational Semantics

The study of how to automate the process of constructing and reasoning with meaning representations of natural language expressions.

Methods in Computational Semantics generally fall in two categories:

- **Formal Semantics:** Construction of precise mathematical models of the relations between expressions in a natural language and the world.
  John chases a bat $\rightarrow \exists x[bat(x) \wedge chase(john, x)]$
- **Distributional Semantics:** The study of statistical patterns of human word usage to extract semantics.

Semantic Parsing

Lexical Semantics

Definition

Lexical semantics is concerned with the systematic meaning related connections among lexical items, and the internal meaning-related structure of individual lexical items.

To identify the semantics of lexical items, we need to focus on the notion of **lexeme**, an individual entry in the lexicon.

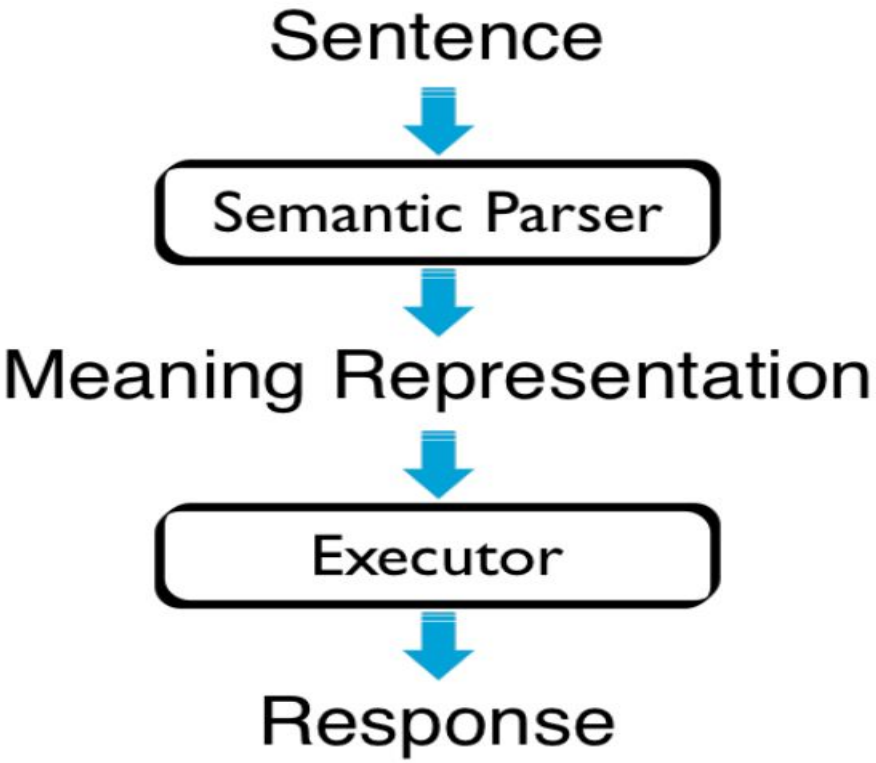
What is a lexeme?

Lexeme should be thought of as a pairing of a particular orthographic and phonological form with some sort of symbolic meaning representation.

- Orthographic form, and phonological form refer to the appropriate form part of a lexeme
- Sense refers to a lexeme's meaning counterpart.

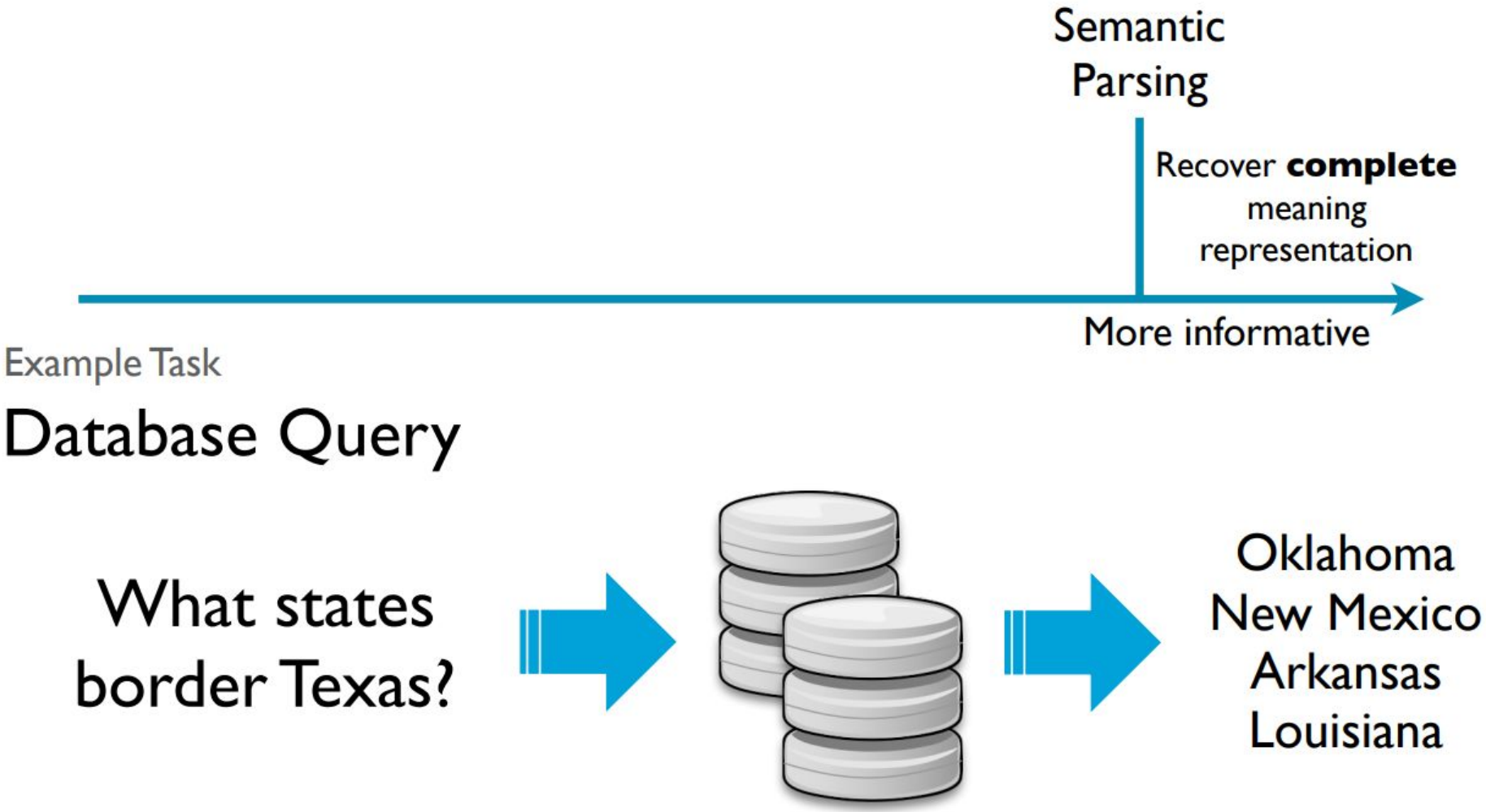
Semantic Parsing

- Translate natural language utterances (NLUs) to meaning representation (MR)
- f : sentence -> logical form



Need of Semantic Parsing

- It provides language with meaning.



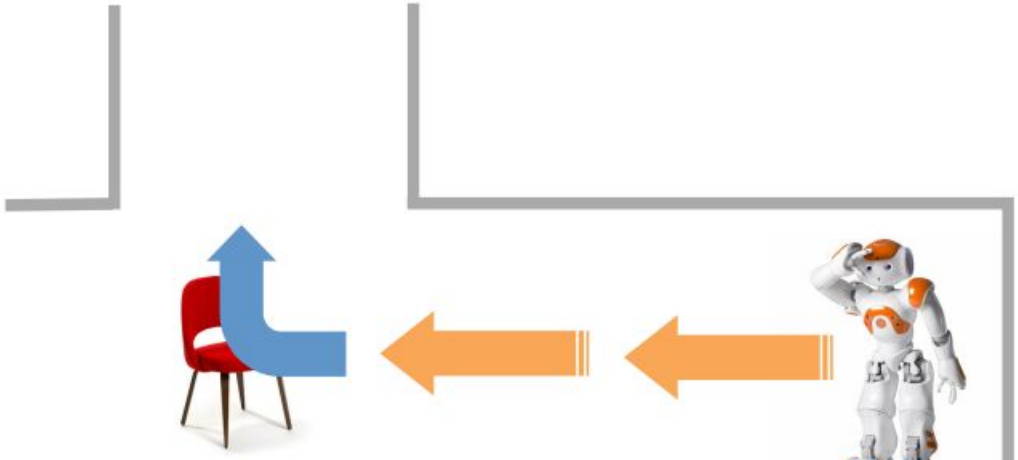
Need of Semantic Parsing

- It provides language with meaning.(Language->Logical)

Example Task

Instructing a Robot

at the chair,
turn right



Need of Semantic Parsing

Complete meaning is sufficient to complete the task

- Convert to database query to get the answer
- Allow a robot to do planning

Example of Semantic Parsing

GEOQuery

This is a standard semantic parsing benchmark which contains 880 queries to a database of U.S. geography.

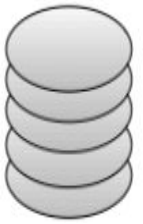
Training Dataset	Test Dataset
600 queries	280 queries

Question:

which state has the most rivers running through it?

Logical form:

```
argmax $0
(state:t $0)
(count $1 (and
        (river:t $1)
        (loc:t $1 $0)))
```



Answer:

Example of Semantic Parsing

ATIS

This dataset has 5,410 queries to a flight booking system.

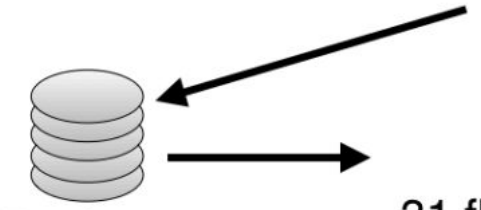
Training Dataset	Development Dataset	Test Dataset
4480 instances	480 instances	450 instances

Request:

Show me flights from Pittsburgh to Seattle

Logical form:

```
lambda $0 e
    (and (flight $0)
    (from $0 pittsburgh:ci)
    (to $0 seattle:ci))
```



Result:

31 flights available

Language to Logical Form

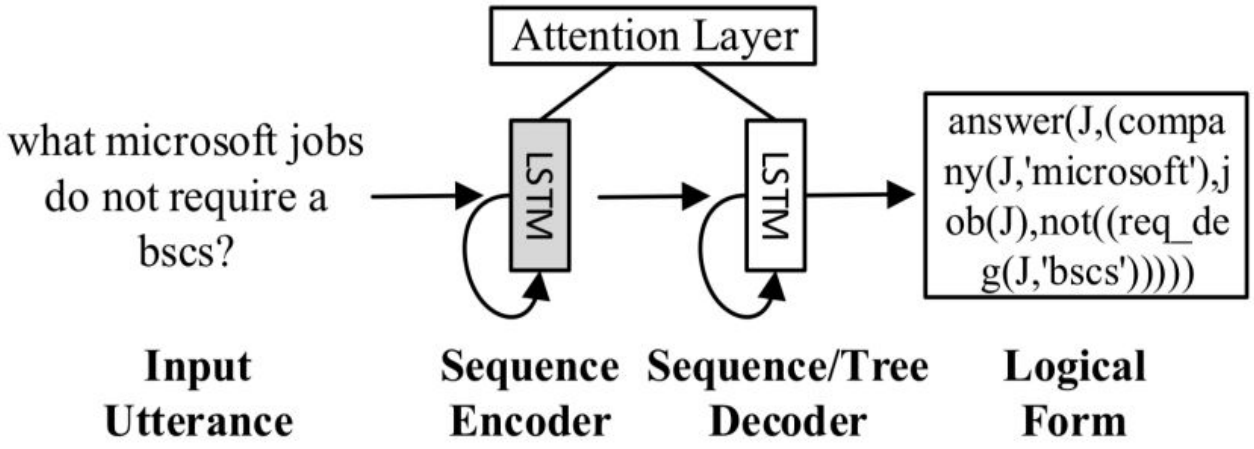
Problem Formulation

- Goal:

  Learn a model which maps natural language input $q = q_1, \ldots, q_{|q|}$ to a logical form representation of its meaning $a = a_1, \ldots, a_{|a|}$.

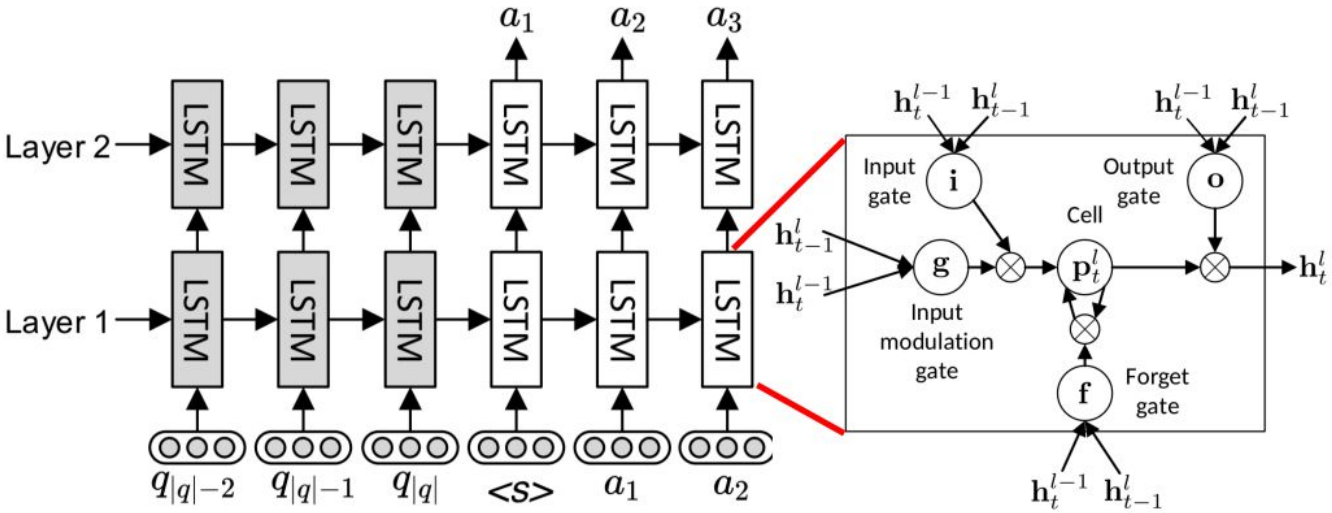
- Conditional probability: $p(a|q) = \prod_{t=1}^{|a|} p(a_t|a_{<t}, q)$

- Framework of Neural Semantic Parsing with Attention



Language to Logical Form

Working Principle of Seq2Seq Neural Semantic Parsing



$$\mathbf{g}_t^0 = \mathbf{W}_q \mathbf{e}(q_t) \qquad \mathbf{h}_t^0 = \mathbf{W}_a \mathbf{e}(a_{t-1})$$

$$\mathbf{W}_q \in \mathbb{R}^{n \times |V_q|} \qquad \mathbf{W}_a \in \mathbb{R}^{n \times |V_a|}$$

$$p(a_t | a_{<t}, q) = \text{softmax}_{a_t}(\mathbf{W}_o \mathbf{h}_t^l)$$

$$\mathbf{W}_o \in \mathbb{R}^{|V_a| \times n}$$

$$\mathbf{h}_t^l = f_{LSTM}(\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1})$$

$$\mathbf{p}_t^l = f \odot \mathbf{p}_{t-1}^l + \mathbf{i} \odot \mathbf{g}$$

$$\mathbf{h}_t^l = \mathbf{o} \odot \tanh(\mathbf{p}_t^l)$$

$$\begin{pmatrix} \mathbf{i} \\ \mathbf{f} \\ \mathbf{o} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} W^l \begin{pmatrix} \mathbf{h}_t^{l-1} \\ \mathbf{h}_{t-1}^l \end{pmatrix}$$

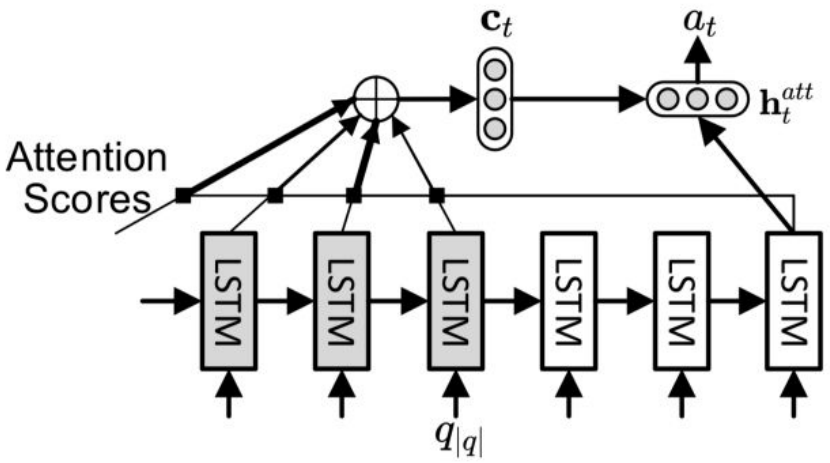
Language to Logical Form

Attention Mechanism for Neural Semantic Parsing

$$\mathbf{c}_t = \sum_{k=1}^{|q|} r_{t,k}\mathbf{h}_k^L$$

$$r_{t,k} \propto exp\{\mathbf{h}_t^L, \mathbf{h}_k^L\}$$

$$\sum_{j=1}^{|q|} r_{i,j} = 1$$



$\mathbf{h}_1^L, ..., \mathbf{h}_{|q|}^L$ are the top layer hidden vectors of the encoder

$$\mathbf{h}_t^{att} = \tanh(\mathbf{W}_1\mathbf{h}_t^L + \mathbf{W}_2\mathbf{c}_t)$$

$$p(a_t|a_{<t}, q) = \text{softmax}_{a_t}(\mathbf{W}_o\mathbf{h}_t^{att})$$

Relations between word meanings

- Homonymy
- Polysemy
- Synonymy
- Antonymy
- Hypernymy
- Hyponymy
- Meronymy

Homonymy

Definition

Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings.

Examples

- Bat (wooden stick-like thing) vs Bat (flying mammal thing)
- Bank (financial institution) vs Bank (riverside)

homophones and homographs

homophones are the words with the same pronunciation but different spellings.

- write vs right
- piece vs peace

homographs are the lexemes with the same orthographic form but different meaning. Ex: bass

Problems for NLP applications

Text-to-Speech

Same orthographic form but different phonological form

Information Retrieval

Different meaning but same orthographic form

Speech Recognition

to, two, too

Perfect homonyms are also problematic

Polysemy

*Multiple **related** meanings within a single lexeme.*

- The *bank* was constructed in 1875 out of local red brick.
- I withdrew the money from the *bank*.

Are those the same sense?

- Sense 1: "The building belonging to a financial institution"
- Sense 2: "A financial institution"

Another example

- Heavy snow caused the roof of the *school* to collapse.
- The *school* hired more teachers this year than ever before.

Polysemy: multiple related meanings

Often, the relationships are systematic

E.g., building vs. organization
school, university, hospital, church, supermarket

More examples:

- Author (Jane Austen wrote Emma) ↔ Works of Author (I really love Jane Austen)
- Animal (The chicken was domesticated in Asia) ↔ Meat (The chicken was overcooked)
- Tree (Plums have beautiful blossoms) ↔ Fruit (I ate a preserved plum yesterday)

Synonymy

Words that have the same meaning in some or all contexts.

- filbert / hazelnut
- couch / sofa
- big / large
- automobile / car
- vomit / throw up
- water / H_2O

Two lexemes are synonyms if they can be successfully substituted for each other in all situations.

Synonyms

Shades of meaning

- What is the cheapest first class *fare*?
- *What is the cheapest first class *price*?

Collocational constraints

- We frustate 'em and frustate 'em, and pretty soon they make a *big* mistake.
- *We frustate 'em and frustate 'em, and pretty soon they make a *large* mistake.

Antonyms

- Senses that are opposites with respect to one feature of their meaning
- Otherwise, they are similar!
  - dark / light
  - short / long
  - hot / cold
  - up / down
  - in / out

More formally: antonyms can

- define a binary opposition or at opposite ends of a scale (*long/short*, *fast/slow*)

Hyponymy and Hypernymy

Hyponymy

One sense is a hyponym of another if the first sense is more specific, denoting a subclass of the other

- *car* is a hyponym of *vehicle*
- *dog* is a hyponym of *animal*
- *mango* is a hyponym of *fruit*

Hypernymy

Conversely

- *vehicle* is a hypernym/superordinate of *car*
- *animal* is a hypernym of *dog*
- *fruit* is a hypernym of *mango*

Hyponymy more formally

Entailment

Sense A is a hyponym of sense B if being an A entails being a B.
Ex: dog, animal

Transitivity

A hypo B and B hypo C entails A hypo C

Meronyms and holonyms

Definition

Meronymy: an asymmetric, transitive relation between senses.
X is a **meronym** of Y if it denotes a part of Y.
The inverse relation is **holonymy**.

meronym	holonym
porch	house
wheel	car
leg	chair
nose	face

Pronoun resolution

- Definition: Refers to the one speaking….
- Examples:
- 1st Person: I, me, mine, we, us, our, ours
- 2nd Person: you, your, yours
- 3rd Person: he, him, his, she, her, hers, it, its, they, them, their, theirs
- Hint:Personal refers to a person
- **Sentence:Last spring, I visited my relatives.**

- **Reflexive Pronoun**: refers to the subject and functions as a complement or an object of a preposition.

- **Hint…Reflexive Refers, or Reflects back to the subject/Reflexive Reflects or Refers!

- **Intensive Pronoun**: emphasizes a noun or another pronoun.

- **Hint…Intensive Intensifies…don't need it! Write what's in black!

- Language consists of collocated, related groups of sentences. We refer to such a group of sentences as a **discourse**.

- There are two basic forms of discourse:
  - Monologue;
  - Dialogue;

- We will focus on techniques commonly applied to the interpretation of **monologues**.

- **Reference**: the process by which speakers use expressions to denote an entity.
- **Referring expression**: expression used to perform reference .
- **Referent**: the entity that is referred to.
- **Coreference**: referring expressions that are used to refer to the same entity.
- **Anaphora**: reference to a previously introduced entity.

Five common types of referring expression	
Type	**Example**
Indefinite noun phrase	I saw **a** Ford Escort today.
Definite noun phrase	I saw a Ford Escort today. **The** Escort was white.
Pronoun	I saw a Ford Escort today. **It** was white.
Demonstratives	I like **this** better than **that**.
One-anaphora	I saw 6 Ford Escort today. Now I want **one**.
Three types of referring expression that complicate the reference resolution	
Type	**Example**
Inferrables	I almost bought a Ford Escort, but a **door** had a dent.
Discontinuous Sets	John and Mary love their Escorts. **They** often drive them.
Generics	I saw 6 Ford Escorts today. **They** are the coolest cars.

- How to develop successful algorithms for reference resolution? There are two necessary steps.

- First is to filter the set of possible referents by certain hard-and-fast constraints.

- Second is to set the preference for possible referents.

Contraints

- **Number Agreement**:
  - To distinguish between singular and plural references.
    - *John has a new car. They are red.
- **Gender Agreement**:
  - To distinguish male, female, and non-personal genders.
    - John has a new car. It is attractive. [It = the new car]
- **Person and Case Agreement**:
  - To distinguish between three forms of person;
    - *You and I have Escorts. They love them.
  - To distinguish between subject position, object position, and genitive position.

- **Syntactic Constraints**:
  - Syntactic relationships between a referring expression and a possible antecedent noun phrase
    - John bought himself a new car. [himself=John]
    - John bought him a new car. [him≠John]

- **Selectional Restrictions**:
  - A verb places restrictions on its arguments.
    - John parked his Acura in the garage. He had driven it around for hours. [it=Acura, it≠garage];
    - I picked up the book and sat in a chair. It broke.

- **Verb Semantics**:
  - Certain verbs appear to place a semantically-oriented emphasis on one of their argument positions.

    - John telephoned Bill. He had lost the book in the mall. [He = John]
    - John criticized Bill. He had lost the book in the mall. [He = Bill]

    - David praised Hans because he … [he = Hans]
    - David apologized to Hans because he… [he = David]

Introduce and compare 3 algorithms for anaphora resolution:

- Hobbs 1978
- Lappin and Leass 1994
- Centering Theory

- This simple algorithm has become a baseline: more complex algorithms should do better than this.

- Hobbs distance: i^{th} candidate NP considered by the algorithm is at a Hobbs distance of *i.*

Multiple Parse trees

Because it assumes parse trees, such an algorithm is inevitably dependent on one's theory of grammar.

1. Mr. Smith saw a driver in his truck.
2. Mr. Smith saw a driver of his truck.

"his" may refer to the driver in 1, but not 2.

- different parse trees explain the difference:
  - in 1, if the PP is attached to the VP, "his" can refer back to the driver;
  - in 2, the PP is obligatorily attached inside the NP, so "his" cannot refer back to the driver.

Hobbs's "Naïve" Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

S5,S6,S7-Types of Disambiguation

Meaning representation

- Semantic analysis represents the meaning of any sentence. These are done by different processes and methods.
- Let us discuss some building blocks of the semantic system:
- **Entities:** Any sentence is made of different entities that are related to each other. It represents any individual category such as name, place, position, etc. We will discuss in detail about entities and their correlation later in this blog.
- **Concepts:** It represents the general category of individual, such as person, city etc.
- **Relations:** It represents the relation between different entities and concepts in a sentence.
- **Predicates:** It represents the verb structure of any sentence.

Different approaches to Meaning Representations

- First-order predicate logic (FOPL)
- Frames
- Semantic Nets
- Case Grammar
- Rule-based architecture
- Conceptual graphs
- Conceptual dependency (CD)

Semantic Analysis

- Semantic analysis is the process of finding the meaning from text.
- This analysis gives the power to computers to understand and interpret sentences, paragraphs, or whole documents, by analyzing their grammatical structure, and identifying the relationships between individual words of the sentence in a particular context.
- Therefore, the goal of semantic analysis is to draw exact meaning or dictionary meaning from the text.
- The work of a semantic analyzer is to check the text for meaningfulness.

Process of semantic analysis

- Word sense disambiguation
- Lexical Disambiguation
- Structural Disambiguation

Word sense disambiguation

- Word sense disambiguation is the automated process of identifying in which sense is a word used according to its context.
- As natural language consists of words with several meanings (polysemic), the objective here is to recognize the correct meaning based on its use.
- For example, ‘Raspberry Pi’ can refer to a fruit, a single-board computer, or even a company (UK-based foundation). Hence, it is critical to identify which meaning suits the word depending on its usage.

Word sense disambiguation

The word *"orange,"* for example, can refer to a color, a fruit, or even a city in Florida!

Introductions

- **Word sense disambiguation (WSD**) in Natural Language Processing (NLP) is the problem of identifying which "sense" (meaning) of a word is activated by the use of the word in a particular context or scenario.
- In people, this appears to be a largely unconscious process.
- The challenge of correctly identifying words in NLP systems is common, and determining the specific usage of a word in a sentence has many applications.
- The application of Word Sense Disambiguation involves the area of Information Retrieval, Question Answering systems, Chat-bots, etc.

Introductions

- Word Sense Disambiguation (WSD) is a subtask of Natural Language Processing that deals with the problem of identifying the correct sense of a word in context.
- Many words in natural language have multiple meanings, and WSD aims to disambiguate the correct sense of a word in a particular context.
- For example, the word “bank” can have different meanings in the sentences “I deposited money in the bank” and “The boat went down the river bank”.

Approaches to WSD

- **Supervised learning:** This involves training a machine learning model on a dataset of annotated examples, where each example contains a target word and its sense in a particular context.
- The model then learns to predict the correct sense of the target word in new contexts.
- **Unsupervised learning:** This involves clustering words that appear in similar contexts together, and then assigning senses to the resulting clusters.
- This approach does not require annotated data, but it is less accurate than supervised learning.
- **Knowledge-based:** This involves using a knowledge base, such as a dictionary or ontology, to map words to their different senses.
- This approach relies on the availability and accuracy of the knowledge base.
- **Hybrid:** This involves combining multiple approaches, such as supervised and knowledge-based methods, to improve accuracy

Difficulties in Word Sense Disambiguation

- **Different Text-Corpus or Dictionary:** One issue with word sense disambiguation is determining what the senses are because different dictionaries and thesauruses divide words into distinct senses.
- Some academics have proposed employing a specific lexicon and its set of senses to address this problem.
- In general, however, research findings based on broad sense distinctions have outperformed those based on limited ones.
- The majority of researchers are still working on fine-grained WSD.
- **PoS Tagging:** Part-of-speech tagging and sense tagging have been shown to be very tightly coupled in any real test, with each potentially constraining the other.
- Both disambiguating and tagging with words are involved in WSM part-of-speech tagging.
- However, algorithms designed for one do not always work well for the other, owing to the fact that a word's part of speech is mostly decided by the one to three words immediately adjacent to it, whereas a word's sense can be determined by words further away.

Sense Inventories for Word Sense Disambiguation

- **Princeton WordNet:** is a vast lexicographic database of English and other languages that is manually curated. For WSD, this is the de facto standard inventory. Its well-organized Synsets, or clusters of contextual synonyms, are nodes in a network.
- **BabelNet:** is a multilingual dictionary that covers both lexicographic and encyclopedic terminology. It was created by semi-automatically mapping numerous resources, including WordNet, multilingual versions of WordNet, and Wikipedia.
- **Wiktionary:** a collaborative project aimed at creating a dictionary for each language separately, is another inventory that has recently gained popularity.

Word Sense Disambiguation

Word Sense Disambiguation (WSD)

Sense ambiguity

- Many words have several meanings or senses
- The meaning of **bass** depends on the context
- Are we talking about music, or fish?
  - An electric guitar and **bass** player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.
  - And it all started when fishermen decided the striped **bass** in Lake Mead were too skinny.

Disambiguation

- The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word.
- This is done by looking at the context of the word's use.

Algorithms

- Knowledge Based Approaches
  - Overlap Based Approaches
- Machine Learning Based Approaches
  - Supervised Approaches
  - Semi-supervised Algorithms
  - Unsupervised Algorithms
- Hybrid Approaches

Knowledge Based Approaches

Overlap Based Approaches

- Require a **Machine Readable Dictionary** (MRD).
- Find the overlap between the features of different senses of an ambiguous word (**sense bag**) and the features of the words in its context (**context bag**).
- The features could be sense definitions, example sentences, hypernyms etc.
- The features could also be given weights.
- The sense which has the maximum overlap is selected as the contextually appropriate sense.

Walker's Algorithms

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*
  - E.g. The money in this bank fetches an interest of 8% per annum
  - Target word: *bank*
  - Clue words from the context: *money, interest, annum, fetch*

	[illegible]	[illegible]
Money	+1	0
Interest	+1	0
Fetch	0	0
Annum	+1	0
Total	3	0

Context words add 1 to the sense when the topic of the word matches that of the sense

The **church bell**s no longer **rung** on **Sunday**s

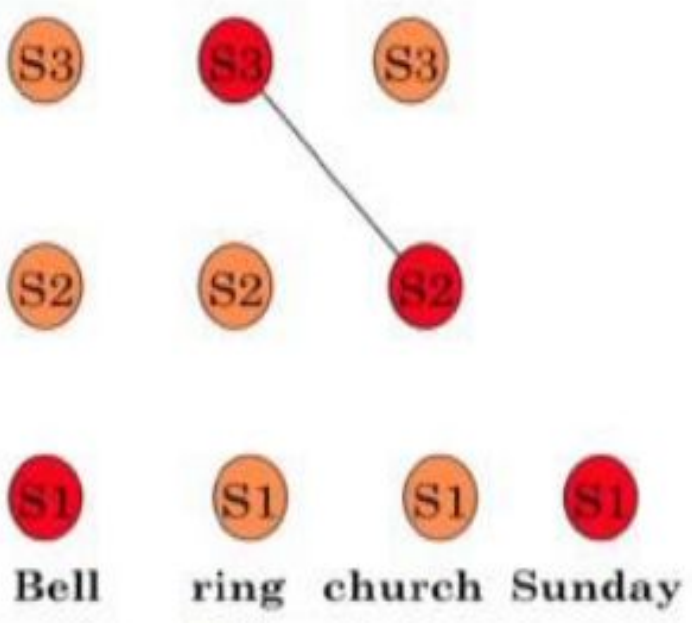
church
1. one of the groups of Christians who have their own beliefs and forms of worship
2. a place for public (especially Christian) worship
3. a service conducted in a church

bell
1. a hollow device made of metal that makes a ringing sound when struck
2. a push button at an outer door that gives a ringing or buzzing signal when pushed
3. the sound of a bell

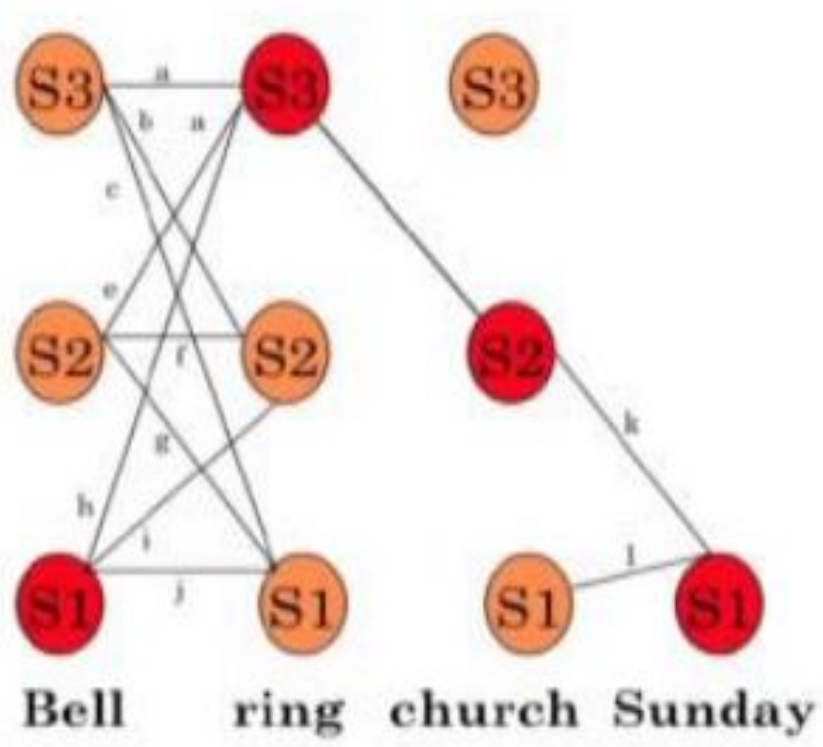
ring
1. make a ringing sound
2. ring or echo with sound
3. make (bells) ring, often for the purposes of musical edification

Sunday
1. first day of the week; observed as a day of rest and worship by most Christians

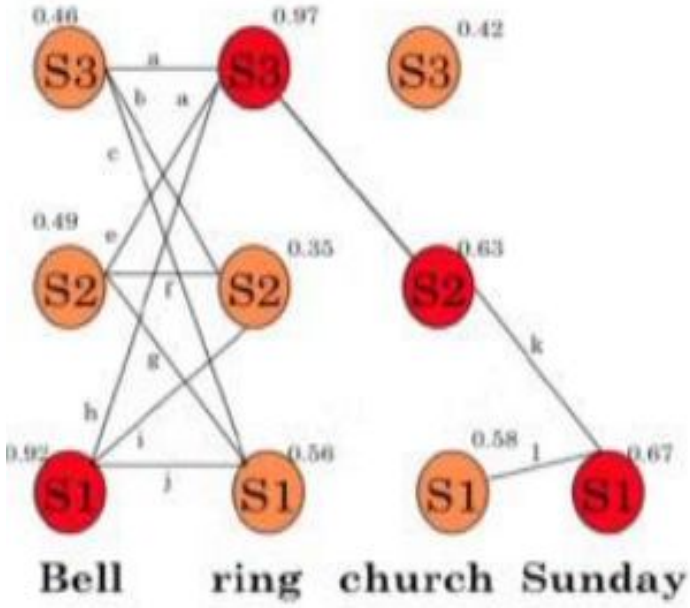
Step 1: Add a vertex for each possible sense of each word in the text.



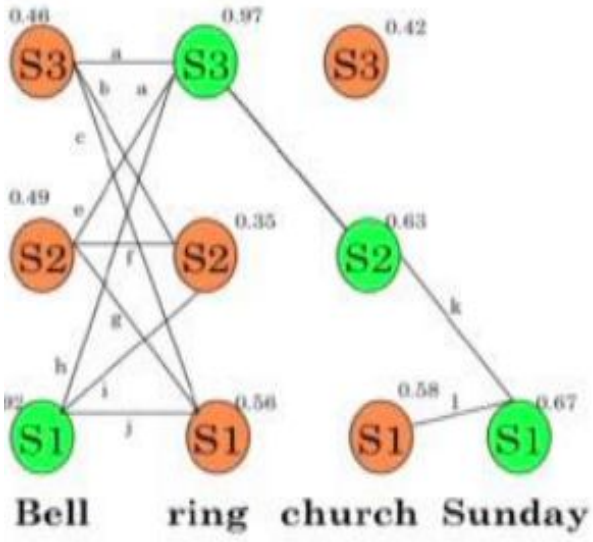
Step 2: Add weighted edges using definition based semantic similarity (Lesk' s method).



Step 3: Apply graph based ranking algorithm to find score of each vertex (i.e. for each word sense).



Step 4: Select the vertex (sense) which has the highest score.



Naïve Bayes for WSD

- A Naïve Bayes classifier chooses the most likely sense for a word given the features of the context:

$$\hat{s} = \arg\max_{s \in S} P(s|f)$$

- Using Bayes' law, this can be expressed as:

$$\hat{s} = \arg\max_{s \in S} \frac{P(s)P(f|s)}{P(f)}$$

$$= \arg\max_{s \in S} P(s)P(f|s)$$

- The 'Naïve' assumption: all the features are conditionally independent, given the sense':

$$\hat{s} = \arg\max_{s \in S} P(s) \prod_{j=1}^{n} P(f_j|s)$$

Training for Naïve Bayes

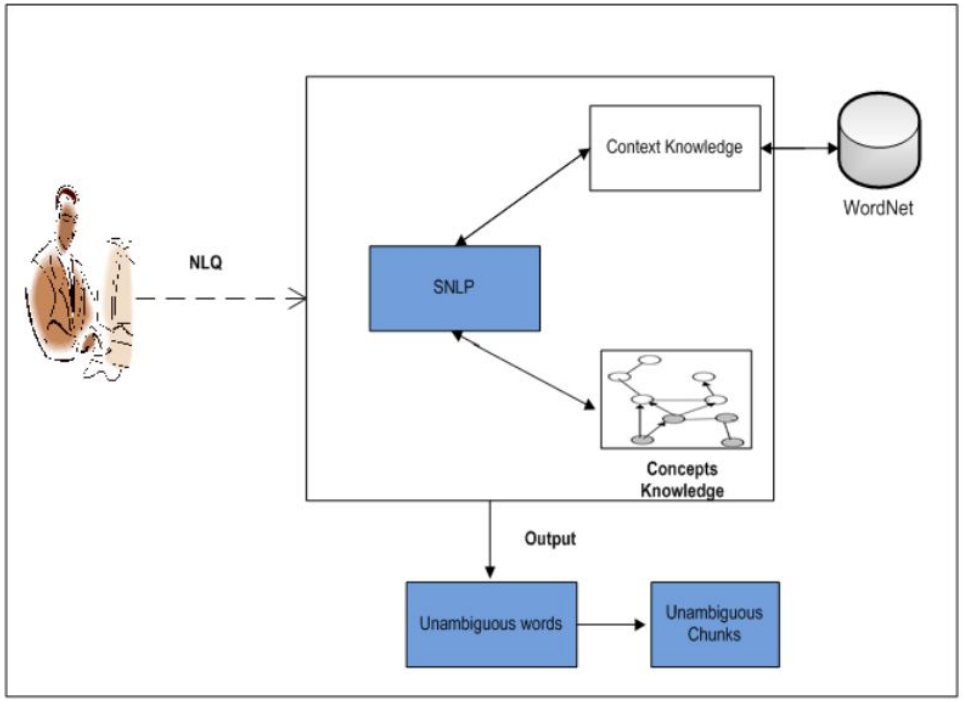
- 'f' is a feature vector consisting of:
  - POS of w
  - Semantic and Syntactic features of w
  - Collocation vector (set of words around it) → next word (+1), +2, -1, -2 and their POS's
  - Co-occurrence vector
- Set parameters of Naïve Bayes using maximum likelihood estimation (MLE) from training data

$$P(s_i) = \frac{count(s_i, w_j)}{count(w_j)}$$

Lexical Disambiguation

- The proposed approach solves lexical ambiguity in QA by considering two pieces of knowledge: context knowledge, and concepts knowledge.
- The combination of these knowledge is used to decide the most possible meaning of the word.

Lexical Disambiguation Framework



Context Knowledge

- Context knowledge contains a set of lexical with their semantic relations.
- The set of lexical with its semantics are extracted from the WordNet database manually.
- All semantics in this work are extracted from the WordNet and combined with a context label.
- For instance, the word bank may have 5 possible meanings as shown in Table

Table 1: Context knowledge of the word *bank*

Sense	***Gloss***	***Context***
#1	Sloping land	Money, Deposit
#2	Financial institute	River. Lake
#3	container	Money
#4	the funds held by gambling house	Money & Play
#5	a flight maneuver	Transport
#6	a supply or stock held in reserve	Money

Concepts Knowledge

- Concepts knowledge is ontology consists of a set of concepts which are within the domain, and the relationships between the concepts.
- The ontology also specifies how knowledge is related to linguistic structures such as grammars and lexicons

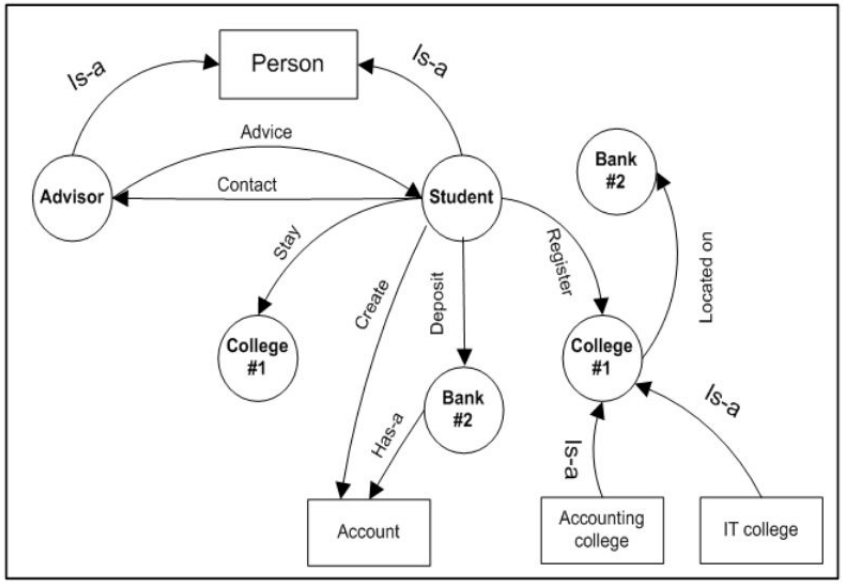
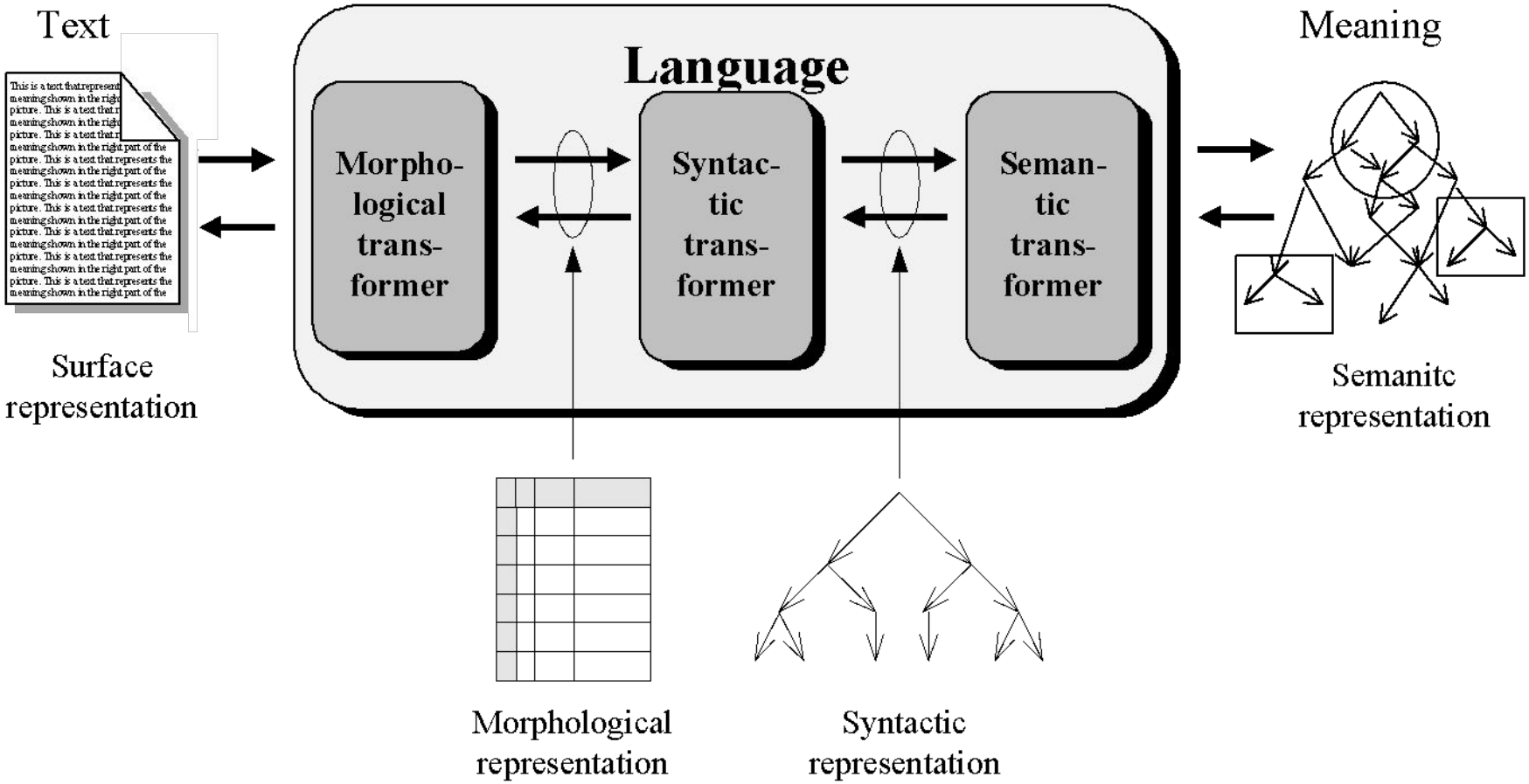


Fig. 3 An example of concepts-domain ontology.

Structural Disambiguation



Example of text

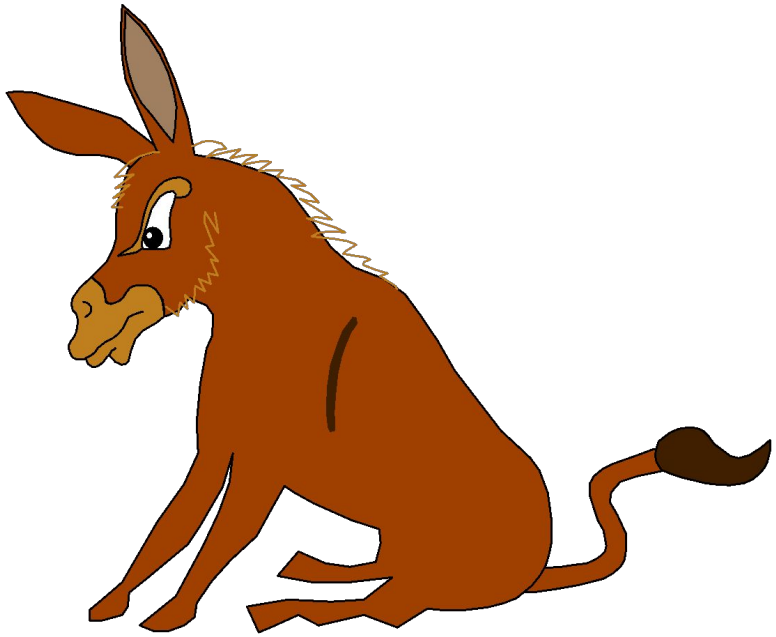
"Science is important for our country.

The Government pays it much attention."

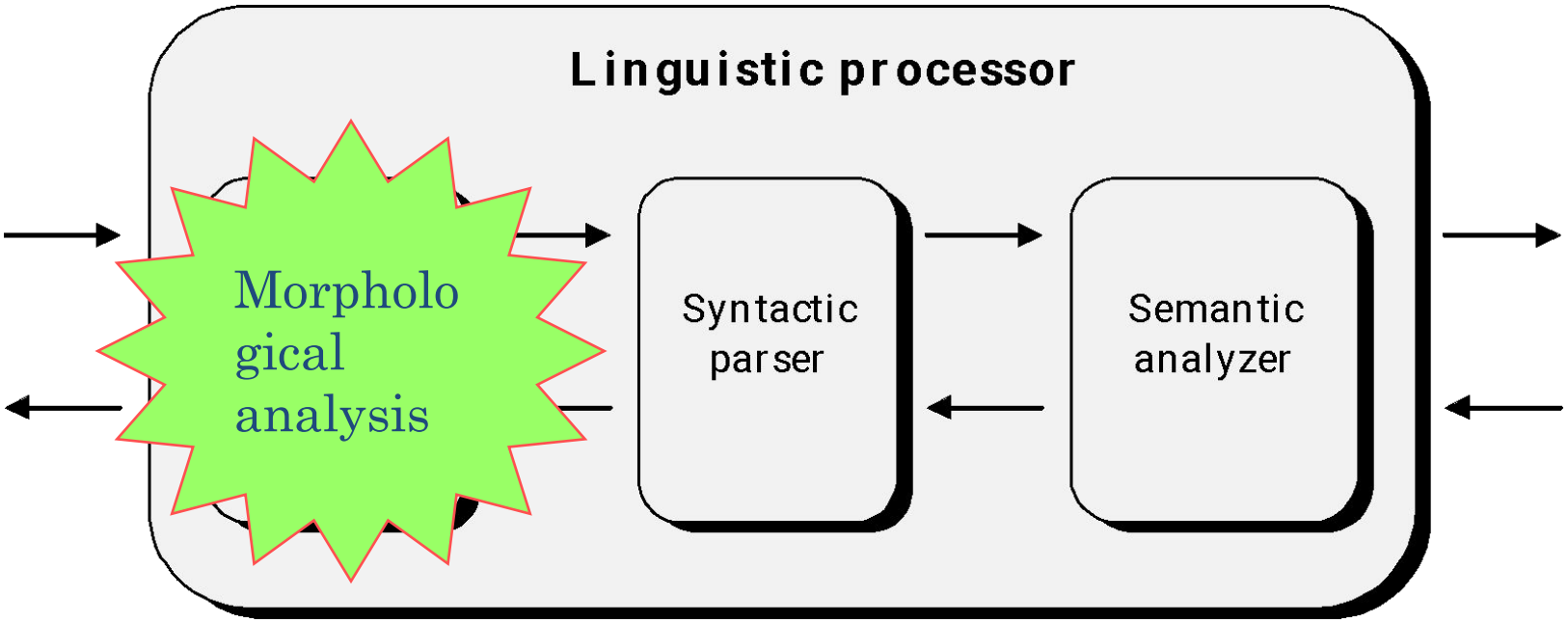
Textual representation

Text is a sequence of letter.

```
S c i e n c e   i s
i m p o r t a n t
f o r   o u r   c o
u n t r y . T h e
G o v e r n m e n t
p a y s   i t   m u
c h   a t t e n t i
o n .
```

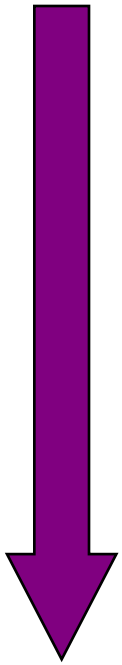


Morphological analysis



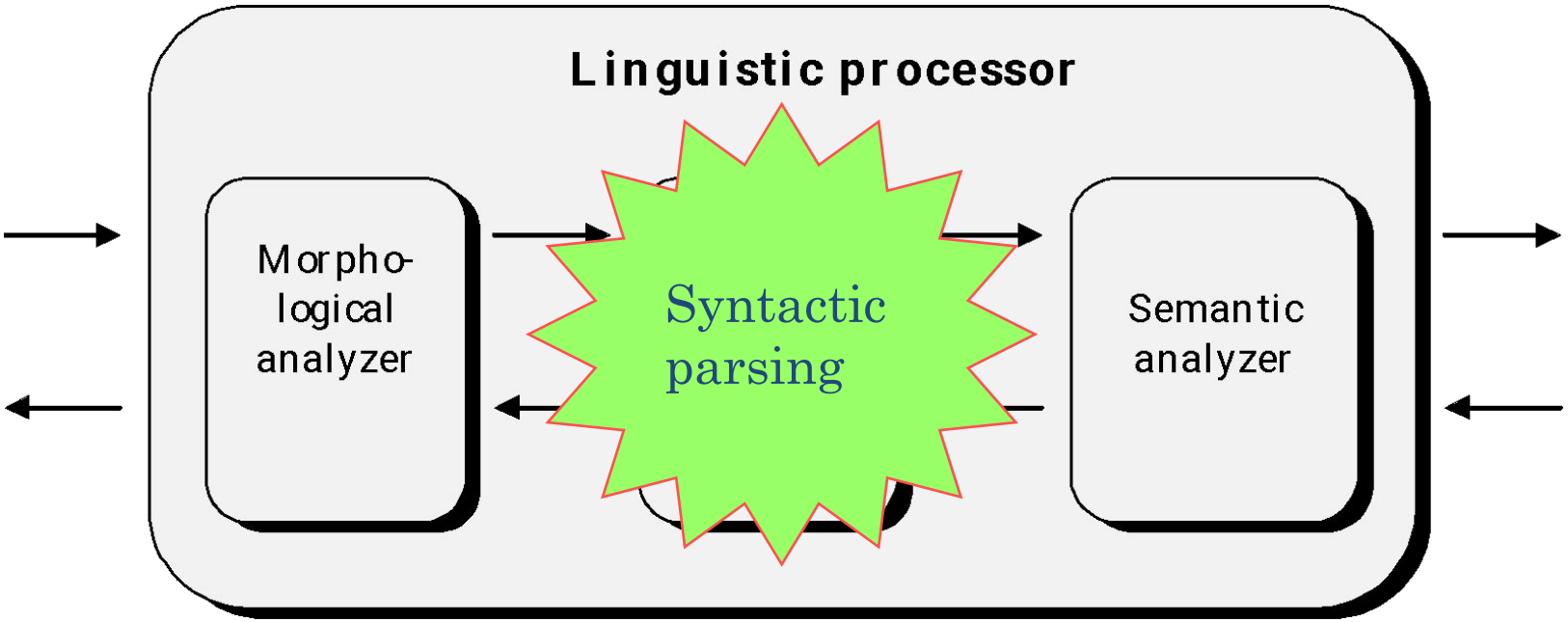
Morphological Representation

A sequence of words.



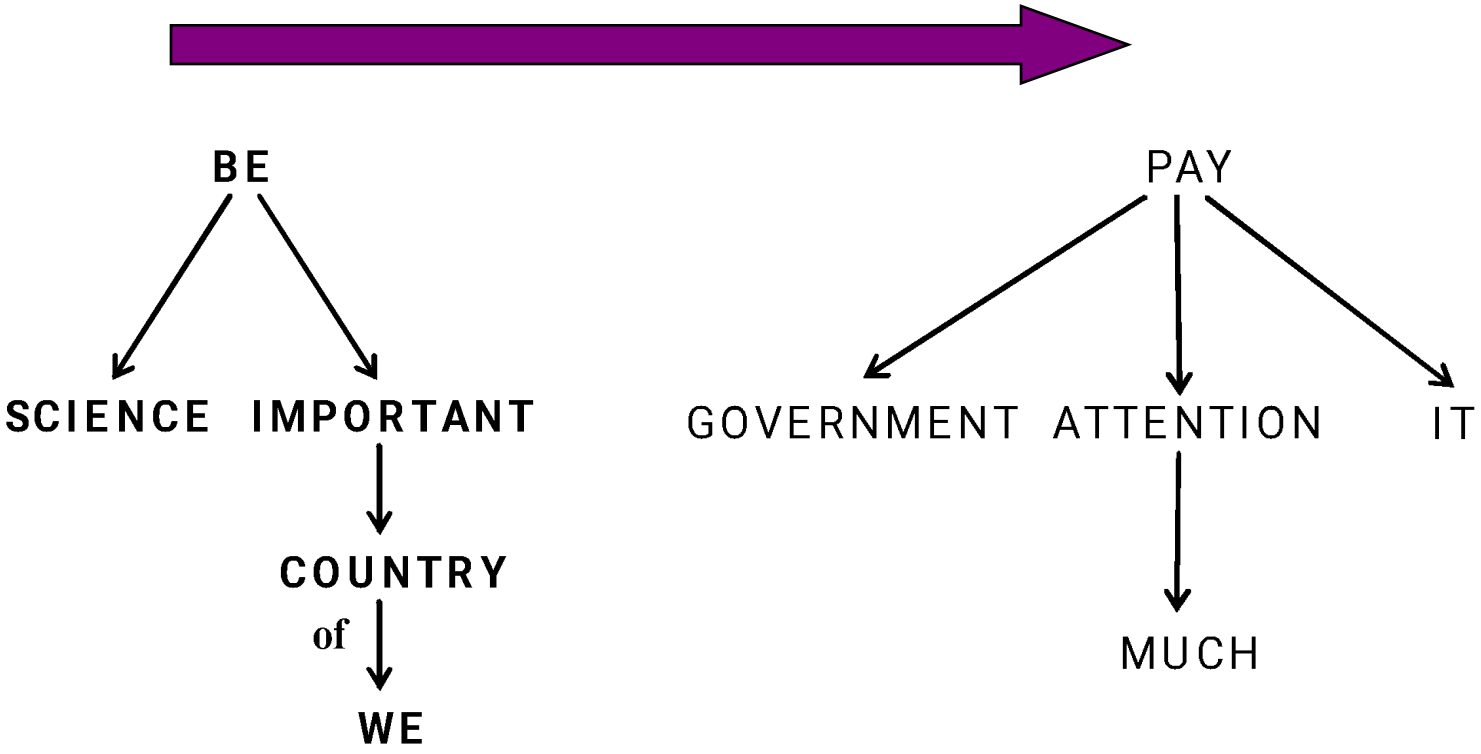
The	THE	article	definite, plural/singular
science	SCIENCE	noun	singular
is	**BE**	**verb**	**present, 3rd person, sing.**
important	IMPORTANT	adjective	
for	FOR	preposition	
our	WE	pronoun	possessive
country	COUNTRY	noun	singular

Syntactic parsing



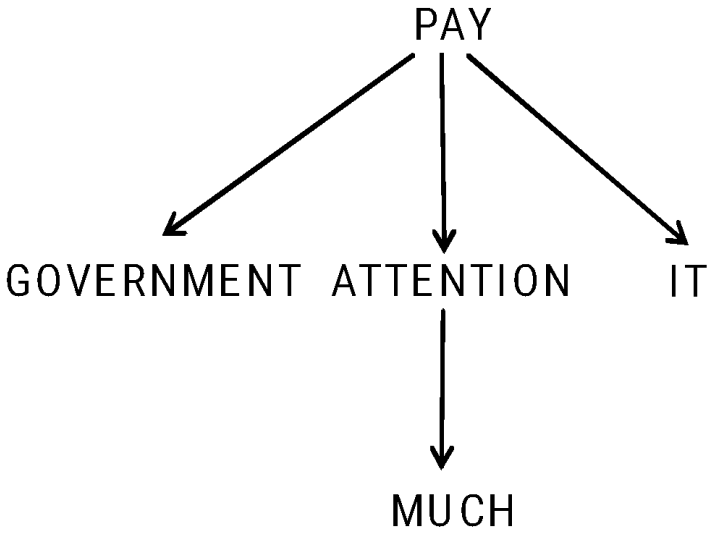
Syntactic Representation

A sequence of syntactic trees.

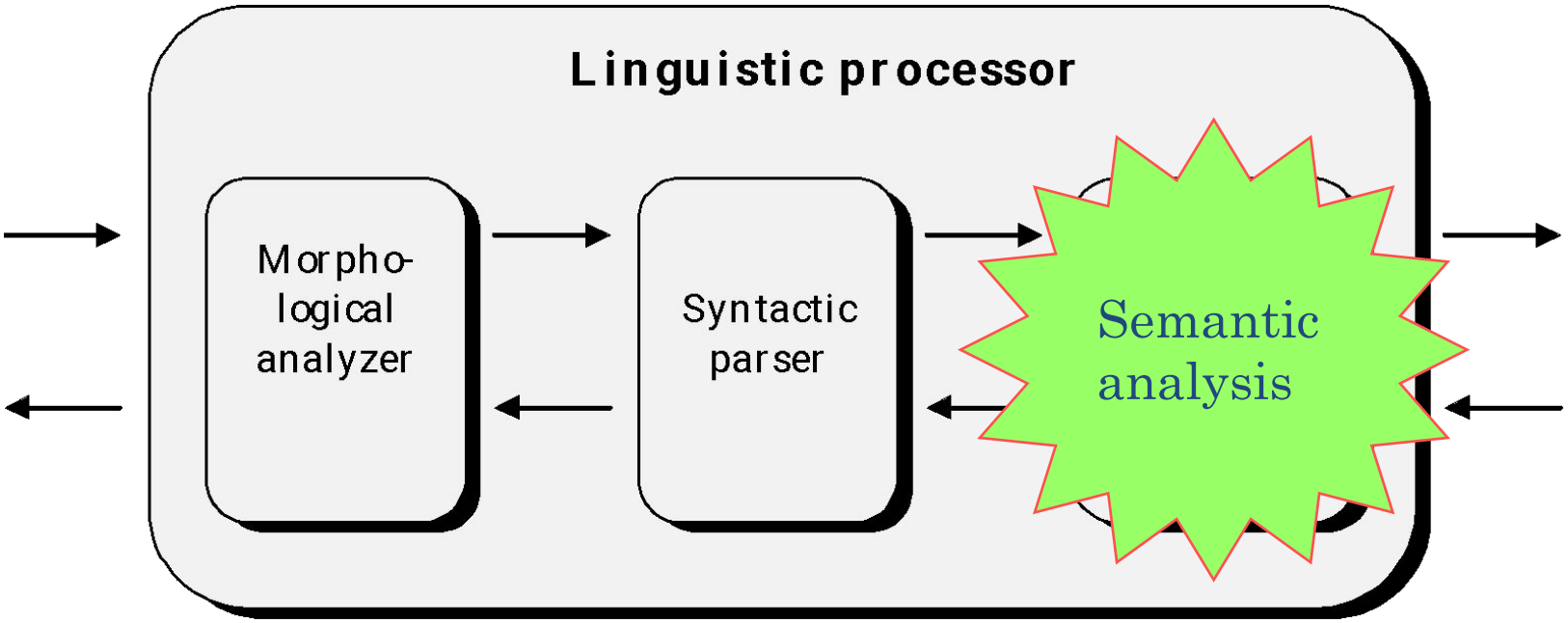


Syntactic representation

- What happened?
- With whom happened?
- ... their details



Semantic analysis



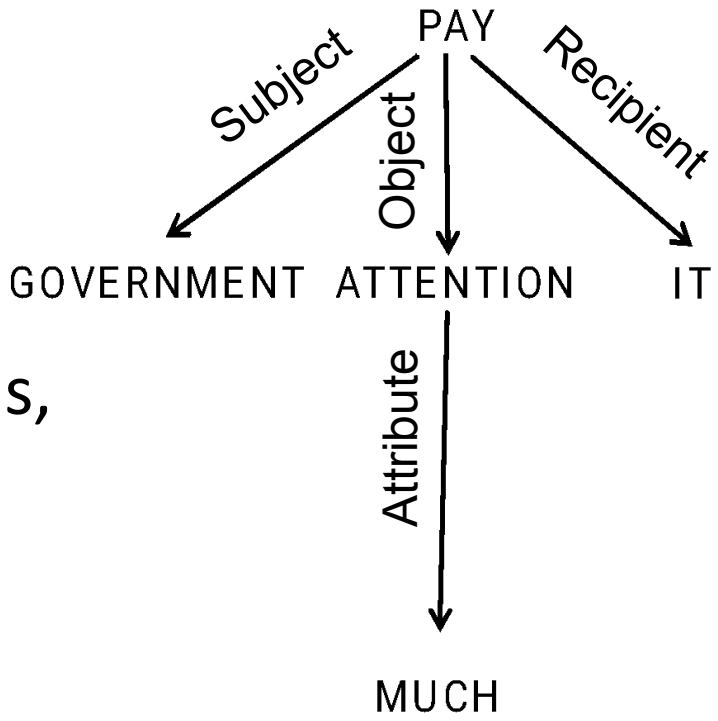
Next lecture...

Syntax

- The structure describing the relationships between words in a sentence
- Describes the relationships implied by grammatical characteristics
  - not by meaning
- Often allows for simple paraphrasing
  - *John reads the book*
  - *The book is read by John*

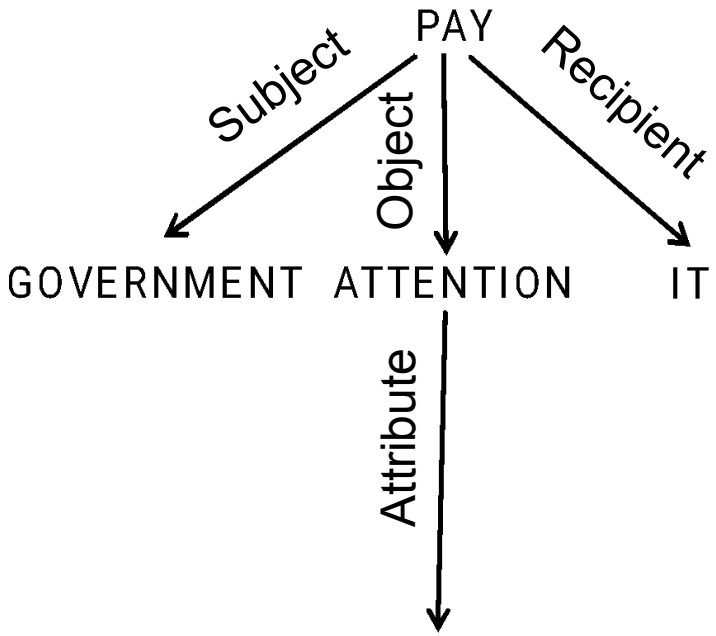
Early approach: Dependency syntax

- Tree
- Nodes: **words**
- Arcs: *modified by*
  - Modifies means adds details, clarifies, chooses of many... makes more specific
- Arcs are **typed**
  - Types are: subject, object, attribute, ...



… Dependency syntax

- General situation: *pay*
- More specifically: the one where:
  - who pays is *government*
  - what is paid is *attention*
  - to whom it is paid is *it*
- More specifically: *attention* that is *much*



Advantages/Disadvantages of Dependency Syntax

Advantages

- Solid linguistic base
- Rather direct translation into semantics
- Easily applicable to languages with free word order
  - Korean? Russian, Latin
  - This is why solid linguistic base: good for classical languages!

Disadvantages

- No nice mathematical base
- No simple algorithms

Most popular approach: Constituency (Phrase Structure grammars)

- Tree
- Nodes: nested segments of the phrase
  - Cannot intersect, only nested
  - Usually are labeled with part-of-speech names
- Arcs: nesting
- In classical approach, arcs are not labeled

[[Our Government] [pays [much attention] [to it]]]

Constituency

[[Our Government] [pays [much attention] [to it]]]

Our Government

pays

much attention

to it

Constituency

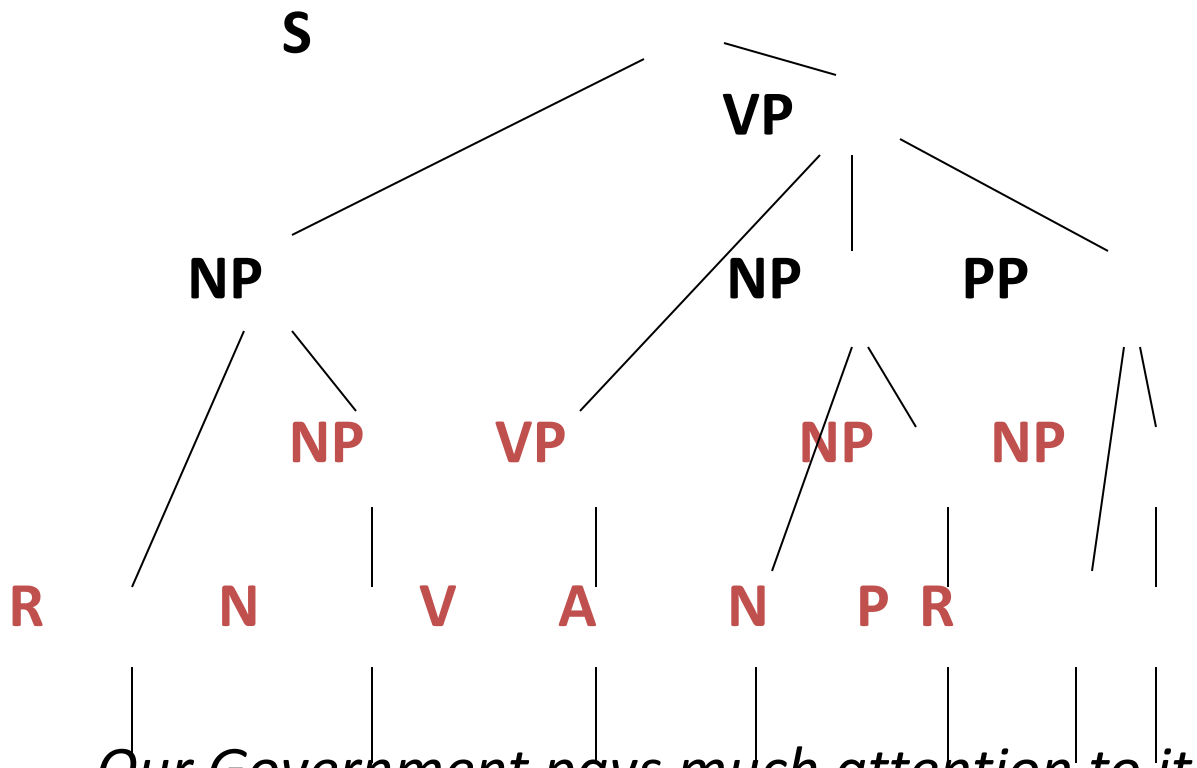
$$\Big[\,\big[\,\text{Our}_{R}\ \text{Government}_{N}\,\big]_{NP}$$

$$\Big[\,\text{pays}_{V}\,\big[\,\text{much}_{A}\ \text{attention}_{N}\big]_{NP}\,\big[\text{to}_{P}\ \text{it}_{R}\,\big]_{PP}\,\big]_{VP}\,\Big]_{S}$$

R: pronoun — NP: noun phrase
N: noun — VP: verb phrase
V: verb — PP: prepositional phrase
A: adjective — S: sentence

Constituency: graphical representation

$$\Big[\big[\text{Our Government}\big]_{NP}\big[\text{pays}\big[\text{much attention}\big]_{NP}\big[\text{to it}\big]_{PP}\big]_{VP}\Big]_{S}$$



Phrase structure grammar

- Enumerates possible configurations at nodes
- Usually recursive

S ← NP VP

NP ← A NP

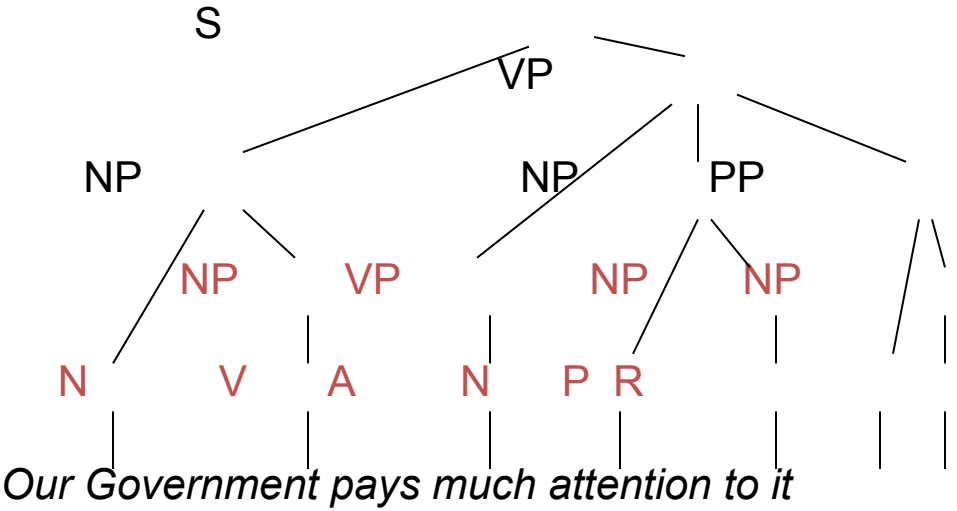
NP ← R NP

NP ← P NP

NP ← N

VP← VP NP PP

VP← V



Context-independency hypothesis

- A configuration is possible or not, regardless of where it is used
  - Wherever you find VP NP PP, it can be VP
  - Wherever you find NP VP, it can be S
  - If you can put together S that covers all the sentence, it is a grammatically correct description
- With this, given a suitable grammar, you can
  - List all sentences of a language
  - List only correct sentences of that language
- List all and only correct structures
- Correctness means a native speaker's intuition

Generative idea

- Find a grammar to list all and only correct sentences (with their structures) of a language
- This is a complete description of that language!

- How can be useful in analysis?
  - Reverse the grammar

Parsing

- Given a grammar and a sentence
- Find all possible structures
- That describe this sentence with this grammar
- Many methods. Not discussed today.
  A lot of research. Very fast algorithms
- Complexity: cubic in the number of words in the sentence (there are better methods, up to 2.8)
- Problem: combinatorics of variants

Advantages and disadvantages of constituency approach

Advantages

- Nice mathematics, very well understood
- Efficient analysis algorithms, very well-elaborated
- Good for languages with fixed word order
  - English. Chinese?

Disadvantages

- Difficult translation into semantics
- Bad when it comes to freer word order
  - Even in English! Worse in other languages

Head-driven approaches

- Combine some advantages of dependency-based and constituency-based approaches
- Syntax is still fixed-order. But word dependency information is added
  - Easier translation into semantics
  - More linguistically-based
- How?
  - In each constituent, the main word (*head*) is marked
  - It modifies the head of the larger constituent

[[Our **Government**] [**pays** [much **attention**] [to **it**]]]

Semantic Analysis

- Semantic analysis is the process of finding the meaning from text. This analysis gives the power to computers to understand and interpret sentences, paragraphs, or whole documents, by analyzing their grammatical structure, and identifying the relationships between individual words of the sentence in a particular context.

- Therefore, the goal of semantic analysis is to draw exact meaning or dictionary meaning from the text. The work of a semantic analyzer is to check the text for meaningfulness.

How is Semantic Analysis different from Lexical Analysis?

- Lexical analysis is based on smaller tokens but on the contrary, the semantic analysis focuses on larger chunks.
- Since semantic analysis focuses on larger chunks, therefore we can divide the semantic analysis into the following two parts:
- **Studying the meaning of the Individual Word**
- It is the first component of semantic analysis in which we study the meaning of individual words. This component is known as lexical semantics.
- **Studying the combination of Individual Words**
- In this component, we combined the individual words to provide meaning in sentences.

- For Example, consider the following sentence:

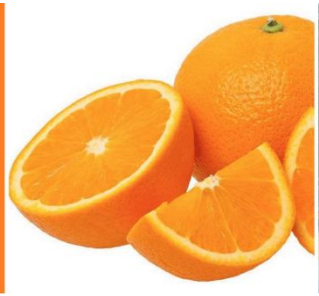
- Sentence: Ram is great
- In the above sentence, the speaker is talking either about Lord Ram or about a person whose name is Ram. That is why the task to get the proper meaning of the sentence is important.

Sentiment Analysis with Machine Learning

- implement a semantic-based approach for machine learning, there are various sub-tasks involved including
- Word sense disambiguation
- Relationship extraction.

- Word Sense Disambiguation
- As we have discussed that Natural language is ambiguous and polysemic; sometimes, the same word can have different meanings depending on its use in the sentence.
- Therefore, in semantic analysis with machine learning, computers use Word Sense Disambiguation to determine which meaning is correct in the given context.

- For Example,
- Consider the word: Orange
- The above word can refer to a color, a fruit, or even a city in Florida!

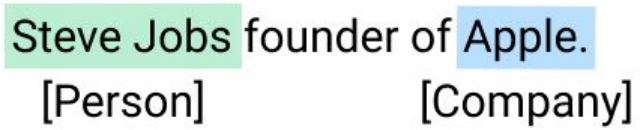




- Relationship Extraction
- In this task, we try to detect the semantic relationships present in a text. Usually, relationships involve two or more entities such as names of people, places, company names, etc.

- These entities are joined through a semantic category, like “works at,” “lives in,” “is the CEO of,” “headquartered at.”

- For Example, Consider the following phrase

- Phrase: Steve Jobs is the founder of Apple, which is headquartered in California



- **Difference between Polysemy and Homonymy**
- Both polysemy and homonymy words have the same syntax or spelling but the main difference between them is that in polysemy, the meanings of the words are related but in homonymy, the meanings of the words are not related.
- **For Example,** if we talk about the same word “Bank” as discussed above, we can write the meaning as
- ‘a financial institution’ or
- ‘a river bank’.
- In that case, it becomes an example of a homonym, as the meanings are unrelated to each other.

- Synonymy
- It represents the relation between two lexical items of different forms but expressing the same or a close meaning.

- For Example,

- ‘author/writer’, ‘fate/destiny'

- Antonymy
- It is the relation between two lexical items having symmetry between their semantic components relative to an axis. The scope of antonymy is as follows −
- Application of property or not:
- For Examples,
- ‘life/death’, ‘certitude/incertitude’

- Application of scalable property:
- For Examples,
- ‘rich/poor’, ‘hot/cold’
- Application of a usage:

- For Examples,
- ‘father/son’, ‘moon/sun’

- Meronomy
- It is defined as the logical arrangement of text and words that denotes a constituent part of or member of something.

- For Example,

- A segment of an orange

- **Meaning Representation**
- The semantic analysis creates a representation of the meaning of a sentence. But before deep dive into the concept and approaches related to meaning representation, firstly we have to understand the building blocks of the semantic system.

- Concepts
- It represents the general category of the individuals such as a person, city, etc.

- Relations
- It represents the relationship between entities and concepts.

- For Example,

- Sentence: Ram is a person

- Predicates
- It represents the verb structures.

- For Example,

- Semantic roles and Case Grammar
- Now, we have a brief idea of meaning representation that shows how to put together the building blocks of semantic systems. In other words, it shows how to put together entities, concepts, relations, and predicates to describe a situation. It also enables reasoning about the semantic world.

- **Approaches to Meaning Representations**
- The semantic analysis uses the following approaches for the representation of meaning −
- First-order predicate logic (FOPL)
- Semantic Nets
- Frames
- Conceptual dependency (CD)
- Rule-based architecture
- Case Grammar
- Conceptual Graphs

- **Need of Meaning Representations**
- The reasons behind the need for the meaning representation are as follows:
- **Linking of linguistic elements to non-linguistic elements**
- With the help of meaning representation, we can link linguistic elements to non-linguistic elements.
- **Representing variety at the lexical level**
- With the help of meaning representation, we can represent unambiguously, canonical forms at the lexical level.
- **Can be used for reasoning**
- The meaning representation can be used to reason for verifying what is correct in the world as well as to extract the knowledge with the help of semantic representation.

- **Lexical Semantics**
- It is the first part of semantic analysis, in which we study the meaning of individual words. It involves words, sub-words, affixes (sub-units), compound words, and phrases also. All the words, sub-words, etc. are collectively known as lexical items.
- In simple words, we can say that lexical semantics represents the relationship between lexical items, the meaning of sentences, and the syntax of the sentence.
- The steps which we have to follow while doing lexical semantics are as follows:
- Classification of lexical items.
- Decomposition of lexical items.
- Differences, as well as similarities between various lexical-semantic structures, are also analyzed.
-

- **Techniques of Semantic Analysis**
- We can any of the below two semantic analysis techniques depending on the type of information you would like to obtain from the given data.
- **text classification model**(which assigns predefined categories to text)
- **text extractor** (which pulls out particular information from the text).

- **Semantic Classification Models**
- **Topic Classification**
- Based on the content, this model sorts the text into predefined categories. In a company, Customer service teams may want to classify support tickets as they drop into their help desk, and based on the category it will distribute the work.
- With the help of semantic analysis, machine learning tools can recognize a ticket either as a “Payment issue” or a“Shipping problem”.
- **Sentiment Analysis**
- In Sentiment analysis, our aim is to detect the emotions as positive, negative, or neutral in a text to denote urgency.
- **For Example**, Tagging Twitter mentions by sentiment to get a sense of how customers feel about your product and can identify unhappy customers in real-time.

- **Intent Classification**
- We can classify the text based on the new user requirement.
- You can these types of models to tag sales emails as either “Interested” or “Not Interested” to proactively reach out to those users who may want to try your product.

- **Semantic Extraction Models**
- **Keyword Extraction**
- It is used to find relevant words and expressions from a text. This technique is used separately or can be used along with one of the above methods to gain more valuable insights.
- **For Example,** you could analyze the keywords in a bunch of tweets that have been categorized as “negative” and detect which words or topics are mentioned most often.
- **Entity Extraction**
- The idea of entity extraction is to identify named entities in text, such as names of people, companies, places, etc.
- This might be useful for a customer service team to automatically extract names of products, shipping numbers, emails, and any other relevant data from customer support tickets.

Pronoun/Anaphora resolution

- (AR) which most commonly appears as pronoun resolution is the problem of resolving references to earlier or later items in the discourse. These items are usually noun phrases representing objects in the real world called referents but can also be verb phrases, whole sentences or paragraphs. AR is classically recognized as a very difficult problem in NLP

- There are primarily three types of anaphora: - Pronominal: This is the most common type where a referent is referred by a pronoun. Example: "John found the love of his life" where 'his' refers to 'John'. - Definite noun phrase: The antecedent is referred by a phrase of the form " ". Continued example: "The relationship did not last long", where 'The relationship' refers to 'the love' in the preceding sentence. - Quantifier/Ordinal: The anphor is a quantifier such as 'one' or an ordinal such as 'first'. Continued Example: "He started a new one" where 'one' refers to 'The relationship' (effectively meaning 'a relationship').

- The traditional resolution techniques include:
- - Eliminative Constraints: An anaphor and a referent must agree in certain attributes to generate a match. These include gender (male/female/neutral), number (singular/plural), and semantic consistency (e.g. ‘a disk’ is ‘copied’ and ‘a computer’ ‘disconnected’, not vice versa).
- - Weighting Preferences: These factors are used to assign likelihood of match to the competing referents. They include proximity (of an antecedent phrase to the anaphor in the text), centering (which determines the center of attention object) and syntactic/semantic (role) parallelism. As examples of the latter, consider the form " however " where is likely to match with .
- There are important applications of anphora resolution in information extraction such as "comprehending" a discourse in order to summarize it or answer questions from it.

References

- https://blog.smart-tribune.com/en/semantic-analysis#:~:text=Syntactic%20analysis%20focuses%20on%20%E2%80%9Cform,not%20just%20a%20single%20word