Course Code	Course Name	Credits
DLO8012	**Natural Language Processing**	**4**

Course objectives:

1. To understand natural language processing and to learn how to apply basic algorithms in this field.
2. To get acquainted with the basic concepts and algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics.
3. To design and implement applications based on natural language processing
4. To implement various language Models.
5. To design systems that uses NLP techniques

Course outcomes: On successful completion of course learner should:

1. Have a broad understanding of the field of natural language processing.
2. Have a sense of the capabilities and limitations of current natural language technologies,
3. Be able to model linguistic phenomena with formal grammars.
4. Be able to Design, implement and test algorithms for NLP problems
5. Understand the mathematical and linguistic foundations underlying approaches to the various areas in NLP
6. Be able to apply NLP techniques to design real world NLP applications such as machine translation, text categorization, text summarization, information extraction...etc.

Prerequisite: Data structure & Algorithms, Theory of computer science, Probability Theory.

Module No.	Unit No.	Topics	Hrs.
1	**Introduction**	History of NLP, Generic NLP system, levels of NLP , Knowledge in language processing , Ambiguity in Natural language , stages in NLP, challenges of NLP ,Applications of NLP	4
2	**Word Level Analysis**	Morphology analysis –survey of English Morphology, Inflectional morphology & Derivational morphology, Lemmatization, Regular expression, finite automata, finite state transducers (FST) ,Morphological parsing with FST , Lexicon free FST Porter stemmer. N –Grams- N-gram language model, N-gram for spelling correction.	10
3	**Syntax analysis**	Part-Of-Speech tagging(POS)- Tag set for English (Penn Treebank) , Rule based POS tagging, Stochastic POS tagging, Issues –Multiple tags & words, Unknown words. Introduction to CFG, Sequence labeling: Hidden Markov Model (HMM), Maximum Entropy, and Conditional Random Field (CRF).	10
4	**Semantic Analysis**	Lexical Semantics, Attachment for fragment of English- sentences, noun phrases, Verb phrases, prepositional phrases, Relations among lexemes & their senses –Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Robust Word Sense Disambiguation (WSD) ,Dictionary based approach	10

5	**Pragmatics**	Discourse –reference resolution, reference phenomenon , syntactic & semantic constraints on co reference	8
6	**Applications (preferably for Indian regional languages)**	Machine translation, Information retrieval, Question answers system, categorization, summarization, sentiment analysis, Named Entity Recognition.	10

Text Books:

1. Daniel Jurafsky, James H. Martin "Speech and Language Processing" Second Edition, Prentice Hall, 2008.
2. Christopher D.Manning and Hinrich Schutze, " Foundations of Statistical Natural Language Processing ", MIT Press, 1999.

Reference Books:

1. Siddiqui and Tiwary U.S., Natural Language Processing and Information Retrieval, Oxford University Press (2008).
2. Daniel M Bikel and Imed Zitouni " Multilingual natural language processing applications" Pearson, 2013
3. Alexander Clark (Editor), Chris Fox (Editor), Shalom Lappin (Editor) " The Handbook of Computational Linguistics and Natural Language Processing " ISBN: 978-1-118-
4. Steven Bird, Ewan Klein, Natural Language Processing with Python, O'Reilly
5. Brian Neil Levine, An Introduction to R Programming
6. Niel J le Roux, Sugnet Lubbe, A step by step tutorial : An introduction into R application and programming

Ch. 1 Introduction

Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of research and application that determines the way computers can be used to understand and manage natural language text or speech to do useful things.

History of NLP

Here, are important events in the history of Natural Language Processing:

1950- NLP started when Alan Turing published an article called "Machine and Intelligence."

1950- Attempts to automate translation between Russian and English

1960- The work of Chomsky and others on formal language theory and generative syntax

1990- Probabilistic and data-driven models had become quite standard

2000- A Large amount of spoken and textual data become available

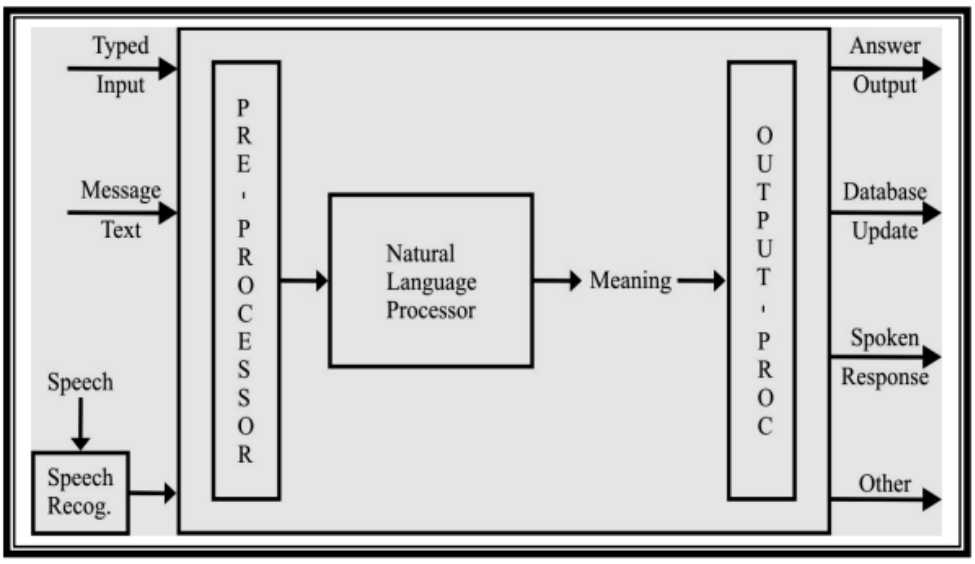
Background

Solving the language-related problems and others like them, is the main concern of the fields known as Natural Language Processing, Computational Linguistics, and Speech Recognition and Synthesis, which together we call Speech and Language Processing(SLP).

Applications of language processing:

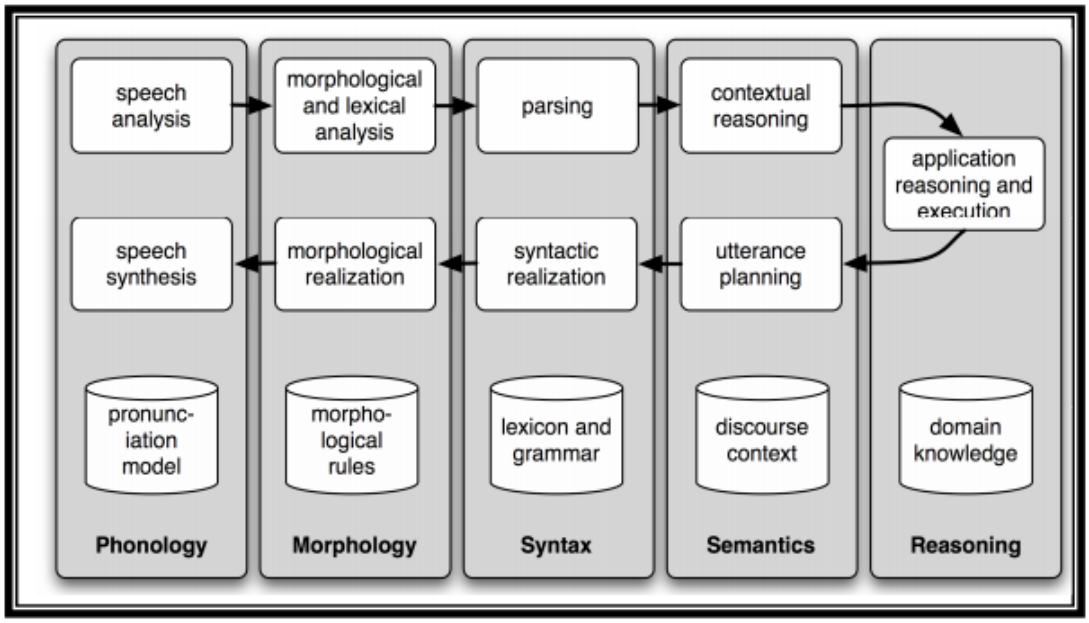
1. spelling correction,
2. grammar checking,
3. information retrieval, and
4. Machine translation.

Generic NLP



Levels of NLP

The NLP can broadly be divided into various levels as shown in figure



1. **Phonology**: It deals with interpretation of speech sound within and across words.
2. **Morphology**: It is a study of the way words are built up from smaller meaning-bearing units called morphemes. For example, the word 'fox' has single morpheme while the word 'cats' have two morphemes 'cat' and morpheme '–s' represents singular and plural concepts. Morphological lexicon is the list of stem and affixes together with basic information, whether the stem is a Noun stem or a Verb stem.
3. **Syntax:** It is a study of formal relationships between words. It is a study of: how words are clustered in classes in the form of Part-of Speech (POS), how they are grouped with their neighbors into phrases, and the way words depend on each other in a sentence.
4. **Semantics:** It is a study of the meaning of words that are associated with grammatical structure. It consists of two kinds of approaches: syntax-driven semantic analysis and semantic grammar. The detailed explanation of this level is discussed in chapter 4. In discourse context, the level of NLP works with text longer than a sentence. There are two types of discourse- anaphora resolution and discourse/text structure recognition. Anaphora resolution is replacing of words such as pronouns. Discourse structure recognition determines the function of sentences in the text which adds meaningful representation of the text.
5. **Reasoning:** To produce an answer to a question which is not explicitly stored in a database; Natural Language Interface to Database (NLIDB) carries out reasoning based on data stored in the database. For example, consider the database that holds the academic information about student, and user posed a query such as: 'Which student is likely to fail in Maths subject?'. To answer the query, NLIDB needs a domain expert to narrow down the reasoning process.

Knowledge in language processing

By speech and language processing, we have in mind those computational techniques that process spoken and written human language, *as language*. As we will see, this is an inclusive definition that encompasses everything from mundane applications such as word counting and automatic hyphenation, to cutting edge applications such as automated question answering on the Web, and real-time spoken language translation.

What distinguishes these language processing applications from other data processing systems is their use of *knowledge of language*.
Consider the Unix `wc` program, which is used to count the total number of bytes, words, and lines in a text file. When used to count bytes and lines, `wc` is an ordinary data processing application. However, when it is used to count the words in a file it requires *knowledge about what it means to be a word*, and thus becomes a language processing system. Of course, `wc` is an extremely simple system with an extremely limited and impoverished knowledge of language.

To summarize, the knowledge of language needed to engage in complex language behavior can be separated into six distinct categories.

1. **Phonetics and Phonology** – The study of linguistic sounds.
2. **Morphology** – The study of the meaningful components of words.
3. **Syntax** – The study of the structural relationships between words.
4. **Semantics** – The study of meaning. Pragmatics – The study of how language is used to accomplish goals.
5. **Discourse** – The study of linguistic units larger than a single utterance.

Ambiguity in Natural Language

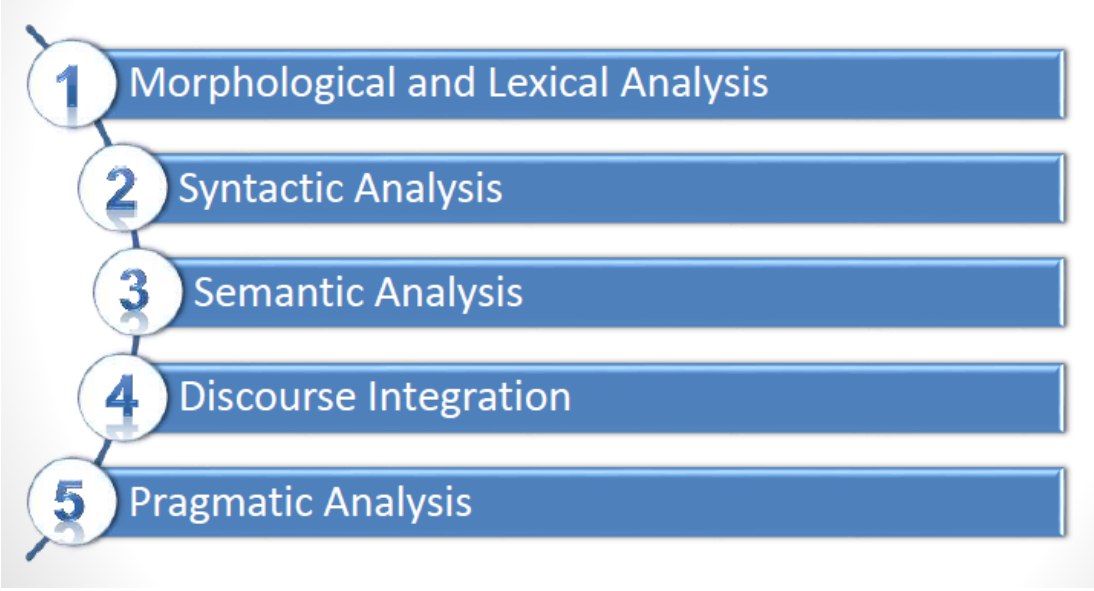
Ambiguity can occur at all NLP levels. It is a property of linguistic expressions. If an expression (word/phrase/sentence) has more than one interpretation we can refer it as ambiguous.
For eg: Consider the sentence,

"The chicken is ready to eat."

The interpretations in the above phrase can be,
The chicken (bird) is ready to be feeder or The chicken (food) is ready to be eaten.

Consider another sentence,
"There was not a single man at the party."
The interpretations in this case can be Lack of bachelors at the party or Lack of men altogether.

Stages in NLP



Morphological and Lexical Analysis

- The lexicon of a language is its vocabulary that includes its words and expressions
- Morphology depicts analyzing, identifying and description of structure of words
- Lexical analysis involves dividing a text into paragraphs, words and the sentences

Syntactic Analysis

- Syntax concerns the proper ordering of words and its affect on meaning
- This involves analysis of the words in a sentence to depict the grammatical structure of the sentence
- The words are transformed into structure that shows how the words are related to each other
- Eg. "the girl the go to the school". This would definitely be rejected by the English syntactic analyzer
- E.g. "Ravi apple eats"

Semantic Analysis

- Semantics concerns the (literal) meaning of words, phrases, and sentences
- This abstracts the dictionary meaning or the exact meaning from context
- The structures which are created by the syntactic analyzer are assigned meaning
- E.g.. "colorless blue idea" .This would be rejected by the analyzer as colorless blue do not make any sense together
- E.g. "Stone eat apple"

Discourse Integration

- Sense of the context
- The meaning of any single sentence depends upon the sentences that precedes it and also invokes the meaning of the sentences that follow it
- E.g. the word "it" in the sentence "she wanted it" depends upon the prior discourse context

Pragmatic Analysis

- Pragmatics concerns the overall communicative and social context and its effect on interpretation
- It means abstracting or deriving the purposeful use of the language in situations
- Importantly those aspects of language which require world knowledge
- The main focus is on what was said is reinterpreted on what it actually means
- E.g. "close the window?" should have been interpreted as a request rather than an order

Challenges of NLP

- Ambiguity
    - Lexical/morphological: change (V,N), training (V,N), even (ADJ, ADV) …
    - Syntactic: Helicopter powered by human flies
    - Semantic: He saw a man on the hill with a telescope.
    - Discourse: anaphora,
- Classical solution
    - Using a later analysis to solve ambiguity of an earlier step

- Eg. He gives him the change.
  - (change as verb does not work for parsing)
  - He changes the place.
    - (change as noun does not work for parsing)
- However: He saw a man on the hill with a telescope.
  - Correct multiple parsings
  - Correct semantic interpretations -> semantic ambiguity
  - Use contextual information to disambiguate (does a sentence in the text mention that "He" holds a telescope?)

Applications of NLP

- Machine Translation
- Database Access
- Information Retrieval
  - Selecting from a set of documents the ones that are relevant to a query
- Text Categorization
  - Sorting text into fixed topic categories
- Extracting data from text
  - unstructured text into structure data
- Spoken language control systems
- Spelling and grammar checkers

Some more NLP Applications

- question-answering systems, where natural language is used to query a database (for example, a query system to a personnel database)
- automated customer service over the telephone (for example, to perform banking transactions or order items from a catalogue)
- tutoring systems, where the machine interacts with a student (for example, an automated mathematics tutoring system)
- spoken language control of a machine (for example, voice control of a VCR or computer)

Ch.2. Word Level Analysis

Outlines: Morphology analysis –survey of English Morphology, Inflectional morphology & Derivational morphology, Lemmatization, Regular expression, finite automata, finite state transducers (FST), Morphological parsing with FST , Lexicon free FST Porter stemmer. N –Grams- N-gram language model, N-gram for spelling correction.

Morphology

- The study of word formation – how words are built up from smaller pieces.
- Identification, analysis, and description of the structure of a given language's MORPHEMES and other linguistic units, such as root words, affixes, parts of speech, intonations and stresses, or implied context.

Morphological analysis:

- **Token**= lemma/Stem + part of speech + grammatical features

Examples:

- cats = cat+N+plur
- played = play+V+past
- katternas = katt+N+plur+def+gen

Words are built up of minimal meaningful elements called morphemes:

- Washing= wash + ing
- Browser= browse + er
- Rats= rat + s
- played = play-ed
- cats = cat-s
- unfriendly = un-friend-ly

- Two types of morphemes:
  - Stems: play, cat, friend
  - Affixes: -ed, -s, un-, -ly

- Two main types of affixes:
  - Prefixes precede the stem: un-
  - Suffixes follow the stem: -ed, -s, un-, -ly

Types of Morphology

- **Inflectional morphology**:-modification of a word to express different grammatical categories.
  Examples- cats, men etc.
- **Derivational Morphology:-** creation of a new word from existing word by changing grammatical category.
  Examples- happiness, brotherhood etc.

Differences between Derivational and Inflectional Morphemes

- There are some differences between inflectional and derivational morphemes. First, *inflectional morphemes never change the grammatical category* (part of speech) of a word. For example, tall

and taller are both adjectives. The inflectional morpheme -er (comparative marker) simply produces a different version of the adjective tall.

- However, *derivational morphemes often change the part of speech* of a word. Thus, the verb read becomes the noun reader when we add the derivational morpheme -er.
  It is simply that read is a verb, but reader is a noun.
- For example, such derivational prefixes as re- and un- in English generally do not change the category of the word to which they are attached. Thus, both happy and unhappy are adjectives, and both fill and refill are verbs, for example. The derivational suffixes -hood and -dom, as in neighborhood and kingdom, are also the typical examples of derivational morphemes that do not change the grammatical category of a word to which they are attached.
- Second, when a derivational suffix and an inflectional suffix are added to the same word, they always appear in a certain relative order within the word. That is, inflectional suffixes follow derivational suffixes. Thus, the derivational (-er) is added to read, then the inflectional (-s) is attached to produce readers.
- Similarly, in organize– organizes the inflectional -s comes after the derivational -ize. When an inflectional suffix is added to a verb, as with organizes, then we cannot add any further derivational suffixes. It is impossible to have a form like organizesable, with inflectional -s after derivational -able because inflectional morphemes occur outside derivational morphemes and attach to the base or stem.
- A third point worth emphasizing is that certain derivational morphemes serve to create new base forms or new stems to which we can attach other derivational or inflectional affixes. For example, we use the derivational -atic to create adjectives from nouns, as in words like systematic and problematic.

Inflectional affixes always have a regular meaning. Derivational affixes may have irregular meaning. If we consider an inflectional affix like the plural 's *in word-forms like* bicycles, dogs, shoes, tins, trees, *and so on, the difference in meaning between the base and the affixed form is always the same: 'more than one'. If, however, we consider the change in meaning caused by a derivational affix like 'age* in words like *bandage*, *peerage*, *shortage*, *spillage*, and so on, it is difficult to sort of any fixed change in meaning, or even a small set of meaning changes.

Approaches to Morphology

There are three principal approaches to morphology

- Morpheme based morphology
- Lexeme based morphology
- Word based morphology

Stemming and Lemmatization

"When we are running a search, we want to find relevant results not only for the exact expression we typed on the search bar, but also for the other possible forms of the words we used. For example, it's very likely we will want to see results containing the form "shoot" if we have typed "shoots" in the search bar."

This can be achieved through two possible methods: **stemming** and **lemmatization**. The aim of both processes is the same: reducing the inflectional forms of each word into a common base or root. However, these two methods are not exactly the same

- **Stemming** algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. This indiscriminate cutting can be successful in some occasions, but not always, and that is why this approach presents some limitations.

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

Stemming

- **Lemmatization**, on the other hand, takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma.

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez

Lemmatization

How do they work?

- **Stemming:** there are different algorithms that can be used in the stemming process, but the most common in English is Porter stemmer. The rules contained in this algorithm are divided in five different phases numbered from 1 to 5. The purpose of these rules is to reduce the words to the root.
- **Lemmatization:** the key to this methodology is linguistics. To extract the proper lemma, it is necessary to look at the morphological analysis of each word. This requires having dictionaries for every language to provide that kind of analysis.

Regular Expression and Finite-state Automata

Regular expression, the standard notation for characterizing text sequences. The regular expression is used for specifying text strings in situations like this web-search example, and in other information retrieval applications, but also plays an important role in word-processing, computation of frequencies from corpora, and other such tasks.

Once we defined regular expression, they can be implemented via finite-state automaton.
The finite-state automaton is not only the mathematical device used to implement regular expressions, but also one of the most significant tools of computational linguistics. Variations of automata such as finite-state transducers, Hidden Markov Models, and *N*-gram grammars are important components of the speech recognition and synthesis, spell-checking, and information-extraction applications.

Basic Regular Expression Patterns

- The simplest kind of regular expression is a sequence of simple characters. For example, to search for *woodchuck*, we type /woodchuck/. We use the slash since this is the notation used by Perl, but the slashes are *not* part of the regular expressions. The search string can consist of a single letter (like /!/) or a sequence of letters (like /urgl/);

RE	Example Patterns Matched
/woodchucks/	"interesting links to woodchucks and lemurs"
/a/	"Mary Ann stopped by Mona's"
/Claire␣says,/	"Dagmar, my gift please," Claire says,"
/song/	"all our pretty songs"
/!/	"You've left the burglar behind again!" said Nori

- **Disjunction:** Regular expressions are **case sensitive**; lower-case /s/ is distinct from upper-case /S/; This can be solved by square braces **[** and **]**. The string of characters inside the braces specify a **disjunction** of characters to match.

RE	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	"Woodchuck"
/[abc]/	'a', 'b', *or* 'c'	"In uomini, in soldati"
/[1234567890]/	any digit	"plenty of 7 to 5"

The use of the brackets [] to specify a disjunction of characters.

- **Caret ^ :** The square braces can also be used to specify what a single character *cannot* be, by use of the caret ^.If the caret ^ is the first symbol after the open square brace [, the resulting pattern is negated.
- For *woodchuck* and *woodchucks*? cases we use the question-mark **/?/**, which means 'the preceding character or nothing'.

RE	Match (single characters)	Example Patterns Matched
[^A-Z]	not an uppercase letter	"Oyfn pripetchik"
[^Ss]	neither 'S' nor 's'	"I have no exquisite reason for't"
[^\.]	not a period	"our resident Djinn"
[e^]	either 'e' or '^'	"look up ^ now"
a^b	the pattern 'a^b'	"look up a^ b now"

Uses of the caret ^ for negation or just to mean ^

RE	Match	Example Patterns Matched
woodchucks?	woodchuck or woodchucks	"woodchuck"
colou?r	color or colour	"colour"

The question-mark ? marks optionality of the previous expression.

- **Ranges:**

Pattern	Matches	Example
[A-Z]	An uppercase letter	Charlie Brown
[a-z]	A lowercase letter	Charlie Brown
[0-9]	A single digit	Chapter 1

- **More disjunction :**Another word for raccoon is coon, the pipe | use for disjunction

Pattern	Matches
raccoon|coon	raccoon, coon
football|soccer	football, soccer
a|b|c	Same as [abc]
[Rr]accoon|[Cc]oon	Raccoon, raccoon, Coon, coon

- Operators for optional or repeated patterns, often called **Kleene** operators

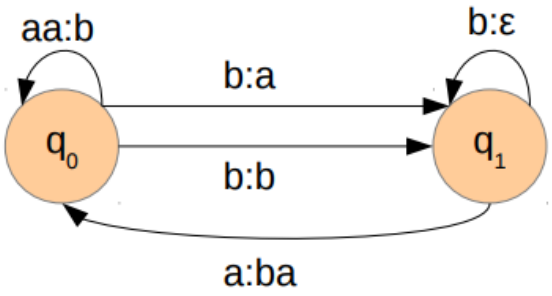
Pattern	Matches	Explanation
aa?	a, aa	0 or 1
aa*	a, aa, aaa, aaaa, ...	0 or more
a+	a, aa, aaa, aaaa, ...	1 or more
a.	aa, ab, ac, ...	any character

- **Anchors:**
  - Beginning of string ^
  - End of string $

Pattern	Examples
^[A-Z]	Charlie, Brown
^[A-Z][a-z]	Raccoon, raccoon
\.$	The end.
.$	The end!, The end?

Morphological parsing with Finite-state transducers (FST)

- FST is a type of FSA which maps between two sets of symbols.
- It is a two-tape automaton that recognizes or generates pairs of strings, one from each type.
- FST defines relations between sets of strings.
- FSAutomata have input labels. i.e. One input tape
- FSTransducers have input:output pairs on labels. i.e. Two tapes: input and output.



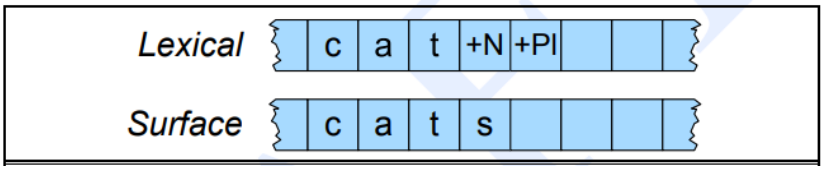
The FST is a multi-function device, and can be viewed in the following ways:

- *Translator*: It reads one string on one tape and outputs another string,
- *Recognizer*: It takes a pair of strings as two tapes and accepts/rejects based on their matching.
- *Generator*: It outputs a pair of strings on two tapes along with yes/no result based on whether they are matching or not.
- *Relater*: It compares the relation between two sets of strings available on two tapes.

- The objective of the morphological parsing is to produce output lexicons for a single input lexicon, e.g., like it is given in table 4.1.
- The second column in the table contains the stem of the corresponding word (lexicon) in first column, along with its morphological features, like, +N means word is noun, +SG means it is singular, +PL means it is plural, +V for verb, and pres-part for present participle.
- We achieve it through two level morphology, which represents a word as a correspondence between lexical level - a simple concatenation of lexicons, as shown in column 2 of table 4.1, and a surface level as shown in column 1. These are shown using two tapes of finite state transducer.

Table 4.1: Lexical Transformation table.

Input	Parsed output
cat	cat +N +SG
cats	cat +N +PL
geese	goose +N +PL
reading	read +V +Pres-part



N-gram language model

Language Models

- Formal grammars (e.g. regular, context free) give a hard "binary" model of the legal sentences in a language.
- For NLP, a ***probabilistic*** model of a language that gives a probability that a string is a member of a language is more useful.
- To specify a correct probability distribution, the probability of all sentences in a language must sum to 1.
- A language model also supports predicting the completion of a sentence.

- Please turn off your cell _____
- Your program does not _____

N-Gram Models

- Estimate probability of each word given prior context.
  - P(phone | Please turn off your cell)
- Number of parameters required grows exponentially with the number of words of prior context.
- An N-gram model uses only N−1 words of prior context.
  - Unigram: P(phone)
  - Bigram: P(phone | cell)
  - Trigram: P(phone | your cell)
- The ***Markov assumption*** is the presumption that the future behavior of a dynamical system only depends on its recent history. In particular, in a ***kth-order Markov model***, the next state only depends on the *k* most recent states, therefore an N-gram model is a (N–1)-order Markov model.

N-Gram Model Formulas

- Word sequences

$$w_1^n = w_1 \ldots w_n$$

- Chain rule of probability

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)\ldots P(w_n \mid w_1^{n-1}) = \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

- Bigram approximation

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

- N-gram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-N+1}^{k-1})$$

Estimating Probabilities

- N-gram conditional probabilities can be estimated from raw text based on the ***relative frequency*** of word sequences.

$$\textbf{Bigram:} \quad P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

$$\textbf{N-gram:} \quad P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

- To have a consistent probabilistic model, append a unique start (<s>) and end (</s>) symbol to every sentence and treat these as additional words.

- An N-gram model can be seen as a probabilistic automata for generating sentences.

Initialize sentence with N–1 <s> symbols
Until </s> is generated do:
    Stochastically pick the next word based on the conditional probability of each word given the previous N –1 words.

Example:

Let's work through an example using a mini-corpus of three sentences

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

Here are the calculations for some of the bigram probabilities from this corpus

$$P(\text{I}|\text{<s>}) = \frac{2}{3} = .67 \qquad P(\text{Sam}|\text{<s>}) = \frac{1}{3} = .33 \qquad P(\text{am}|\text{I}) = \frac{2}{3} = .67$$

$$P(\text{</s>}|\text{Sam}) = \frac{1}{2} = 0.5 \qquad P(\text{Sam}|\text{am}) = \frac{1}{2} = .5 \qquad P(\text{do}|\text{I}) = \frac{1}{3} = .33$$

Ch.3 Syntax Analysis

Part-Of-Speech Tagging(POS)

- In corpus linguistics, **part-of-speech tagging** (**POS tagging** or **PoS tagging** or **POST**), also called **grammatical tagging** or **word-category disambiguation**, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context — i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph.
- Identifying part of speech tags is much more complicated than simply mapping words to their part of speech tags. This is because POS tagging is not something that is generic.
- It is quite possible for a single word to have a different part of speech tag in different sentences based on different contexts. That is why it is impossible to have a generic mapping for POS tags.

Closed Class Vs. Open Class

- Parts-of-speech can be divided into two broad categories: closed class types and open class types.
- Closed classes are those with relatively fixed membership, such as prepositions—new prepositions are rarely coined.
- By contrast, nouns and verbs are open classes—new nouns and verbs like iPhone or to fax are continually being created or borrowed.
- Any given speaker or corpus may have different open class words, but all speakers of a language, and sufficiently large corpora, likely share the set of closed class words.
- Closed class words are generally function word words like of, it, and, or you, which tend to be very short, occur frequently, and often have structuring uses in grammar.

Why do we want to identify word classes or POS?

Someone say

- Project (PROject vs. proJECT: noun vs. verb)
- Compact (COMpact vs. comPACT: noun vs. adjective/verb)
- Content (CONtent vs. conTENT: noun vs. adjective)

Why do we want to identify them?

- Pronunciation
- Stemming
- Semantics
- More accurate N-grams
- Simple syntactic information

Tagset Corpus

Brown Corpus tagset (87 tags)
https://en.wikipedia.org/wiki/Brown_Corpus
Penn Treebank tagset (45 tags)

CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular
NNP	Proper noun, singular
NNS	Noun, plural
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Posessive ending
PRP	Personal pronoun
PP	Possessive pronoun

Penn Tree Bank tagset

RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund
VBN	Verb, past participle
VBP	Verb, non-3rd ps. sing. present
VBZ	Verb, 3rd ps. sing. present
WDT	wh-determiner
WP	wh-pronoun
WP	Possessive wh-pronoun
WRB	wh-adverb

#	Pound sign
$	Dollar sign
.	Sentence-final punctuation
,	Comma
:	Colon, semi-colon
(	Left bracket character
)	Right bracket character
"	Straight double quote
`	Left open single quote
``	Left open double quote
'	Right close single quote
''	Right close double quote

Methods for POS Tagging

1. **Rule-based Tagging**
    - (ENGTWOL)
2. **Stochastic POS Tagging**
    - Probabilistic sequence models
        - HMM (Hidden Markov Model) tagging
        - MEMMs (Maximum Entropy Markov Models)

Rule-based POS Tagging

1. Start with a dictionary
2. Assign all possible tags to words from the dictionary
3. Write rules by hand to selectively remove tags
4. Leaving the correct tag for each word.

1. **Start With a Dictionary(Eg: She promised to back the bill)**
    - **she: PRP**
    - **promised: VBN,VBD**
    - **to TO**
    - **back: VB, JJ, RB, NN**
    - **the: DT**
    - **bill: NN, VB**
2. **Assign Every Possible Tag**

NN

RB

VBN JJ VB

PRP VBD TO VB DT NN

She promised to back the bill

3. **Write Rules to Eliminate Tags**

Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"

			NN		
			RB		
	~~VBN~~		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	promised	to	back	the	bill

- **Example using ENGTWOL Tagging**

Stage 1 of ENGTWOL Tagging

First Stage: Run words through FST morphological analyzer to get all parts of speech.

Example: ***Pavlov had shown that salivation …***

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	**ADV**
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**

Stage 2 of ENGTWOL Tagging

Second Stage: Apply NEGATIVE constraints.

Example: Adverbial "that" rule

- **Eliminates all readings of "that" except the one in**
  - **"It isn't *that* odd"**

Given input: "that"
If
(+1 A/ADV/QUANT) ;if next word is adj/adv/quantifier

(+2 SENT-LIM) ;following which is E-O-S

(NOT -1 SVOC/A) ; and the previous word is not a

; verb like "consider" which

; allows adjective complements

; in "I consider that odd"

Then eliminate non-ADV tags
Else eliminate ADV

Stochastic Tagging

- Stochastic taggers generally resolve tagging ambiguities by using a training corpus to compute the probability of a given word having a given tag in a given context.
- Stochastic tagger called also **HMM tagger** or a **Maximum Likelihood Tagger**, or a **Markov model** HMM TAGGER **tagger**, based on the Hidden Markov Model.

The simplest stochastic tagger applies the following approaches for POS tagging –

Approach 1: Word Frequency Approach

In this approach, the stochastic taggers disambiguate the words based on the probability that a word occurs with a particular tag. We can also say that the tag encountered most frequently with the word in the training set is the one assigned to an ambiguous instance of that word. The main issue with this approach is that it may yield inadmissible sequence of tags.

- **Assign each word its most likely POS tag**
    - **If w has tags t1, …, tk, then can use**

      P(ti | w) = c(w,ti)/(c(w,t1) + … + c(w,tk)), where

      c(w,ti) = number of times w/ti appears in the corpus
    - **Success: 91% for English**
- **Example heat :: noun/89, verb/5**

Apprach 2: Tag Sequence Probabilities

It is another approach of stochastic tagging, where the tagger calculates the probability of a given sequence of tags occurring. It is also called n-gram approach. It is called so because the best tag for a given word is determined by the probability at which it occurs with the n previous tags.

- Given: sequence of words W
  W = w1,w2,…,wn (a sentence)
  – e.g., W = heat water in a large vessel
- Assign sequence of tags T:
  T = t1, t2, … , tn
- Find T that maximizes P(T | W)
  (For more example visit: https://www.cs.umd.edu/~nau/cmsc421/part-of-speech-tagging.pdf)

Introduction to CFG

The word **syntax** meaning ‘setting out together or arrangement’, and refers to the way words are arranged together. This chapter and the following ones introduce a number of more complex notions of syntax and grammar. There are three main new ideas: **constituency**, **grammatical relations**, and **subcategorization and dependencies**.

CONSTITUENCY

- A sequence of words that acts as a single unit
    - Noun phrases
    - Verb phrases
- These units form coherent classes that behave in similar ways
    - For example, we can say that noun phrases can come before verbs
- For example, following are all ***noun phrases*** in English...
    - Harry the Horse
    - The Broadway coppers
- I hit the man with a cleaver
  I hit [the man with a cleaver]
  I hit [the man] with a cleaver

Constituent Phrases

For constituents, we usually name them as phrases based on the word that heads the constituent:

the man from Amherst	is a Noun Phrase (NP) because the head man is a noun
extremely clever	is an Adjective Phrase (AP) because the head clever is an adjective
down the river	is a Prepositional Phrase (PP) because the head down is a preposition
killed the rabbit	is a Verb Phrase (VP) because the head killed is a verb

Context-Free Grammars

- Context –free grammars also known as
    - Phrase structure grammars
    - Backus-Naur form
- Context free grammars consist of
    - **Terminals** -words
    - **Non-terminals**-constituents in a language such as noun phrases, verb phrases and sentences.
    - **Rules**-are equations that consist of a single non-terminal on the left and any number of terminals and non-terminals on the right.

$G = \langle T, N, S, R \rangle$

- T is set of terminals (lexicon)
- N is set of non-terminals For NLP, we usually distinguish out a set $P \subset N$ of *preterminals* which always rewrite as terminals.
- S is start symbol (one of the nonterminals)
- R is rules/productions of the form $X \rightarrow \gamma$, where X is a nonterminal and γ is a sequence of terminals and nonterminals (may be empty).
- A grammar G generates a language L.

An example context-free grammar

$G = \langle T, N, S, R \rangle$

$T = \{that, this, a, the, man, book, flight, meal, include, read, does\}$

$N = \{$S, NP, NOM, VP, Det, Noun, Verb, Aux$\}$

$S = S$

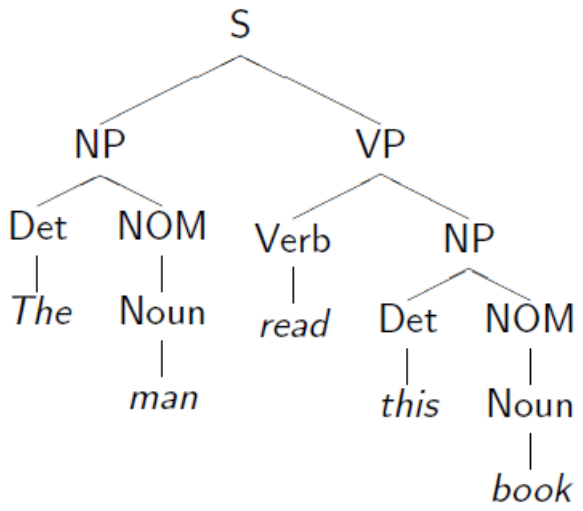
$R = \{$

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

$\}$

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*

Parse tree



Coordination

- Noun phrases and other units can be **conjoined** with **conjunctions** like *and*, *or*, and *but*.
- For example a **coordinate** noun phrase can consist of two other noun phrases separated by a conjunction (we used brackets to mark the constituents):

  Please repeat [*NP* [*NP* the flights] *and* [*NP* the costs]]
  I need to know [*NP* [*NP* the aircraft] *and* [*NP* flight number]]
  I would like to fly from Denver stopping in [*NP* [*NP* Pittsburgh] *and* [*NP* Atlanta]]

Here's a new rule for this:

$$NP \rightarrow NP \text{ and } NP$$

Similar conjunction rules can be built for *VP* and *S* conjunction:

$$VP \rightarrow VP \text{ and } VP$$
$$S \rightarrow S \text{ and } S$$

Agreement

- Most verbs in English can appear in two forms in the present tense: the form used for third-person, singular subjects (*the flight does*), and the form used for all other kinds of subjects (*all the flights do*, *I do*).
- The third-person-singular (*3sg* form usually has a final *-s* where the non-3sg form does not).
- Agreement phenomenon occurs whenever there is a verb that has some noun acting as its subject.
- Note that sentences in which the subject does not agree with the verb are ungrammatical:

  What flights *leave* in the morning?
  What flight *leaves* from Pittsburgh?
- To handle these agreement phenomena we can modify grammer with multiple set of rules,
- one rule set for *3sg* subjects, and one for non-*3sg* subjects.

For example, the rule that handled these yes-no-questions (1. Does [*NP* you] have a flight from Boston to Forth Worth? 2. Do [*NP* this flight] stop in Dallas?) used to look like this:

$$S \rightarrow Aux \ NP \ VP$$

We could replace this with two rules of the following form:

$S \rightarrow 3sgAux\ \ 3sgNP\ \ VP$

$S \rightarrow Non3sgAux\ \ Non3sgNP\ \ VP$

Verb Phrases and Subcategorization

- English *VP*s consist of a head verb along with 0 or more following constituents which we'll call ***arguments***.

$VP \rightarrow Verb$ disappear
$VP \rightarrow Verb\ NP$ prefer a morning flight
$VP \rightarrow Verb\ NP\ PP$ leave Boston in the morning
$VP \rightarrow Verb\ PP$ leaving on Thursday

- But, even though there are many valid VP rules in English, not all verbs are allowed to participate in all those VP rules.
- We can subcategorize the verbs in a language according to the sets of VP rules that they participate in.
- This is a modern take on the traditional notion of transitive/intransitive.
- Modern grammars may have 100s or such classes.
- Subcategorization examples,
  - Sneeze: John sneezed
  - Find: Please find [a flight to NY]$_{NP}$
  - Give: Give [me]$_{NP}$[a cheaper fare]$_{NP}$
  - Help: Can you help [me]$_{NP}$[with a flight]$_{PP}$
  - Prefer: I prefer [to leave earlier]$_{TO\text{-}VP}$
  - Told: I was told [United has a flight]$_{S}$

Sequence labeling

- The task of assigning label sequences to a set of observation sequences arises in many fields, including bioinformatics, computational linguistics and speech recognition.
- For example, consider the natural language processing task of labeling the words in a sentence with their corresponding part-of-speech (POS) tags.
- In this task, each word is labeled with a tag indicating its appropriate part of speech, resulting in annotated text, such as:

  [PRP He] [VBZ reckons] [DT the] [JJ current] [NN account] [NN deficit] [MD will] [VB narrow] [TO to] [RB only] [# #] [CD 1.8] [CD billion] [IN in] [NNP September] [. .]

- One of the most common methods for performing such labeling and segmentation tasks is that of employing hidden Markov models (HMMs) or probabilistic finite-state automata to identify the most likely sequence of labels for the words in any given sentence.

Hidden Markov Model

- HMMs are a form of generative model, that defines a joint probability distribution p(X,Y) where X and Y are random variables respectively ranging over observation sequences and their corresponding label sequences.
- In order to define a joint distribution of this nature, generative models must enumerate all possible observation sequences – a task which, for most domains, is intractable unless observation elements are represented as isolated units, independent from the other elements in an observation sequence.
- More precisely, the observation element at any given instant in time may only directly depend on the state, or label, at that time.
- This is an appropriate assumption for a few simple data sets, however most real-world observation sequences are best represented in terms of multiple interacting features and long-range dependencies between observation elements.

Conditional Random Field

- Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data, based on the conditional approach described in hidden markov model.
- A CRF is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence.
- **The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference.**
- **Additionally, CRFs avoid the label bias problem; a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models.**
- **CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks.**
- The graphical structure of a conditional random field may be used to factorize the joint distribution over elements Y_v of Y into a normalized product of strictly positive, real-valued potential functions, derived from the notion of conditional independence.
- The probability of a particular label sequence ***y*** given observation sequence ***x*** to be a normalized product of potential functions, each of the form

$$\exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_k \mu_k s_k(y_i, \boldsymbol{x}, i)\right),$$

- Where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of the entire observation sequence and the labels at positions i and i-1 in the label sequence;
- $s_k(y_i, x, i)$ is a state feature function of the label at position i and the observation sequence;
- And λ_j and μ_k are parameters to be estimated from training data.

- When defining feature functions, we construct a set of real-valued features b(x, i) of the observation to expresses some characteristic of the empirical distribution of the training data that should also hold of the model distribution.
- An example of such a feature is

$$b(\boldsymbol{x}, i) = \begin{cases} 1 & \text{if the observation at position } i \text{ is the word "September"} \\ 0 & \text{otherwise.} \end{cases}$$

- Each feature function takes on the value of one of these real-valued observation features b(x, i) if the current state (in the case of a state function) or previous and current states (in the case of a transition function) take on particular values.
- All feature functions are therefore real-valued. For example, consider the following transition function:

$$t_j(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} b(\boldsymbol{x}, i) & \text{if } y_{i-1} = \text{IN and } y_i = \text{NNP} \\ 0 & \text{otherwise.} \end{cases}$$

In the remainder of this report, notation is simplified by writing

$$s(y_i, \boldsymbol{x}, i) = s(y_{i-1}, y_i, \boldsymbol{x}, i)$$

and

$$F_j(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \boldsymbol{x}, i),$$

where each $f_j(y_{i-1}, y_i, \boldsymbol{x}, i)$ is either a state function $s(y_{i-1}, y_i, \boldsymbol{x}, i)$ or a transition function $t(y_{i-1}, y_i, \boldsymbol{x}, i)$. This allows the probability of a label sequence $\boldsymbol{y}$ given an observation sequence $\boldsymbol{x}$ to be written as

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_j \lambda_j F_j(\boldsymbol{y}, \boldsymbol{x})\right).$$

$Z(\boldsymbol{x})$ is a normalization factor.

Maximum Entropy

- The form of a CRF, as given in, is heavily motivated by the principle of maximum entropy – a framework for estimating probability distributions from a set of training data.
- Entropy of a probability distribution is a measure of uncertainty and is maximized when the distribution in question is as uniform as possible.
- The principle of maximum entropy asserts that the only probability distribution that can justifiably be constructed from incomplete information, such as finite training data, is that which has maximum entropy subject to a set of constraints representing the information available.

Ch.4 Semantic Analysis

Semantic analysis, the process whereby meaning representations of the kind are composed and assigned to linguistic inputs.

The first approach is a kind of **syntax-driven semantic analysis** that is fairly limited in its scope. It assigns meaning representations to inputs based solely on static knowledge from the lexicon and the grammar. In this approach, when we refer to an input's meaning, or meaning representation, we have in mind an impoverished representation that is both contexts independent and inference-free.

- Syntax-driven semantic analysis is based on the principle of compositionality.
    - Meaning of the whole built up from the meaning of the parts
    - more specifically, in a way that is guided by word order and syntactic relations.
- Build up the MR by augmenting CFG rules with semantic composition rules.
- Representation produced is literal meaning: context independent and free of inference

Attachments for a fragment of English

This section describes a set of semantic attachments for a small fragment of English. To keep the presentation simple, we omit the feature structures associated with these rules when they are not needed. Remember that these features are needed to ensure that the correct rules are applied in the correct situations. Most importantly for this discussion, they are needed to ensure that the correct verb entries are being employed based on their subcategorization feature structures.

Sentences

- For the most part, our semantic discussions have only dealt with declarative sentences.
- This section expands coverage to include the other sentence types: imperatives, Yes/No questions, and WH questions.
- Let's start by considering the following examples.
    - Flight 487 serves lunch.
    - Serve lunch.
    - Does Flight 207 serve lunch?
    - Which flights serve lunch?
- The meaning representations of these examples all contain propositions concerning the serving of lunch on flights.
- However, they differ with respect to the role that these propositions are intended to serve in the settings in which they are uttered.
- More specifically, the first example is intended to convey factual information to a hearer, the second is a request for an action, and the last two are requests for information.
- To capture these differences, we will introduce a set of operators that can be applied to FOPC sentences in the same way those belief operators. Specifically, the operators *DCL*, *IMP*, *YNQ*, and *WHQ* will be applied to the FOPC representations of declaratives, imperatives, yes-no questions, and wh-questions, respectively.
- Producing meaning representations that make appropriate use of these operators requires the right set of semantic attachments for each of the possible sentence types.

- For declarative sentences, we can simply alter the basic sentence rule we have been using as follows.

$$S \rightarrow NP\ VP \qquad \{DCL(VP.sem(NP.sem))\}$$

- **Imperative** sentences begin with a verb phrase and lack an overt subject.
- The *IMP* operator can then be applied to this representation as in the following semantic attachment.

$$S \rightarrow VP \qquad \{IMP(VP.sem(DummyYou))\}$$

- **Yes-no-questions** consist of a sentenceinitial auxiliary verb, followed by a subject noun phrase and then a verb phrase.

$$S \rightarrow Aux\ NP\ VP \qquad \{YNQ(VP.sem(NP.sem))\}$$

- Unlike yes-no-questions, **wh-subject-questions** ask for specific information about the subject of the sentence rather than the sentence as a whole.

$$S \rightarrow WhWord\ NP\ VP \qquad \{WHQ(NP.sem.var, VP.sem(NP.sem))\}$$

Noun Phrases

- The meaning representations for noun phrases can be either normal FOPC terms or complex-terms.
- **Compound Nominals:** Compound nominals, also known as noun-noun sequences, consist of simple sequences of nouns, as in the following examples.

  Flight schedule
  Summer flight schedule
- The syntactic structure of this construction can be captured by the regular expression *Noun* or by the following context-free grammar rules.

$$Nominal \rightarrow Noun$$
$$Nominal \rightarrow Noun\ Nominal$$

Rule

$$Nominal \rightarrow Noun\ Nominal$$
$$\{\lambda x\ Nominal.sem(x) \wedge NN(Noun.sem, x)\}$$

- **Genitive Noun Phrases:** genitive noun phrases make use of complex determiners that consist of noun phrases with possessive markers, as in *Atlanta's airport* and *Maharani's menu*.

$$NP \rightarrow ComplexDet\ Nominal$$
$$\{< \exists x Nominal.sem(x) \wedge GN(x, ComplexDet.sem) >\}$$

Adjective Phrases

- English adjectives can be split into two major categories: pre-nominal and predicate.
- These categories are exemplified by the following BERP examples.

  I don't mind a cheap restaurant.

  This restaurant is cheap.
- For the pre-nominal case, an obvious *and often incorrect* proposal for the semantic attachment is illustrated in the following rules.

$$Nominal \rightarrow Adj\ Nominal$$
$$\{\lambda x\ Nominal.sem(x) \wedge Isa(x, Adj.sem)\}$$

$$Adj \rightarrow cheap \qquad \{Cheap\}$$

- For our cheap restaurant example, this yields the following fairly reasonable representation.

$$\lambda x\ Isa(x, Restaurant) \wedge Isa(x, Cheap)$$

Verb Phrases

- The general schema for computing the semantics of verb phrases relies on the notion of function application.
- **Infinitive Verb Phrases:** A fair number of English verbs take some form of verb phrase as one of their arguments.
- Consider the following example.

  I told Harry to go to Maharani.
- The meaning representation for this example should be something like the following.

$$\exists e, f, x\ Isa(e, Telling) \wedge Isa(f, Going)$$
$$\wedge Teller(e, Speaker) \wedge Tellee(e, Harry) \wedge ToldThing(e, f)$$
$$\wedge Goer(f, Harry) \wedge Destination(f, x)$$

Prepositional Phrases

- Prepositional phrases serve two distinct functions: they assert binary relations between their heads and the constituents to which they are attached, and they signal arguments to constituents that have an argument structure.
- These two functions argue for two distinct types of prepositional phrases that differ based on their semantic attachments.
- Prepositional phrases serve these roles: modifiers of noun phrases, modifiers of verb phrases, and arguments to verb phrases.
- **Nominal Modifier Prepositional Phrases:** Modifier prepositional phrases denote a binary relation between the concepts being modified, which is external to the prepositional phrase, and the head of the prepositional phrase.
- Consider the following example and its associated meaning representation.

  A restaurant on Pearl

$$\exists x\ Isa(x, Restaurant) \wedge On(x, Pearl)$$

- The relevant grammar rules that govern this example are the following.

$$NP \rightarrow Det\ Nominal$$
$$Nominal \rightarrow Nominal\ PP$$
$$PP \rightarrow P\ NP$$

- **Verb Phrase Modifier Prepositional Phrases:** The general approach to modifying verb phrases is similar to that of modifying nominals.
- The differences lie in the details of the modification in the verb phrase rule; the attachments for the preposition and prepositional phrase rules are unchanged.
- Let's consider the phrase *ate dinner in a hurry* which is governed by the following verb phrase rule.

$$VP \rightarrow VP\ PP$$

- The meaning representation of the verb phrase constituent in this construction, *ate dinner*, is a λ-expression where the λ variable represents the as yet unseen subject.

$$\lambda x \exists e\ Isa(e, Eating) \wedge Eater(e, x) \wedge Eaten(e, Dinner)$$

Lexical Semantics

- A **lexicon** generally has a highly structured form
  - It stores the meanings and uses of each word
  - It encodes the relations between words and meanings
- A **lexeme** is the minimal unit represented in the lexicon
  - It pairs a stem (the orthographic/phonological form chosen to represent words) with a symbolic form for meaning representation (**sense**)
- A **dictionary** is a kind of lexicon where meanings are expressed through definitions and examples

son noun
a boy or man in relation to either or both of his parents.
• a male offspring of an animal.
• a male descendant : *the sons of Adam.*
• (the Son) (in Christian belief) the second person of the Trinity; Christ.
• a man considered in relation to his native country or area : *one of Nevada's most famous sons.*
• a man regarded as the product of a particular person, influence, or environment : *sons of the French Revolution.*
• (also my son) used by an elder person as a form of address for a boy or young man : "*You're on private land, son.*"

Lexicons & dictionaries

- Definitions in dictionaries exploit words and they may be circular (a word definition uses words whose definitions exploit that word)

right adj.
1. of, relating to, situated on, or being the side of the body which is away from the side on which the heart is mostly located
2. located nearer to the **right** hand than to the left
3. done with the **right** hand

.....

- The paradox is that the dictionary elements are not direct definitions
  - The are description of the lexemes made up of other lexemes assuming that the user has enough information on these other terms!
  - This approach would fail without the assumption that the user has already enough a priori knowledge deriving from the real world.
  - However the description provide a great amount of information on the relationships among the words allowing to perform semantic inferences

Relations among lexemes & their senses

- Several kinds of relationships can be defined between lexemes and senses (some of them are important for automatic processing)

- **Homonymy**
  It is a relation between words that have the same form (and the same PoS) but unrelated meanings
  - e.g. *bank* (the financial institution, the river bank)
  - It causes ambiguities for the interpretation of a sentence since it defines a set of different lexemes with the same orthographic form (bank1, bank2,..)
  - Related properties are homophony (same pronunciation but different orthography, e.g. *be-bee*) and homography (same orthography but different pronunciation pésca/pèsca)

- **Polysemy**

  It happens when a lexeme has more related meanings
  - It depends on the word etymology (unrelated meanings usually have a different origin) - e.g. *bank/data bank/blood bank*

  For polysemous lexemes we need to manage all the meanings
  - We should define a method to determine the meanings (their number and semantics) and if they are really distinct (by experts in lexicography)
  - We need to describe the eventual correlations among the meanings
  - We need to define how the meanings can be distinguished in order to attach the correct meaning to a word in a given context (word sense disambiguation)

- **Synonymy**

  It is a relationship between two distinct lexemes with the same meaning (i.e. they can be substituted for one another in a given context without changing its meaning and correctness)
  – e.g. I received a gift/present
  - The substitutability may not be valid for any context due to small semantic differences (e.g. *price/fare of a service – the bus fare/the ticket price*)
  - In general substitutability depends on the "semantic intersection" of the senses of the two lexemes and, in some cases, also by social factors (*father/dad*)

- **Hyponymy**
- It is a relationship between two lexemes (more precisely two senses) such that one denotes a subclass of the other
  - car, vehicle – shark, fish – apple, fruit
  - The relationship is not symmetric
    - The more specialized concept is the hyponym of the more general one
    - The more general concept is the hypernym of the more specialized one
  - Hyponymy (hypernymy) is the basis for the definition of a taxonomy (a tree structure that defines inclusion relationships in an object ontology) even if it is not properly a taxonomy
    - The definition of a formal taxonomy would require a more uniform/rigorous formalism in the interpretation of the inclusion relationship
    - However the relationship defines a inheritance mechanism of the properties from the ancestors of a given a concept in the hierarchy

WordNet

- It is a lexical database for English (versions for other languages are available) organized as a **semantic network of senses**
  - It represents nouns, verbs, adjectives, and adverbs but it does not include functional terms in the closed classes (prepositions, conjunctions, etc.)
  - The lexemes are grouped into sets of cognitive synonyms (synset), each representing a distinct concept
  - A set of senses (synset) is associated to each lexeme (unique orthographic form)
  - Synsets are linked by conceptual/semantic and lexical relationships
  - Wordnet consists in lexicographic files, an application to load these files into a database and a library of search and browsing functions to visualize and access the database contents.

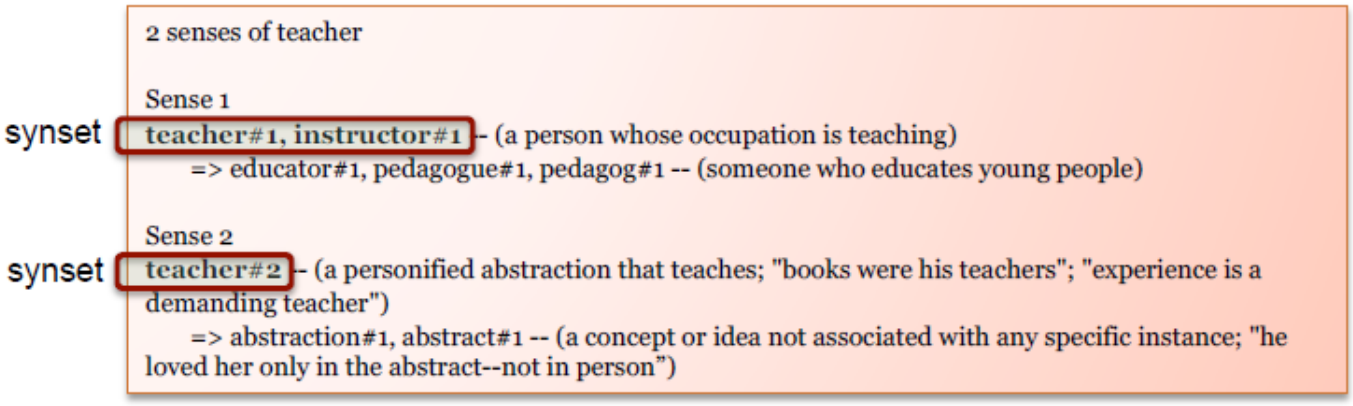
Wordnet Statistics

PoS	Unique strings	Synset	pairs word-sense
Noun	117,798	82,115	146,312
Verb	11,529	13,767	25,047
Adjective	21,479	18,156	30,002
Adverb	4,481	3,621	5,580
Total	155,287	117,659	206,941

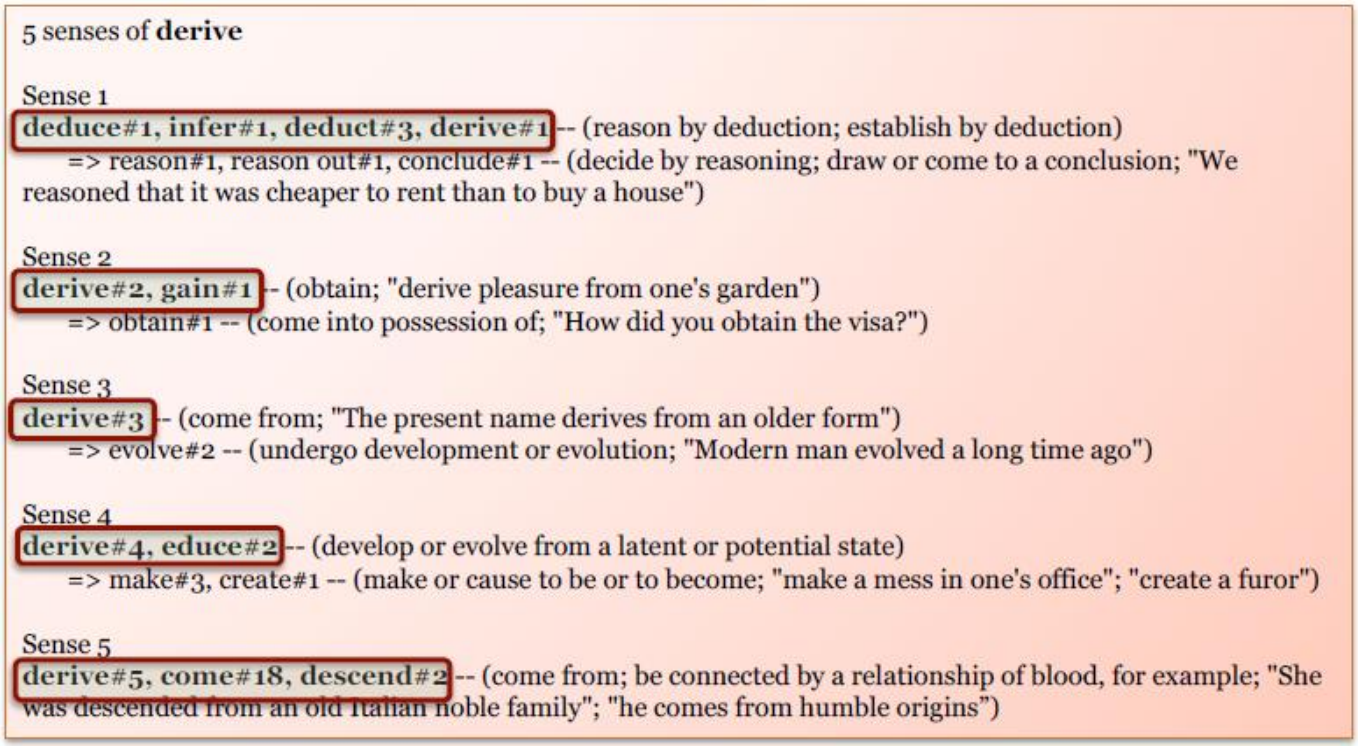
- Nouns have an average of 1.24 senses, verbs 2.17, adjectives 1.40, adverbs 1.25
- The actual total number of distinct strings is 147,278 (the same string can belong to more than one PoS class)

Synset

- A synset is a set of synonyms that define a concept or word meaning
  - About half of the synsets (~54%) contains only one term, about one third (~29%) 2 terms, about 10% 3 terms
  - An annotation (gloss) explaining the meaning is associated to each synset (especially to those containing a single term)
    - A synset contains ~1.75 terms in average)



Synset – verb example



Names

- Names are organized in a hierarchy of specializations (hyponyms) and generalizations (hypernyms)
  - In the 3.0 version there is a unique root category {entity} referred to as **unique beginner** whereas in the previous versions there are mode unique beginners (25 in version 1.7.1)

vase#1 -- (an open jar of glass or porcelain used as an ornament or to hold flowers)
=> **jar#1** -- (a vessel (usually cylindrical) with a wide mouth and without handles)
=> **vessel#3** -- (an object used as a container (especially for liquids))
=> **container#1** -- (any object that can be used to hold things (especially a large metal boxlike object of standardized dimensions that can be loaded from one form of transport to another))
=> **instrumentality#3, instrumentation#1** -- (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
=> **artifact#1, artefact#1** -- (a man-made object taken as a whole)
=> **whole#2, unit#6** -- (an assemblage of parts that is regarded as a single entity; "how big is that part compared to the whole?"; "the team is a unit")
=> **object#1, physical object#1** -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls and other objects")
=> **physical entity#1** -- (an entity that has physical existence)
=> **entity#1** -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Wordnet relationships– names & verbs

- For nouns the following relationships are provided among synsets
  - Hyperonymy - X is a kind of Y (car → vehicle)
  - Hyponymy – Y is a kind of X (vehicle → car)
  - Coordinate terms– Y is a coordinate term of X is X and Y share a common hyperonym (car and motorcycle)
  - Holonymy – X is part of Y (wheel → bicycle)
  - Meronymy – Y is part of X (bicycle → wheel)
- For verbs the following relationships are provided among synsets
  - Hyperonymy – the activity of X is a kind of Y (to see → to perceive)
  - Troponymy – the activity Y executes X in some sense (to eat → to devour)
  - Entailment – Y is required to perform X (to snore→to sleep)
  - Coordinate terms– Terms share a common hyperonym (to hear-to see as cases of to perceive)

Wordnet relations– adjectives & adverbs

- Words can be linked to other words by lexical relationships such as antonomy (words that have opposite meanings)
  - good ⇔ bad, day ⇔ night, exit ⇔ entrance
- For adjectives the following relationships are defined
  - Related nouns– (noisy → noise)
  - Similar to– (noisy → clanking)
    - The **descriptive adjectives** are organized into groups containing a main synset (head) and satellite synsets. Each group is organized around a pair (sometimes a triple) of antonyms corresponding to the main terms. The satellite synsets are those linked by the "*Similar to*" relationship.
    - Relational adjectives are used to categorize the noun and they have neither a group structure nor an antonym (e.g. musical, nervous)
- For the adverbs the following relationships are defined
  - Base adjective– (slowly → slow)

Robust Word sense disambiguation

- Word sense disambiguation (WSD) is the task of selecting the correct sense for a word in a given sentence
  - This problem has to be faced for words having more meanings

- It requires a dictionary listing all the possible senses for each word
- It can be faced for each single word or jointly for all the words in the sentence (all the meaning combinations should be considered)

I ate a cold **dish**
I washed a dirty **dish**
The *served* a cold **dish**

- Several approached to WSD have been proposed
  - (Machine Readable) Dictionary and knowledge-based, Machine Learning Supervised methods, Semi-supervised and Unsupervised methods

Supervised learning

- WSD can be approached as a classification task
  - The correct sense is the class to be predicted
  - The word is represented by a set (vector) of features to be processed as the classifier input
    - Usually the feature include a representation of the word to be disambiguated (target) and of its context (a given number of words at the left and the right of the target word)
    - The word itself, the word stem, the word PoS can be exploited as features
  - The classifier can be learnt from examples given a labeled dataset
  - Different models can be exploited to implement the classifier (Naïve Bayes, neural networks, decision trees…)
  - The limitation of the learning based approach is scalability when a large number of labeled examples is required

Naïve Bayes

- The bayesian approach aims at maximizing of the probability of sense *s* given the feature vector *fw* describing the target word

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}}\, p(s|f_w) = \underset{s \in S}{\operatorname{argmax}}\, \frac{p(f_w|s)p(s)}{p(f_w)}$$

  - With the simplifying assumption that the feature vector entries (words in context) are independent of each other p(*fw*|*s*) can be written as

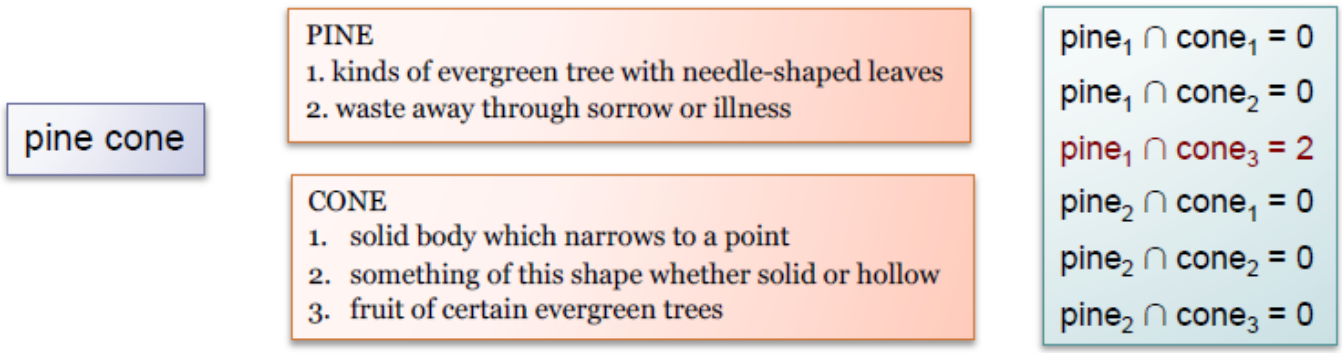
$$p(f_w|s) = \prod_{j=1}^{n} p(f_j|s)$$

    - the probabilities p(*fj*|*s*) model the statistics for distribution of feature j (e.g. a given word) in the context of word w when having the sense s.
    - p(s) is the a priori probability of each sense of the word

Dictionary-based methods

- A dictionary can provide useful information about the contexts related to the word senses (glosses)
- As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base.
- They do not use corpora evidences for disambiguation.

- The Lesk method is the seminal dictionary-based method introduced by Michael Lesk in 1986.
- The Lesk definition, on which the Lesk algorithm is based is **"measure overlap between sense definitions for all words in context"**.
    - A simple approach is the Lesk algorithm (1986)
        - The algorithm computes the intersection among the glosses associated to the different meanings of the words in the sentence
        - The combination yielding the maximum overall intersection is selected (the complexity is combinatorial in the number of senses)



Limitations of the Lesk algorithm

- The Lesk algorithm yields a 50-70% accuracy
    - The main limitation is its dependence on the quality of the glosses/examples provided for each sense in the dictionary since they are usually short and do not carry enough information to train a classifier
        - The words in the context and their definition should share a significant intersection (they should share the maximum number of terms)
    - The coverage can be improved by adding the words related to the target but not already contained in the glosses
        - for example the words of the definitions containing the target word only when the actual sense of the target word is clear in that context
    - In the computation of the intersection/similarity among the context more flexible measure can be exploited
        - Correlation with TermFrequency-InverseDocumentFrequency weights in order to reduce the importance of most common words.

Ch.5 Pragmatics

- **Pragmatics** is the study of (some parts of) the relation between language and context-of-use. Context-of-use includes such things as the identities of people and objects, and so pragmatics includes studies of how language is used to refer (and re-refer) to people and things.
- Context-of-use includes the discourse context, and so pragmatics includes studies of how discourses are structured, and how the listener manages to interpret a conversational partner in a conversation.

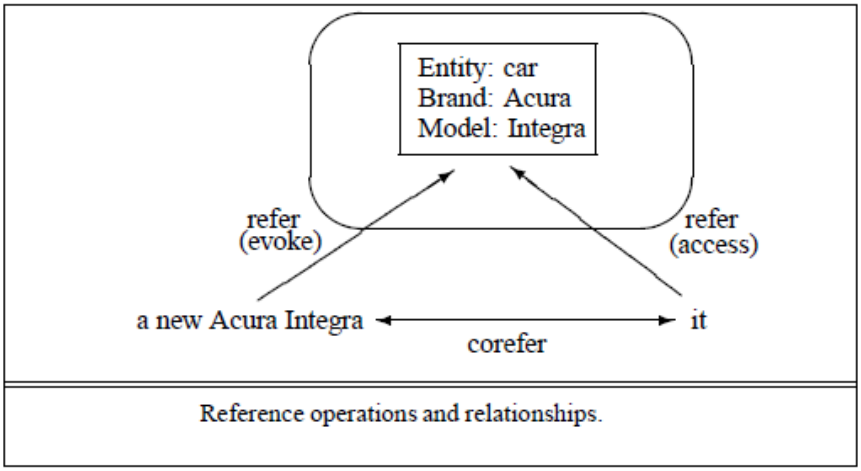
Discourse

- Earlier stages of natural language processing focused primarily on language phenomena that operate at the word or sentence level.
- Of course, language does not normally consist of isolated, unrelated sentences, but instead of collocated, related groups of sentences.
- Such a group of sentences refer as a **discourse**.
- Here we will study discourse of particular sort: a **monologue**
- Monologues are characterized by a *speaker* (a term which will be used to include writers, as it is here), and a *hearer* (which, analogously, includes readers).
- The communication flows in only one direction in a monologue, that is, from the speaker to the hearer.
- Consider the discourse shown in example,
  "(18.1)John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour."
- In following section, we describe a method where pronoun *he* referred to john and car but not bill and *it* denotes to Acura Integra car but not bill's car dealership.

Reference resolution

- In this section we will study problem of **reference**, the process by which REFERENCE speakers use expressions like *John* and *he* in passage (18.1) to denote a person named John.
- A natural language expression used to perform reference is called a **referring expression**, and the entity that is referred to is called the **referent**.
- Thus, *John* and *he* in passage (18.1) are referring expressions, and John is REFERENT their referent.
- Two referring expressions that are used to refer to the same entity are said to **corefer**, thus *John* and *he* are corefer.
- The discourse model contains representations of the entities that have been referred to in the discourse and the relationships in which they participate.
- Thus, there are two components required by a system to successfully produce and interpret referring expressions: a method for constructing a discourse model that evolves with the dynamically-changing discourse it represents, and a method for mapping between the signals that various referring expressions encode and the hearer's set of beliefs, the latter of which includes this discourse model.
- We will speak in terms of two fundamental operations to the discourse model.

- When a referent is first mentioned in a discourse, we say that a representation for it is **evoked** into the model.
- Upon subsequent mention, this representation is **accessed** from the model. The operations and relationships are illustrated in Figure



Reference operations and relationships.

Reference Phenomena

- The set of referential phenomena that natural languages provide is quite rich indeed.
- In this section, we provide a brief description of several basic reference phenomena.
- We first survey five types of referring expression: *indefinite noun phrases*, *definite noun phrases*, *pronouns*, *demonstratives*, and *one-anaphora*.
- We then describe three types of referents that complicate the reference resolution problem: *inferrables*, *discontinuous sets*, and *generics*.

Indefinite Noun Phrases

- Indefinite reference introduces entities that are new to the hearer into the discourse context.
- The most common form of indefinite reference is marked with the determiner *a* (or *an*), as in (1), but it can also be marked by a quantifier such as *some* (2) or even the determiner *this* (3).
  (1) I saw an Acura Integra today.
  (2) Some Acura Integras were being unloaded at the local dealership today.
  (3) I saw this awesome Acura Integra today.
- Such noun phrases evoke a representation for a new entity that satisfies the given description into the discourse model.

Definite Noun Phrases

- Definite reference is used to refer to an entity that is identifiable to the hearer, either because it has already been mentioned in the discourse context (and thus is represented in the discourse model), it is contained in the hearer's set of beliefs about the world, or the uniqueness of the object is implied by the description itself.
- The case in which the referent is identifiable from discourse context is shown in (4).
  (4) I saw an Acura Integra today. *The Integra* was white and needed to be washed.

- Definite noun phrase reference requires that an entity be accessed from either the discourse model or the hearer's set of beliefs about the world.

Pronouns

- Another form of definite reference is pronominalization, illustrated in example (5).
  (5) I saw an Acura Integra today. *It* was white and needed to be washed.
- The constraints on using pronominal reference are stronger than for full definite noun phrases, requiring that the referent have a high degree of activation or **salience** in the discourse model.
- Pronouns usually (but not always) refer to entities that were introduced no further than one or two sentences back in the ongoing discourse, whereas definite noun phrases can often refer further back.
- This is illustrated by the difference between sentences (6d) and (6d').
  6d. ?? He also said that he bought *it* yesterday.
  6d'. He also said that he bought *the Acura* yesterday.

Demonstratives

- Demonstrative pronouns, like *this* and *that*, behave somewhat differently that simple definite pronouns like *it*.
- They can appear either alone or as determiners, for instance, *this Acura, that Acura*.
- The choice between two demonstratives is generally associated with some notion of spatial proximity: *this* indicating closeness and *that* signaling distance.
- Spatial distance might be measured with respect to the discourse participants' situational context, as in (7).
  (7) [John shows Bob an Acura Integra and a Mazda Miata] Bob (pointing): I like *this* better than *that*.

One Anaphora

- *One*-anaphora, exemplified in (8), blends properties of definite and indefinite reference
  (8) I saw no less than 6 Acura Integras today. Now I want *one*.
- This use of *one* can be roughly paraphrased by *one of them*, in which *them* refers to a plural referent (or generic one, as in the case of (8), see below), and *one* selects a member from this set.
- Thus, *one* may evoke a new entity into the discourse model, but it is necessarily dependent on an existing referent for the description of this new entity.

Inferrables

- In some cases a referring expression does not refer to an entity that has been explicitly evoked in the text, but instead one that is inferentially related to an evoked entity.Such referents are called *inferrables.*
- Consider following expressions *door* and *engine* in sentence(9)

(9) I almost bought an Acura Integra today, but *a door* had a dent and *the engine* seemed noisy.

Discontinuous Sets

- In some cases, references using plural referring expressions like *they* and *them* refer to sets of entities that are evoked together, for instance (10), using another plural expression (*their Acuras*) or a conjoined noun phrase (*John and Mary*):
  (10)John and Mary love their Acuras. They drive them all the time.

Generics

- Making the reference problem even more complicated is the existence of *generic* reference. Consider example (11)
  (11) I saw no less than 6 Acura Integras today. *They* are the coolest cars.
- Here, the most natural reading is not the one in which *they* refers to the particular 6 Integras mentioned in the first sentence, but instead to the class of Integras in general.

Syntactic and Semantic Constraints on Coreference

- Having described a variety of reference phenomena that are found in natural language, we can now consider how one might develop algorithms for identifying the referents of referential expressions.
- One step that needs to be taken in any successful reference resolution algorithm is to filter the set of possible referents on the basis of certain relatively hard-and-fast constraints.
- We describe some of these constraints here.

Number Agreement

- Referring expressions and their referents must agree in number; for English, this means distinguishing between *singular* and *plural* references.
- A categorization of pronouns with respect to number is shown in Figure below.

Singular	Plural	Unspecified
she, her, he, him, his, it	we, us, they, them	you

Number agreement in the English pronominal system.

- The following examples illustrate constraints on number agreement.
  John has a new Acura. It is red.
  John has three new Acuras. They are red.
  * John has a new Acura. They are red.
  * John has three new Acuras. It is red.

Person and Case Agreement

- English distinguishes between three forms of person: first, second, and third.
- A categorization of pronoun types with respect to person is shown in Figure below.
- The following examples illustrate constraints on person agreement.
  You and I have Acuras. We love them.
  John and Mary have Acuras. They love them.
  * John and Mary have Acuras. We love them. (where *We*=John and Mary)
  * You and I have Acuras. They love them. (where *They*=You and I)

	First	Second	Third
Nominative	I, we	you	he, she, they
Accusative	me, us	you	him, her, them
Genitive	my, our	your	his, her, their

Person and case agreement in the English pronominal system/

Gender Agreement

- Referents also must agree with the gender specified by the referring expression.
- English third person pronouns distinguish between *male*, *female*, and *nonpersonal* genders, and unlike many languages, the first two only apply to animate entities.
- Some examples are shown in Figure below.

masculine	feminine	nonpersonal
he, him, his	she, her	it

Gender agreement in the English pronominal system.

- The following examples illustrate constraints on gender agreement.

  John has an Acura. He is attractive. (he=John, not the Acura)
  John has an Acura. It is attractive. (it=the Acura, not John)

Syntactic Constraints

- Reference relations may also be constrained by the syntactic relationships between a referential expression and a possible antecedent noun phrase when both occur in the same sentence.
- For instance, the pronouns in all of the following sentences are subject to the constraints indicated in brackets.

  John bought himself a new Acura. [himself = John]
  John bought him a new Acura. [him ≠ John]
  John said that Bill bought him a new Acura. [him ≠ Bill]
  John said that Bill bought himself a new Acura. [himself = Bill]
  He said that he bought John a new Acura. [He ≠ John; he ≠ John]