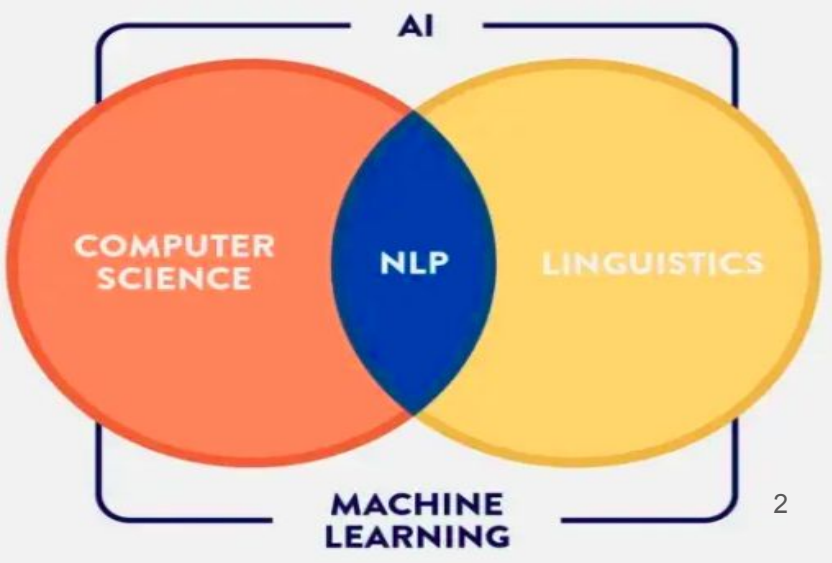
Introduction to Natural language Processing

Unit-I

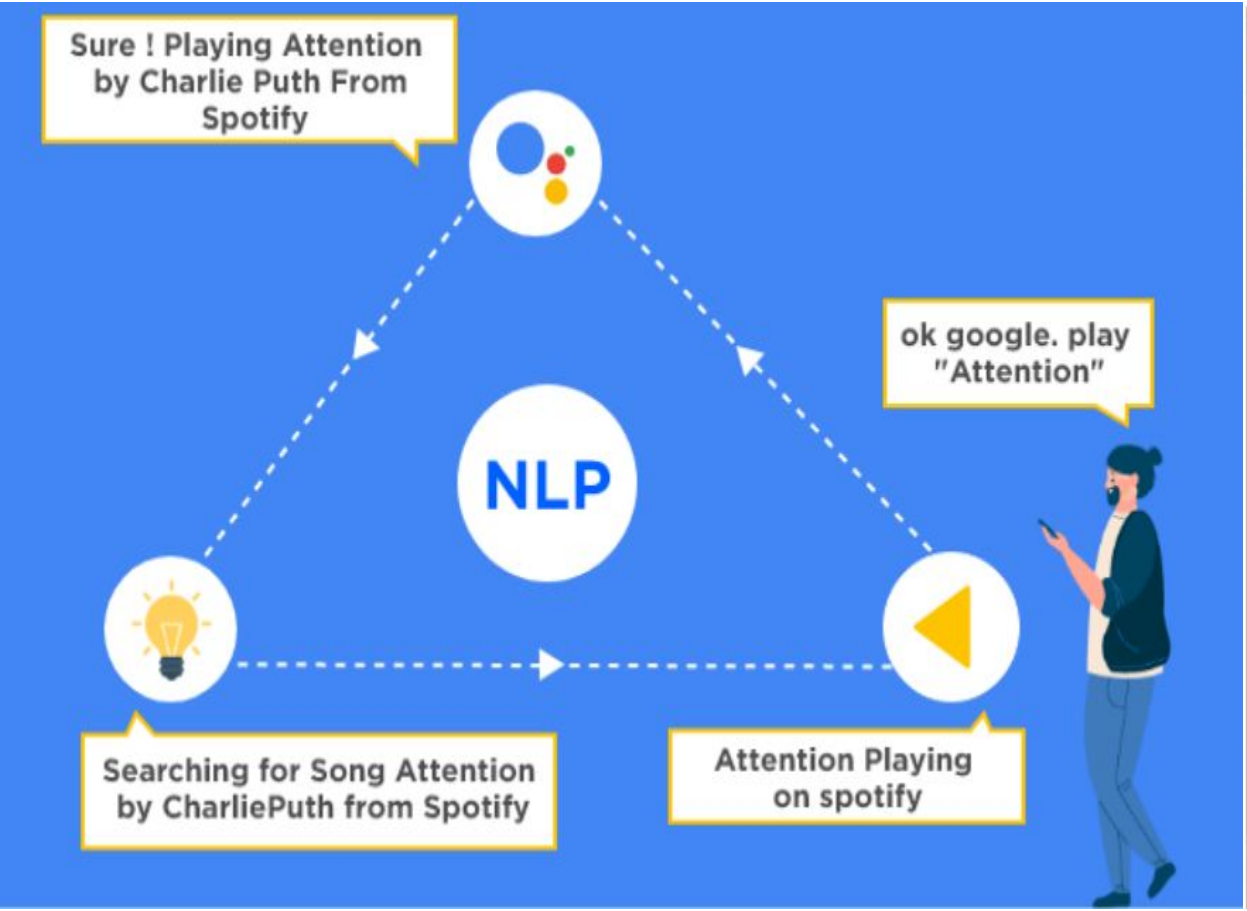
Syed Rameem Zahra
(Assistant Professor)
Department of CSE, NSUT

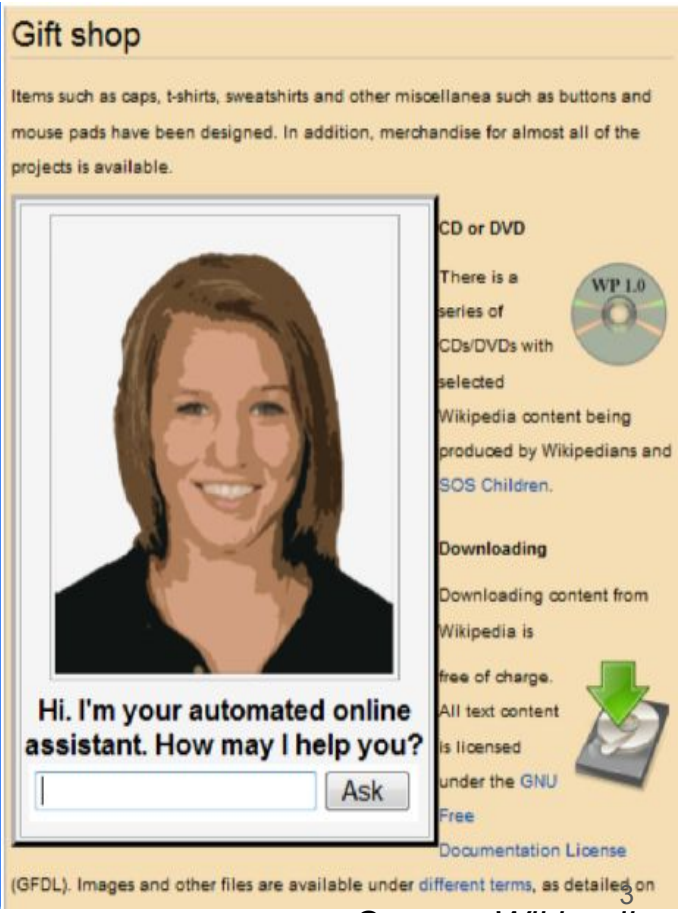
Introduction to NLP

- Natural Language Processing (**NLP**) refers to AI method of communicating with an intelligent systems using a human languages (e.g. English) — speech or text
- NLP-powered software helps us in our daily lives in various ways, for example:
  - **Personal assistants**: Siri, Cortana, and Google Assistant.
  - **Auto-complete**: In search engines (*e.g.* Google).
  - **Spell checking**: Almost everywhere, in your browser, your IDE (*e.g.* Visual Studio), desktop apps (*e.g.* Microsoft Word).
  - **Machine Translation**: Google Translate.



NLP: Real World Examples





Source: Wikipedia

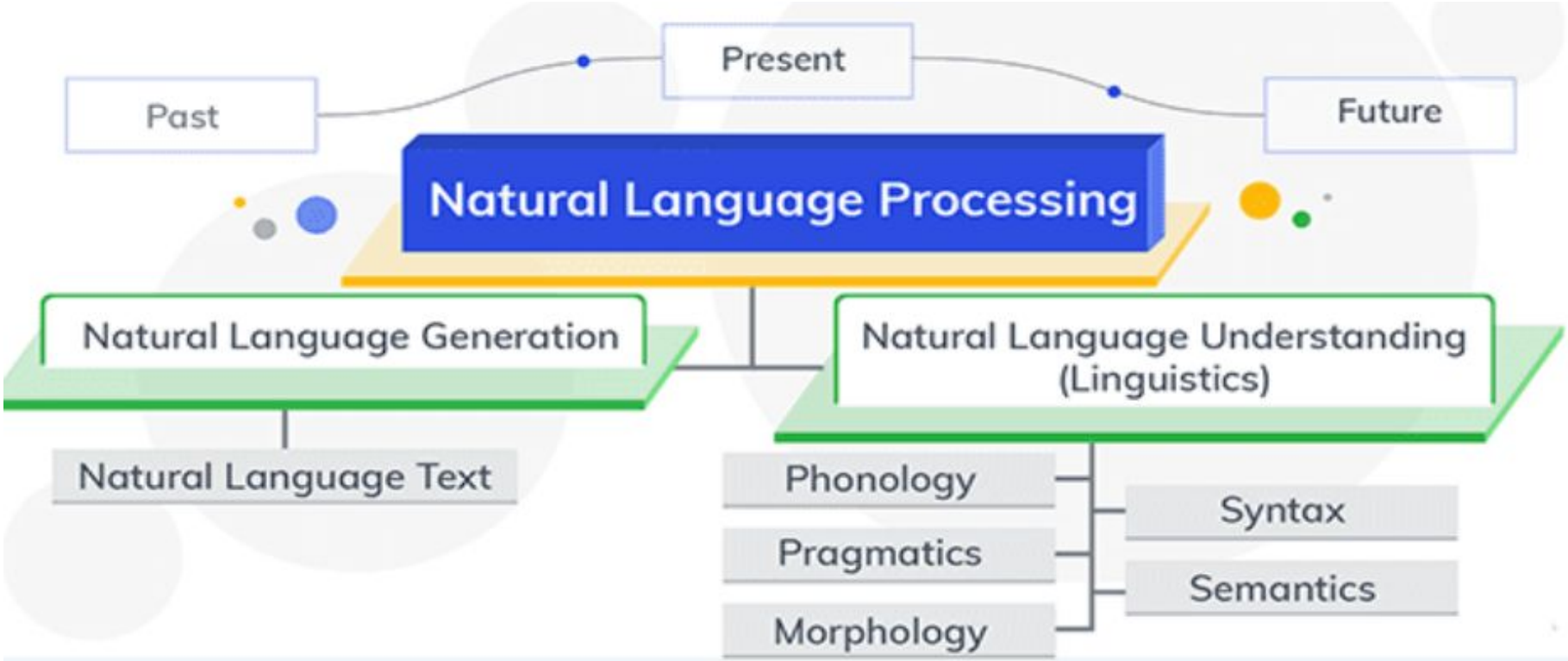
Advantages of NLP

- Computers can infer and analyze human language
- It is the ability of a computer program to understand the human speech.
- Automatic Text Summarization (like in newspapers).
- Finding relationships between sentences.
- Ease in web search.
- Text/speech translation.
- Understanding sentiment in tweets and blogs (Sentiment Analysis).

Applications of NLP

- **Machine Translation** (it is the translation of text or speech by a computer with no human involvement.)
- **Information Retrieval** (software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information.
- **Question Answering** (is concerned with building systems that automatically answer questions posed by humans in a natural language.
- **Dialogue Systems** (computer system intended to converse with a human)
- **Information Extraction** (refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes)
- **Summarization** (refers to the technique of shortening long pieces of text.)
- **Sentiment Analysis** (tries to identify and extract opinions within a given text across blogs, reviews, social media, forums, news etc)

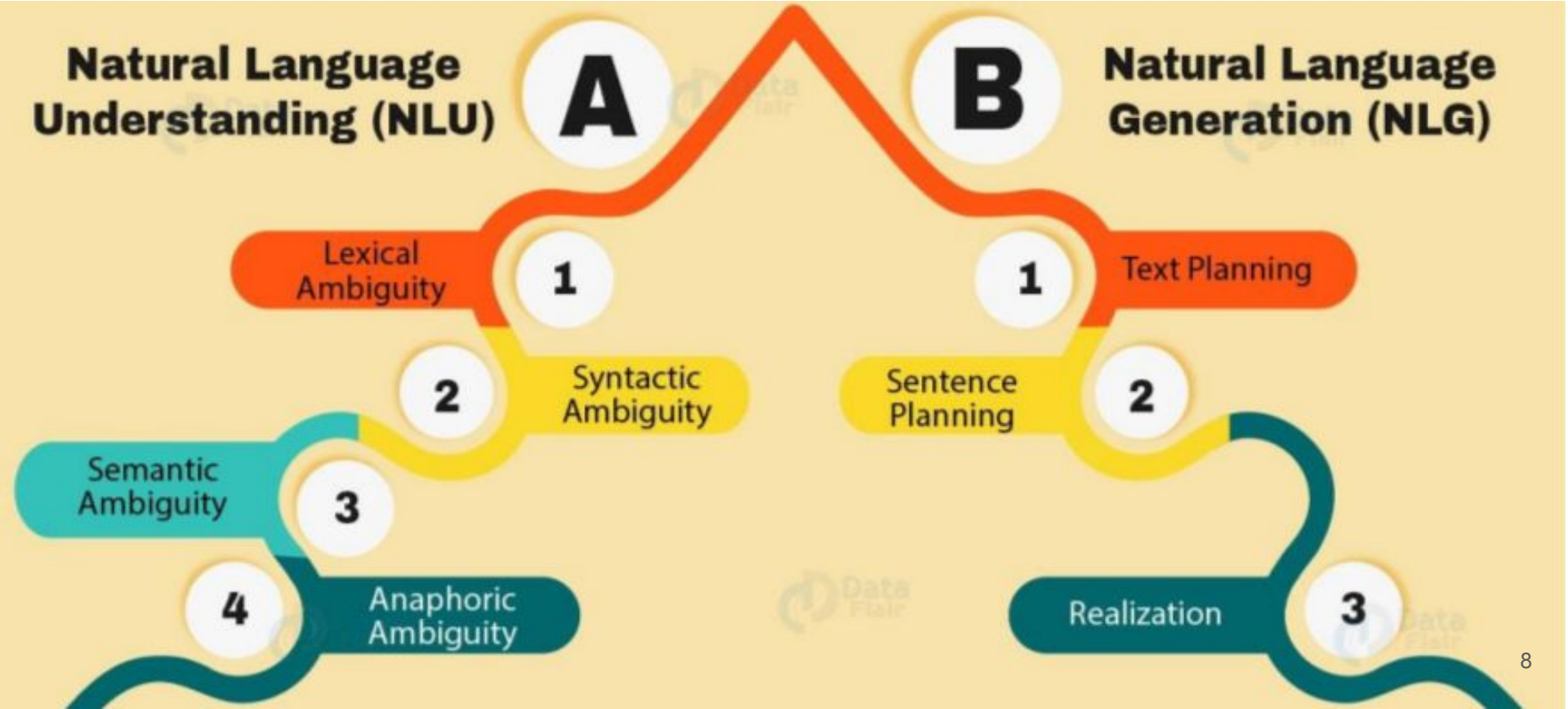
Evolution of NLP



NLP Terminology

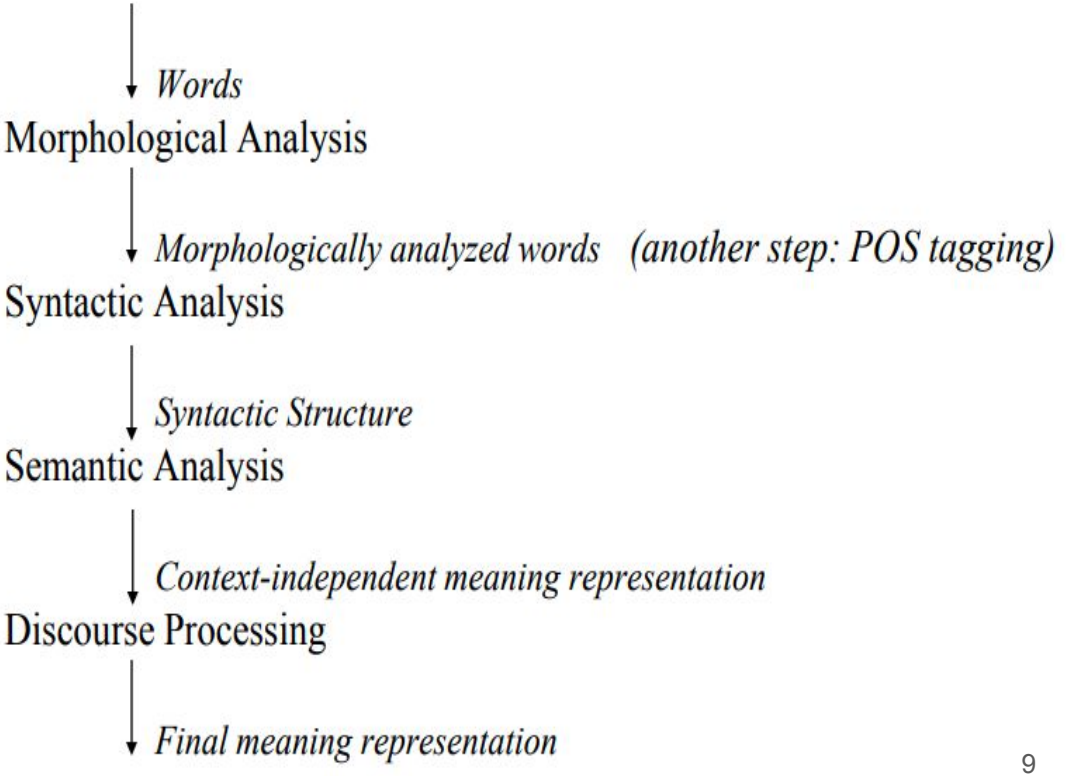
- **Phonology −** It is study of organizing sound systematically.
- **Morphology −** It is a study of construction of words from primitive meaningful units.
- **Morpheme −** It is primitive unit of meaning in a language.
- **Syntax −** It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.
- **Semantics −** It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.
- **Pragmatics −** It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.
- **Discourse −** It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.
- **World Knowledge −** It includes the general knowledge about the world.

Process of NLP



Natural Language Understanding (NLU)

- Mapping the given input in natural language into useful representations.
- Analyzing different aspects of the language.
- The NLU is harder than NLG.



Part-of-Speech (POS) Tagging

- Each word has a part-of-speech tag to describe its category.
- Part-of-speech tag of a word is one of major word groups (or its subgroups).
  - **open classes** -- noun, verb, adjective, adverb
  - **closed classes** -- prepositions, determiners, conjuctions, pronouns, particples
- POS Taggers try to find POS tags for the words.
- duck is a verb or noun? (morphological analyzer cannot make decision).
- A POS tagger may make that decision by looking the surrounding words.
  - Duck! (verb)
  - Duck is delicious for dinner. (noun)

Lexical Processing

- The purpose of lexical processing is to determine meanings of individual words.
- Basic methods is to lookup in a database of meanings -- **lexicon**
- We should also identify non-words such as punctuation marks.
- Word-level ambiguity -- words may have several meanings, and the correct one cannot be chosen based solely on the word itself.
  - bank in English
  - yüz in Turkish
- Solution -- resolve the ambiguity on the spot by POS tagging (if possible) or pass-on the ambiguity to the other levels.

Syntactic Processing

- **Parsing** -- converting a flat input sentence into a hierarchical structure that corresponds to the units of meaning in the sentence.
- There are different parsing formalisms and algorithms.
- Most formalisms have two main components:
  - **grammar** -- a declarative representation describing the syntactic structure of sentences in the language.
  - **parser** -- an algorithm that analyzes the input and outputs its structural representation (its parse) consistent with the grammar specification.
- CFGs are in the center of many of the parsing mechanisms. But they are complemented by some additional features that make the formalism more suitable to handle natural languages.

Semantic Analysis

- Assigning meanings to the structures created by syntactic analysis.
- Mapping words and structures to particular domain objects in way consistent with our knowledge of the world.
- Semantic can play an import role in selecting among competing syntactic analyses and discarding illogical analyses.
  - I robbed the bank -- bank is a river bank or a financial institution
- We have to decide the formalisms which will be used in the meaning representation.

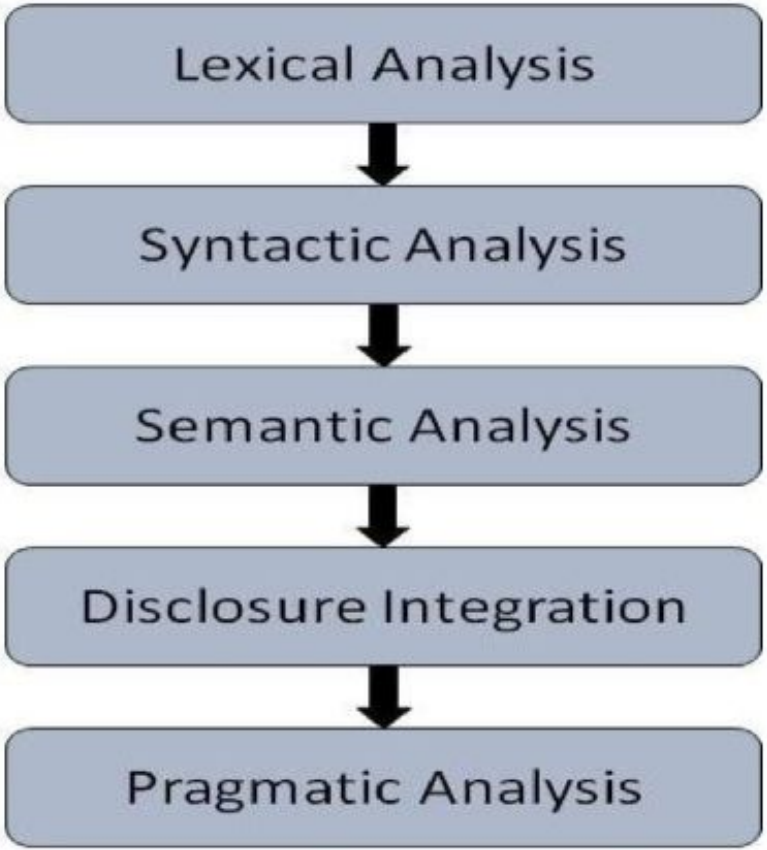
Knowledge Representation for NLP

- Which knowledge representation will be used depends on the application .
  - Requires the choice of representational framework, as well as the specific meaning vocabulary (what are concepts and relationship between these concepts -- ontology)
  - Must be computationally effective.
- Common representational formalisms:
  - first order predicate logic
  - conceptual dependency graphs
  - semantic networks
  - Frame-based representations
  - Vector-space models

Natural Language Generation (NLG)

- It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.
- It involves:
  - **Text planning −** It includes retrieving the relevant content from knowledge base.
  - **Sentence planning −** It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
  - **Text Realization −** It is mapping sentence plan into sentence structure.

Stages of NLP



- **Lexical Analysis:**

➢It involves identifying and analyzing the structure of words.

➢Lexicon of a language means the collection of words and phrases in a language.

➢Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.

- **Syntactic Analysis (Parsing):**

➢It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words.

➢The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.

Stages of NLP (Contd…)

- **Semantic Analysis:**

➢It draws the exact meaning or the dictionary meaning from the text.

➢The text is checked for meaningfulness.

➢It is done by mapping syntactic structures and objects in the task domain.

➢The semantic analyzer disregards sentence such as “hot ice-cream”.

- **Discourse Integration:**

➢The meaning of any sentence depends upon the meaning of the sentence just before it.

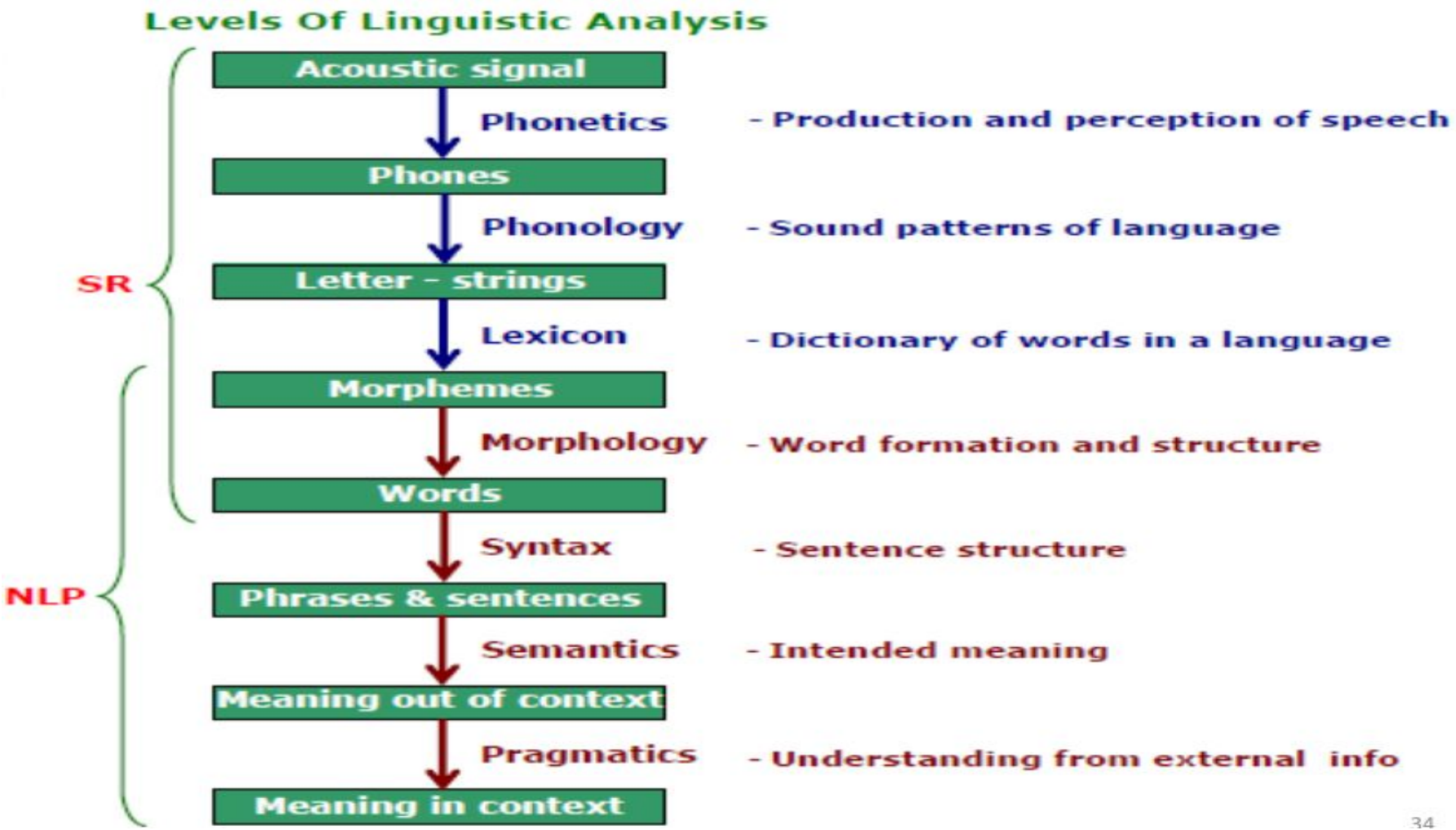
➢In addition, it also brings about the meaning of immediately succeeding sentence.

Stages of NLP (Contd…)

- **Pragmatic Analysis:**

➢During this, what was said is re-interpreted on what it actually meant.

➢It involves deriving those aspects of language which require real world knowledge.


Levels Of Linguistic Analysis
Acoustic signal
Phonetics
- Production and perception of speech
Phones
Phonology
- Sound patterns of language
Letter - strings
Lexicon
- Dictionary of words in a language
Morphemes
Morphology
- Word formation and structure
Words
Syntax
- Sentence structure
Phrases & sentences
Semantics
- Intended meaning
Meaning out of context
Pragmatics
- Understanding from external info
Meaning in context
SR
NLP

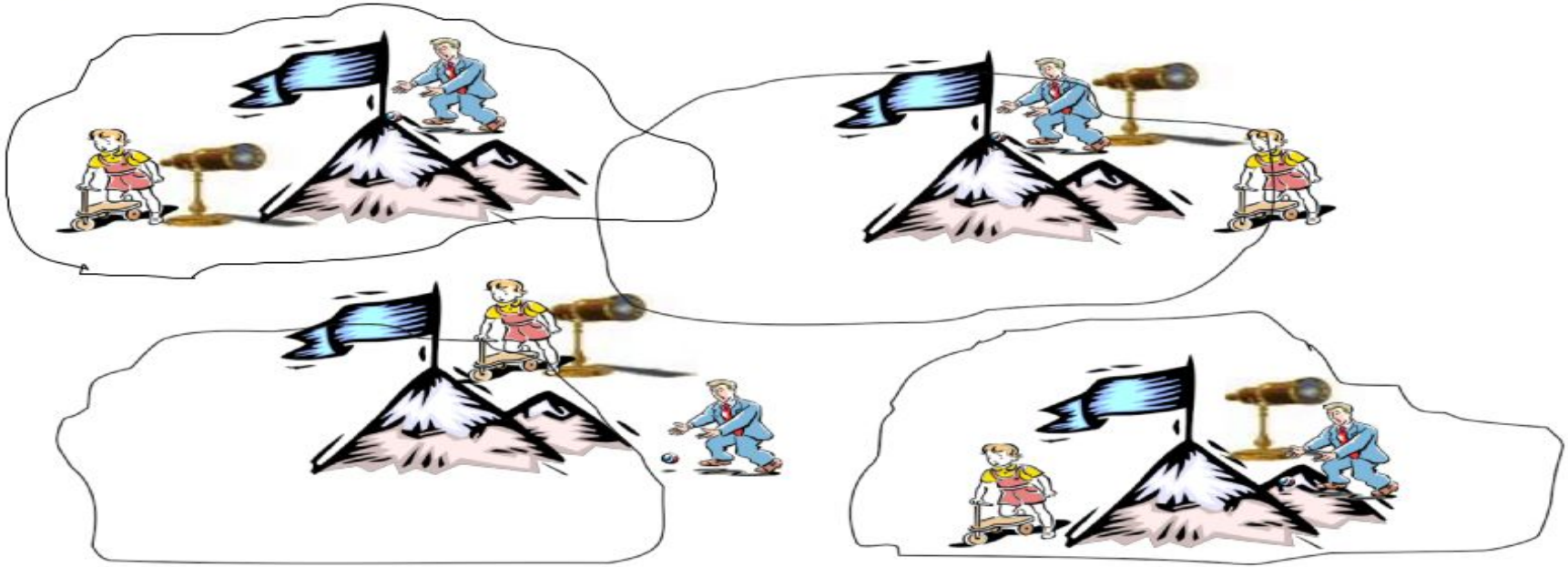
Why NLP is hard: Difficulties in NLU

- **Lexical ambiguity −** It is at very primitive level such as word-level.
  - For example, treating the word “board” as noun or verb?
- **Syntax Level ambiguity −** A sentence can be parsed in different ways.
  - For example, “He lifted the beetle with red cap.” − Did he use cap to lift the beetle or he lifted a beetle that had red cap?
- **Referential ambiguity −** Referring to something using pronouns. For
  - example, Rima went to Gauri. She said, “I am tired.” − Exactly who is tired?
  - One input can mean different meanings.
  - Many inputs can mean the same thing.

An example of Ambiguity

The boy saw the man on the mountain with a telescope

Prepositional phrase attachment



Classical NLP Problems

- *Mostly Solved:*
  - **Text Classification** *(e.g.* spam detection in Gmail).
  - Part of Speech (**POS**) tagging: Given a sentence, determine the POS tag for each word (*e.g.* NOUN, VERB, ADV, ADJ).
  - Named Entity Recognition (**NER**): Given a sentence, determine named entities (*e.g.* person names, locations, organizations).
- *Making a Solid Progress:*
  - **Sentiment Analysis:** Given a sentence, determine it's polarity (*e.g.* positive, negative, neutral), or emotions (*e.g.* happy, sad, surprised, angry, etc)
  - **Co-reference Resolution:** Given a sentence, determine which words ("mentions") refer to the same objects ("entities"). for example (**Manning** is a great NLP professor, **he** worked in the field for over two decades).
  - Word Sense Disambiguation (**WSD**): Many words have more than one meaning; we have to select the meaning which makes the most sense based on the context (*e.g.* I went to the bank to get some money), here bank means a financial institution, not the land beside a river.
  - Machine Translation (*e.*g. Google Translate).
- *Still Challenging:*
  - Dialogue agents and chat-bots, especially open-domain ones.
  - Question Answering.
  - Abstractive Summarization.
  - NLP for low-resource languages (e.g. African languages)

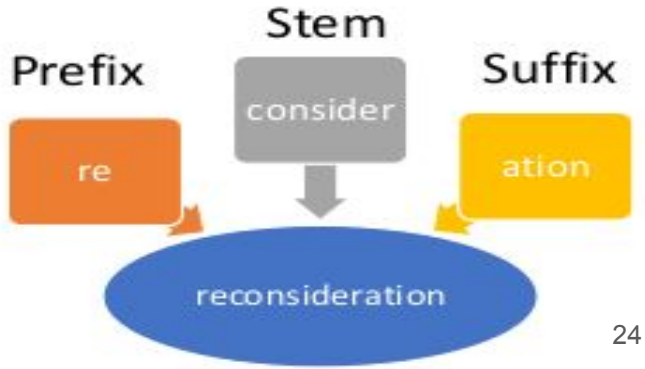
Morphology

- Morphology comes from a Greek word meaning “**Shape**” or “**Form**” and is used in linguistics to denote the **study of words**, both with regard to their internal structure (e.g. ***washing -> wash + ing***) and their combination or formulation to form new or larger units (e.g. ***bat->bats :: rat->rats***)
- Morphology tries to formulate rules.
- It helps in different domains such as spell checkers and machine translation.
- Morphological Analyzer and generator is a tool for analyzing the given word and generator for generating word given the stem and its features (like affixes).
- It identifies how a word is produced through the use of **morphemes**.

Morpheme and its Types

- The **morpheme** is the smallest element of a word that has grammatical function and meaning.
- Types:
  - **Free morpheme**: A single free morpheme can become a complete word, For instance, a bus, a bicycle, and so forth.
  - **Bound morpheme:** It cannot stand alone and must be joined to a free morpheme to produce a word. ing, un, and other bound morphemes are examples.

Root	Morphological variants
walk	walks, walked, walking
noise	noisy, noisily
atom	atomic
order	reorder, orderly
active	hyperactive, proactive
elect	reelect, reelection
view	preview, previewer, previewers



Basic word classes (parts of speech)

- **Content words (open-class):**

– **Nouns:** student, university, knowledge,...

– **Verbs:** write, learn, teach,...

– **Adjectives:** difficult, boring, hard, ...

– **Adverbs:** easily, repeatedly,...

- **Function words (closed-class):**

– **Prepositions:** in, with, under,...

– **Conjunctions:** and, or,...

– **Determiners:** a, the, every,...

Words aren’t just defined by blanks

- **Problem 1:** Compounding
  - “ice cream”, “website”, “web site”, “New York-based”
- **Problem 2:** Other writing systems have no blanks, like Chinese
- **Problem 3:** Clitics
  - English: “doesn’t” , “I’m” ,
  - Italian: “dirglielo” = dir + gli(e) + lo (meaning: tell + him + it)

How many words are there?

"Of course he wants to take the advanced course too. He already took two beginners' courses."

- **How many word tokens are there?**
  - (16 to 19, depending on how we count punctuation)
- **How many word types are there?**
  - i.e. How many different words are there?
  - Again, this depends on how you count, but it's usually much less than the number of tokens
- The same (underlying) word can take different forms: course/courses, take/took
- We distinguish concrete word forms (take, taking) from abstract lemmas or dictionary forms (take)
- Different words may be spelled/pronounced the same: of course vs. advanced course; two vs. too

How many different words are there?

- **Inflection** creates different forms of the same word:
  - Verbs: to be, being, I am, you are, he is, I was,
  - Nouns: one book, two books
- **Derivation** creates different words from the same lemma:
  - grace ⇒ disgrace ⇒ disgraceful ⇒ disgracefully
- **Compounding** combines two words into a new word:
  - cream ⇒ ice cream ⇒ ice cream cone ⇒ ice cream cone bakery
- **Word formation is productive**:
  - New words are subject to all of these processes:
  - Google ⇒ Googler, to google, to ungoogle, to misgoogle, googlification, ungooglification, googlified, Google Maps, Google Maps service,...

Inflectional morphology in English

- **Verbs:**
  - Infinitive/present tense: walk, go
  - 3rd person singular present tense (s-form): walks, goes
  - Simple past: walked, went
  - Past participle (ed-form): walked, gone
  - Present participle (ing-form): walking, going
- **Nouns:**
  - Number: singular (book) vs. plural (books)
  - Plural: books
  - Possessive (~ genitive case): book’s, books
  - Personal pronouns inflect for person, number, gender, case: I saw him; he saw me; you saw her; we saw them; they saw us.

Derivational morphology

- **Nominalization:**
  - V + -ation: computerization
  - V+ -er: killer
  - Adj + -ness: fuzziness
- **Negation:**
  - un-: undo, unseen, …
  - mis-: mistake,...
- **Adjectivization:**
  - V+ -able: doable
  - N + -al: national

Morphemes: stems, affixes

dis-grace-ful-ly

prefix-stem-suffix-suffix

- Many word forms consist of a stem plus a number of affixes (prefixes or suffixes)
  - Infixes are inserted inside the stem.
  - Circumfixes (German gesehen) surround the stem
- Morphemes: the smallest (meaningful/grammatical) parts of words.
  - Stems (grace) are often free morphemes.
  - Free morphemes can occur by themselves as words.
  - Affixes (dis-, -ful, -ly) are usually bound morphemes.
  - Bound morphemes have to combine with others to form words.

Morphemes and morphs

- There are many irregular word forms:
  - **Plural nouns** add -s to singular: book-books,
  - but: box-box**es**, fly-fl**ies**, child-child**ren**
  - **Past tense** verbs add -ed to infinitive: walk-walked,
  - but: like-like**d**, leap-leap**t**
- Morphemes are abstract categories
  - Examples: plural morpheme, past tense morpheme
  - The same morpheme (e.g. for plural nouns) can be realized as different surface forms (morphs): -s/-es/-ren
  - Allomorphs: two different realizations (-s/-es/-ren)of the same underlying morpheme (plural)

Morphological parsing

disgracefully

dis grace ful ly

prefix stem suffix suffix

NEG grace+N +ADJ +ADV

Morphological generation

- Generate possible English words:
  - grace, graceful, gracefully
  - disgrace, disgraceful, disgracefully,
  - ungraceful, ungracefully,
  - undisgraceful, undisgracefully,...
- Don’t generate impossible English words:
  - *gracelyful, *gracefuly, *disungracefully,...

Finite-State Automata and Regular Languages: review

- An **alphabet** $\sum$ is a set of symbols:
  - e.g. $\sum$= {a, b, c}
- A **string ω** is a sequence of symbols, e.g ω=abcb.
  - The empty string ε consists of zero symbols.
- The **Kleene closure** $\sum^*$ (‘sigma star’) is the (infinite) set of all strings that can be formed from $\sum$:
  - $\sum^*$= {ε, a, b, c, aa, ab, ba, aaa, ...}
- A **language L** $\subseteq \sum^*$ over $\sum$ is also a set of strings.
  - Typically we only care about proper subsets of $\sum^*$ ($L \subset \Sigma$).

Automata and languages

- An **automaton** is an abstract model of a computer which reads an input string, and changes its internal state depending on the current input symbol.
- It can either accept or reject the input string.
- Every automaton defines a **language** (the set of strings it accepts).
- Different automata define different language classes:
  - **Finite-state automata** define regular languages
  - **Pushdown automata** define context-free languages
  - **Turing machines** define recursively enumerable languages

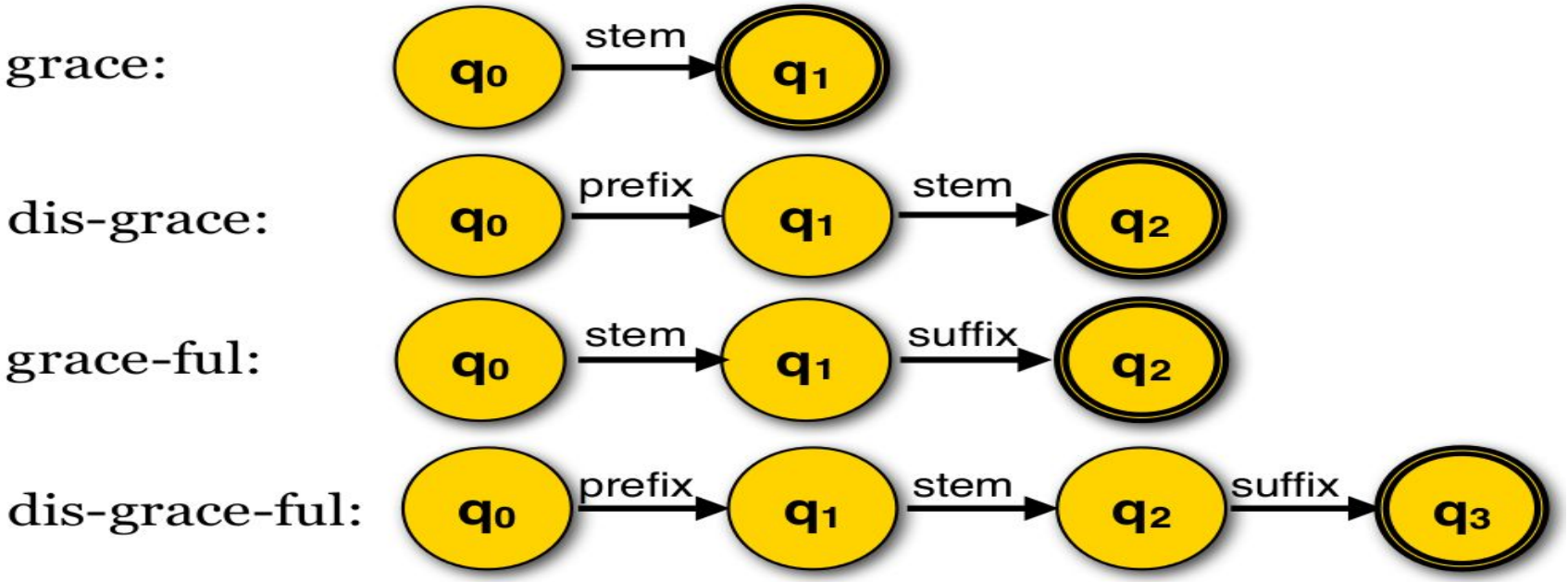
Finite State Automata (FSAs)

- A finite-state automaton M = $\langle Q, \Sigma, qo , F, \delta \rangle$ consists of:
  - A finite set of states **Q** = {qo , q1 ,.., qn}
  - A finite alphabet **Σ** of input symbols (e.g. Σ = {a, b, c,...})
  - A designated start state **qo** $\in Q$
  - A set of final states **F** $\subseteq Q$
  - A transition function **δ**:
    - The transition function for a **deterministic (D)FSA**: $Q \times \Sigma \rightarrow Q$
      - $\delta(q,w) = q'$ for $q, q' \in Q, w \in \Sigma$
      - If the current state is q and the current input is w, go to q'
    - The transition function for a **nondeterministic (N)FSA**: $Q \times \Sigma \rightarrow 2^Q$
      - $\delta(q,w) = Q'$ for $q \in Q, Q' \subseteq Q, w \in \Sigma$
      - If the current state is q and the current input is w, go to any $q' \in Q'$
  - Every **NFA** can be transformed into an equivalent **DFA**

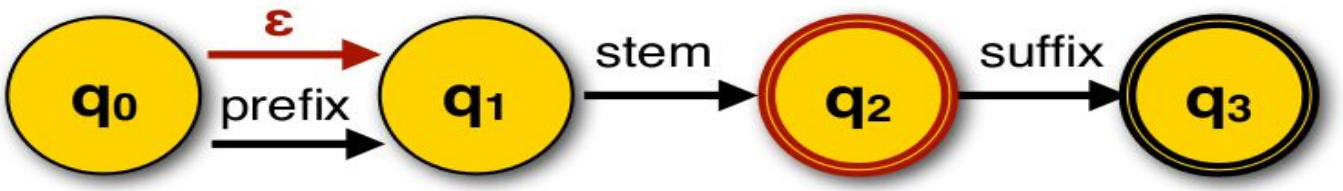
Regular Expressions

- Simple patterns:
  - Standard characters match themselves: ‘a’, ‘1’
  - Character classes: ‘[abc]’, ‘[0-9]’, negation: ‘[^aeiou]’
  - (Predefined: \s (whitespace), \w (alphanumeric), etc.)
  - Any character (except newline) is matched by ‘.’
- Complex patterns: (e.g. ^[A-Z]([a-z])+\s)
  - Group: ‘(...)’
  - Repetition: 0 or more times: ‘*’, 1 or more times: ‘+’
  - Disjunction: ‘...|...’
  - Beginning of line ‘^’ and end of line ‘$’

Finite-state methods for morphology

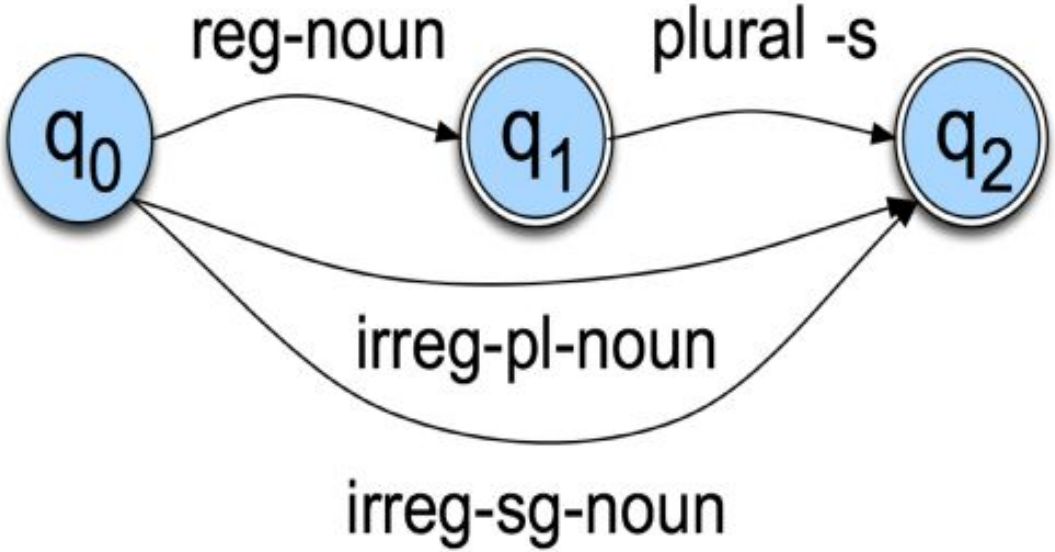


grace,
dis-grace,
grace-ful,
dis-grace-ful

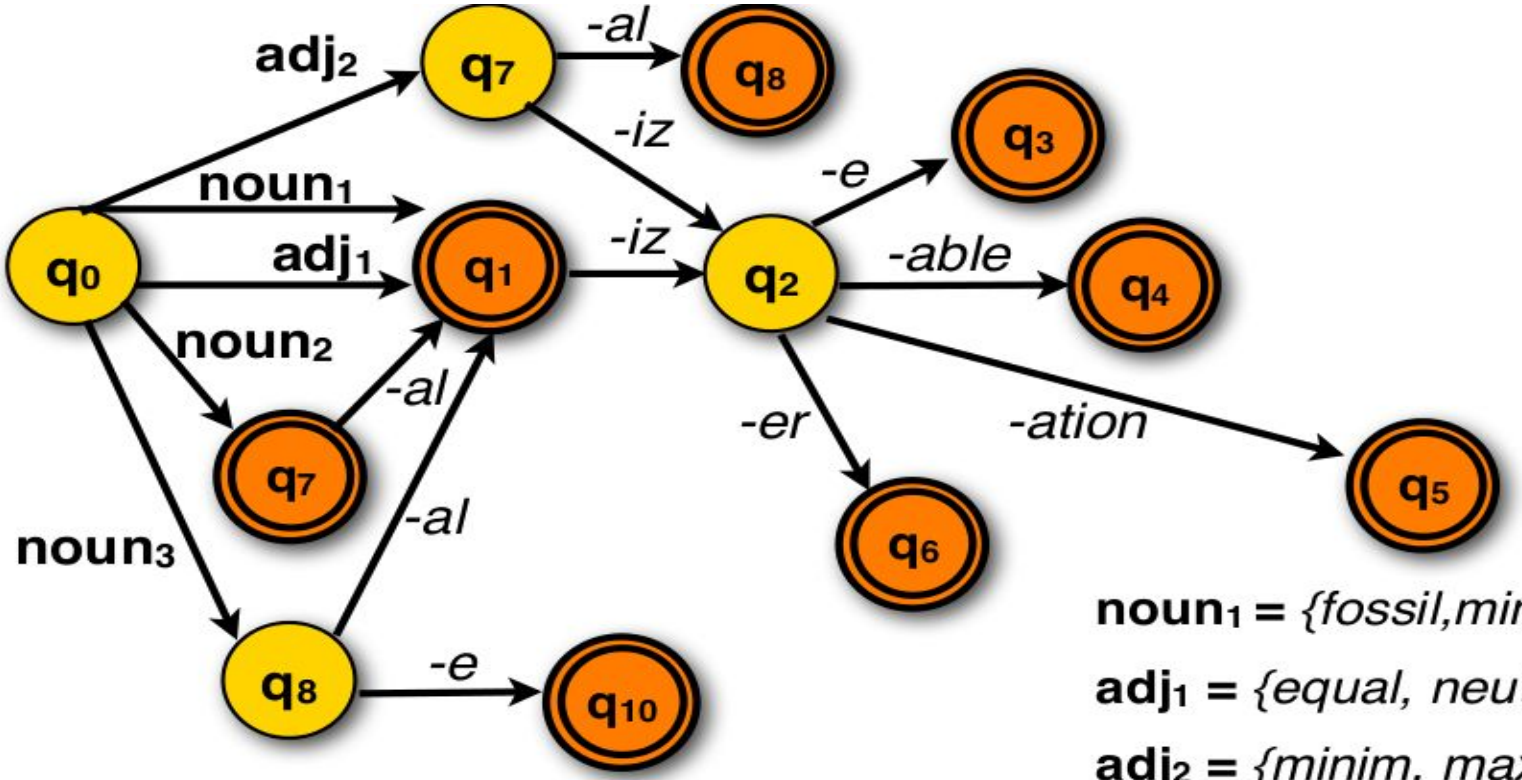


Stem changes

- Some irregular words require stem changes:
  - Past tense verbs: teach-taught, go-went, write-wrote
  - Plural nouns: mouse-mice, foot-feet, wife-wives



FSAs for derivational morphology



noun$_1$ = *{fossil,mineral,...}*

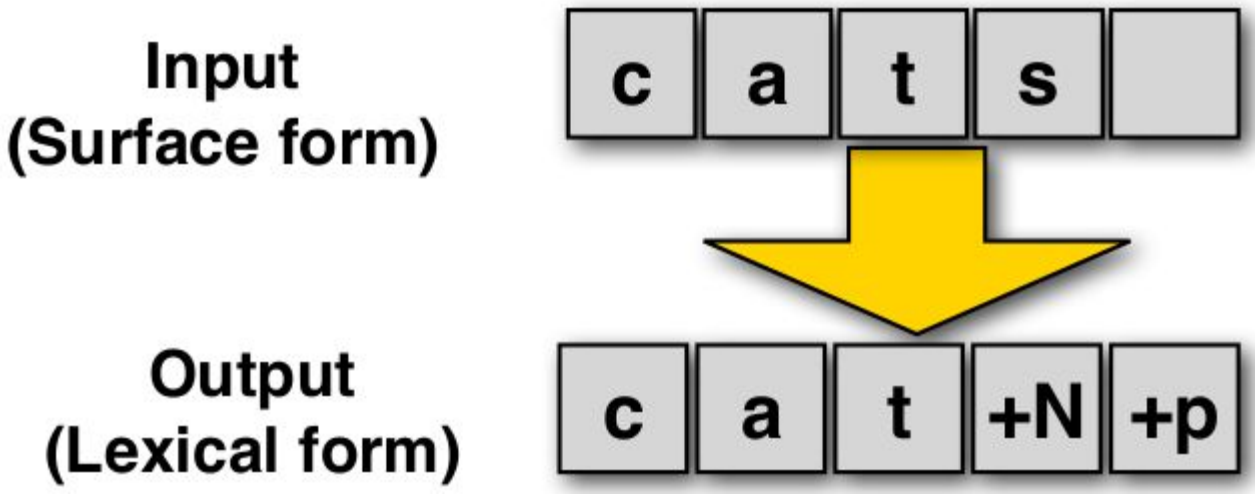
adj$_1$ = *{equal, neutral}*

adj$_2$ = *{minim, maxim}*

noun$_2$ = {nation, form,...}
noun$_3$ = {natur, structur,...}

Recognition vs. Analysis

- FSAs can recognize (accept) a string, but they don't tell us its internal structure.
- We need is a machine that maps (**transduces**) the input string into an output string that encodes its structure:



Finite-state transducers

- A finite-state transducer $T = \langle Q, \Sigma, \Delta, qo , F, \delta, \sigma \rangle$ consists of:
  - A finite set of states **Q** = {qo , q1 ,.., qn }
  - A finite alphabet **Σ** of input symbols (e.g. Σ = {a, b, c,...})
  - A finite alphabet **Δ** of output symbols (e.g. Δ = {+N, +pl,...})
  - A designated start state **qo** $\in Q$
  - A set of final states **F** $\subseteq Q$
  - A transition function **δ**: $Q \times \Sigma \rightarrow 2^Q$
    - $\delta(q,w) = Q'$ for $q \in Q$, $Q' \subseteq Q$, $w \in \Sigma$
  - An output function σ: $Q \times \Sigma \rightarrow \Delta^*$
    - $\sigma(q,w) = \omega$ for $q \in Q$, $w \in \Sigma$, $\omega \in \Delta^*$
    - If the current state is q and the current input is w, write ω.

Finite-state transducers

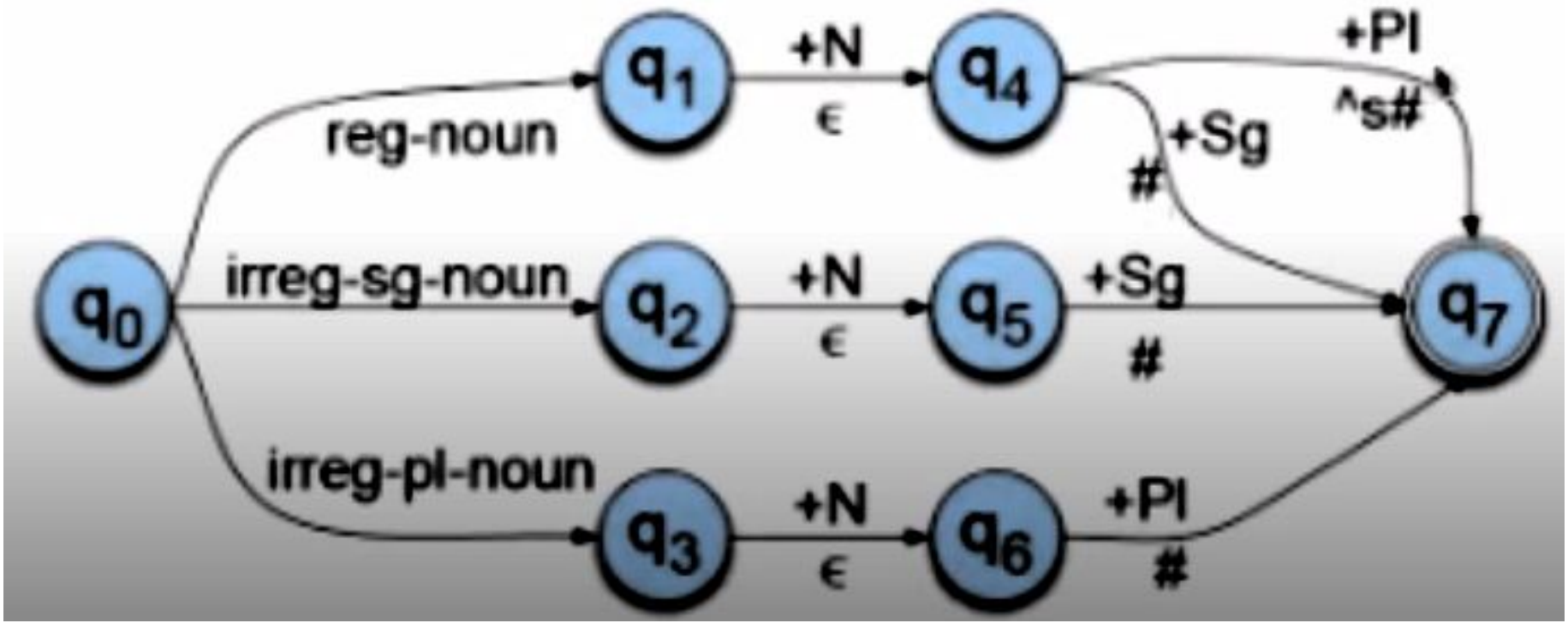
- An FST **T** = $L_{in} \times L_{out}$ defines a relation between two regular languages L_{in} and L_{out} :
  - L_{in} = {cat, cats, fox, foxes, ...}
  - L_{out} = {cat+N+sg, cat+N+pl, fox+N+sg, fox+N+PL ...}
  - **T** = { <cat, cat+N+sg>, <cats, cat+N+pl>, <fox, fox+N+sg>, <foxes, fox+N+pl> }

Note: N: Noun, pl: Plural, sg: Singular

Intermediate representations

- English plural -s: cat ⇒ cat**s** dog ⇒ dog**s**
  - but: fox ⇒ fox**es**, bus ⇒ bus**es** buzz ⇒ buzz**es**
- We define an intermediate representation which captures morpheme boundaries (^) and word boundaries (#):
  - Lexicon: cat+N+PL fox+N+PL
  - ⇒ Intermediate representation: cat^s# fox^s#
  - ⇒ Surface string: cats foxes
- Intermediate-to-Surface Spelling Rule:
  - If plural ‘ **s** ’ follows a morpheme ending in ‘ **x** ’,‘**z**’ or ‘**s**’, insert ‘ **e** ’.

Simplified Morpholgical Parsing FST



Some FST operations

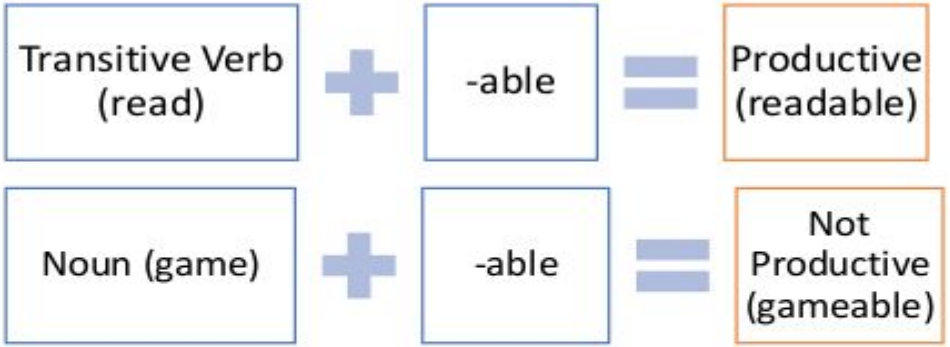
- **Inversion T^{-1} :**
  - The inversion (T^{-1}) of a transducer switches input and output labels.
  - This can be used to switch from parsing words to generating words.
- **Composition (T∘T'): (Cascade)**
  - Two transducers $T = L_1 \times L_2$ and $T' = L_2 \times L_3$ can be composed into a third transducer $T'' = L_1 \times L_3$.

Problems in Morphological Analyzer

- Productivity
- False Analysis
- Bound Base Morphemes

Productivity

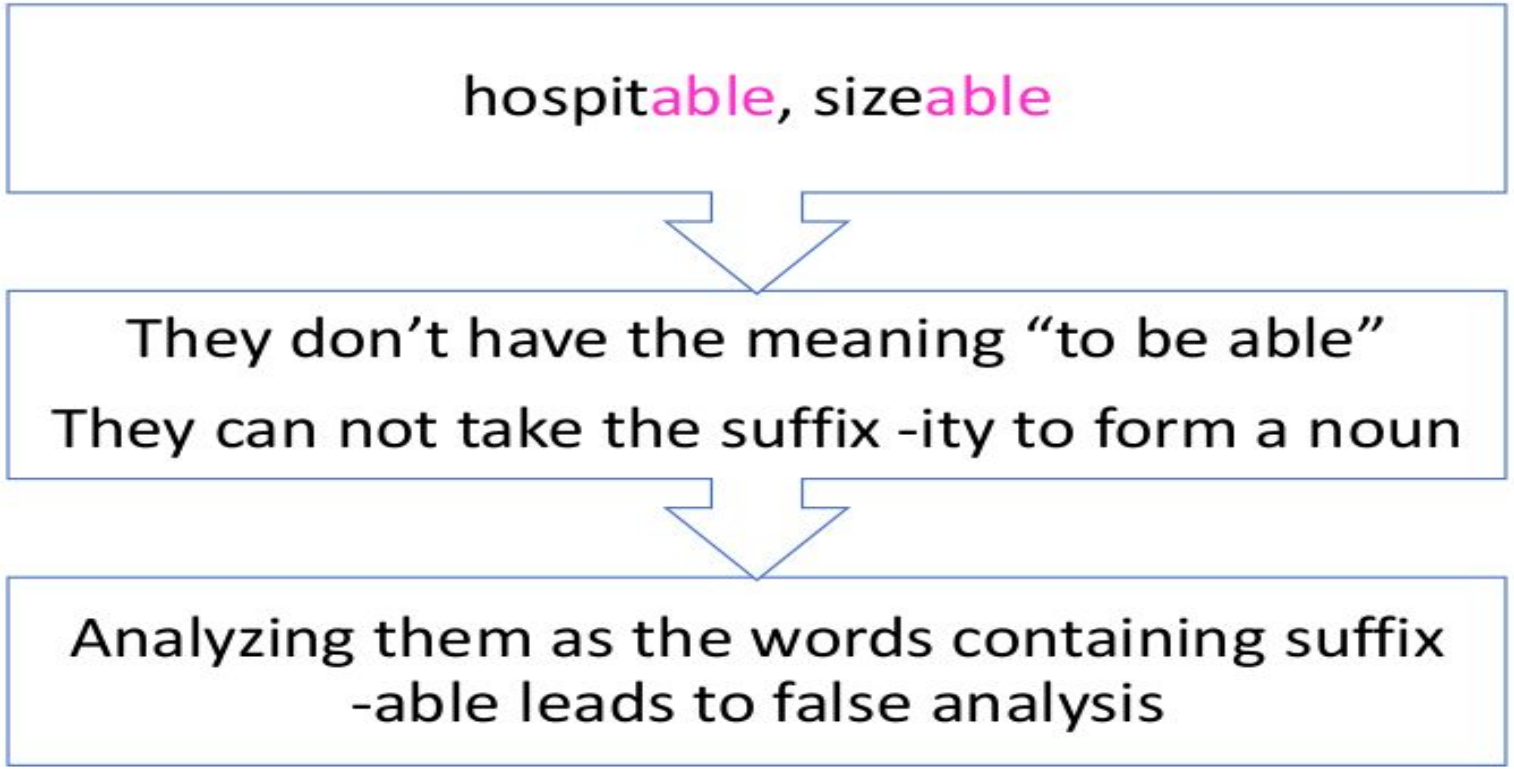
Property of a morphological process to give rise to new formations on a systematic basis



Exceptions

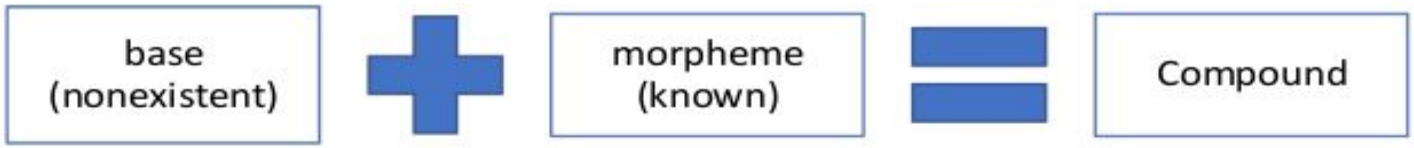
Peaceable	Actionable	Companionable
Saleable	Marriageable	Reasonable
Impressionable	Fashionable	knowledgeable

False analysis



Bound Base Morphemes

- Occur only in a particular complex word
- Do not have independent existence



malleable

feasible
(fease+ible)

- -able has the regular meaning "be able"
- -ity form is possible
- Base words don't exit independently