Natural Language Processing

Paper Code: CS-821/6
UG 8th Sem

Dr. Samit Biswas, *Assistant Professor*

Department of Computer Sc. and Technology

Indian Institute of Engineering Science and Technology, Shibpur

Email: samit@cs.iiests.ac.in

Plan for Today

- Morphology - An Introduction
- Different Approaches

Morphology

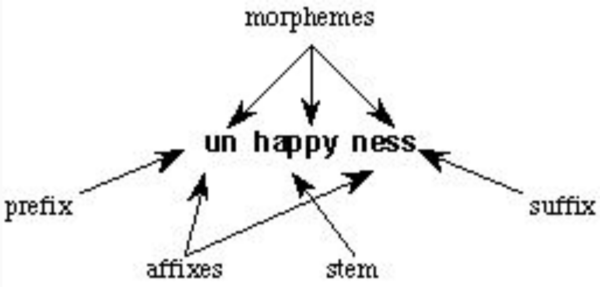
- **What's in a word?**
  - Word processing so far:
    - **Tokenization** - segmenting sentences into words.
    - **Part-of-Speech tagging** - classifying words grammatically.
- **Words have structure:**
  - runs, ran and running are inflected forms of the verb run.
  - unfriendly is derived from friendly, which is derived from friend.
- **Morphological analysis** - exploring the structure of words.
- **Morphology** tries to formulate rules.

Morphology

- **Why does morphology matter?**
  - **Information retrieval:** A query for *phones* should match both *phone* and *phones*.
  - **Language modeling:** If we have seen *scrutinize*, we can predict *scrutinized*.
  - **Machine translation:** English to Bengali.

Morphological Analysis

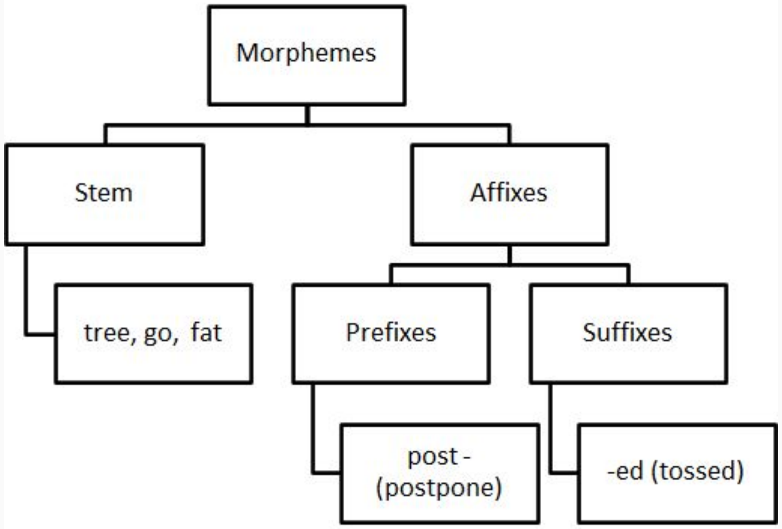
- Morphology is a subdiscipline of linguistics that studies word structure. Analyzing words into their linguistic components (morphemes).
- **“minimal unit of meaning” - “the minimal unit of grammatical analysis”.**
- **Consider a word like: “unhappines”:**



- **There are three Morphemes:**
  un means “not”
  ness means “being in a state or condition”
  Happy is a free morpheme

Morphems

- **Smallest meaning bearing units constituting a word**



Few more examples

Root	Morphological variants
walk	walks, walked, walking
noise	Noisy, noisily
atom	atomic
order	reorder, orderly
active	hyperactive, proactive

Morphology

- **Morphology**
  - More example of morphemes
    - 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, -ly

Morphology

Morphology

- There are many ways to combine morphemes to create words:
  - Inflectional morphology.
  - Derivational morphology.
  - cliticization

Inflectional Morphology

Inflectional morphology

- Inflection relates different forms of the same word

Lemma	Singular	Plural
Cat	Cat	Cats
Dog	Dog	Dogs
Knife	Knife	Knives
Sheep	Sheep	Sheep
Mouse	Mouse	Mice

Note:

- Lemma is the Canonical form found in Dictionaries.
- Affixation sometimes involves spelling changes (Knife - knives)
- Inflection does not always involve affixation (Mouse - Mice)

Derivational Morphology

Derivational Morphology / Word Formation

- Morphological processes can be used to form new words.
- Derivation = stem + affix

  friend + -ly = friendly

  un- + -friendly = unfriendly

  unfriendly + -ness = unfriendliness
- Word composed of more than one free morpheme.

  Compounding = stem + stem

Modifier	**Head**	**Compound**
Noun	Noun	football
Adjective	Noun	blackboard
Preposition	Adverb	without

- usually applies to words of one lexical category and changes them into words of another category. Example: the English derivational suffix -ly changes adjectives into adverbs.

Inflectional vs. Derivational

Inflectional Morphology

- used to show some aspects of the grammatical function of a word.
- We use inflectional morphemes to indicate if a word is singular or plural, whether it is past tense or not, and whether it is a comparative or possessive form.
- **inflectional morphemes** never change the grammatical category

Derivational Morphology

- make words of a different grammatical class from the stem.
- addition of the derivational morpheme -ize changes the adjective normal to the verb normalize.
- **Derivational morphemes** often change the part of speech of a word.

Cliticization

- Combination of a word stem with a **clitic**
- **Clitic:** a morpheme that acts like a word but is reduced and attached to another word
- Example

Full Form	Clitic	Full Form	Clitic
am	‘m	have	‘ve
are	‘re	has	‘s
is	‘s	had	‘d
will	‘ll	would	‘d

- Note: Clitics in English are ambiguous.

Morphological Analysis

Morphological Analysis
Morphological analyzers takes a word in isolation and predict all the possible analyses for that word.

- token → lemma + part of speech + grammatical features
    - Examples
    - cats → cat+N+plur
    - played → play+V+past
- **Morphological Analyzer:**
    - Input: flies
    - Output:
        - Lemma 1 = fly-1 (to move in the air) tag 1 = VBZ (verb, present tense 3rd person singular)
        - Lemma 2 = fly-2 (an insect) tag 2 = NNS (noun, plural)
- Output is not disambiguated with respect to context

Morphological parsing

To build a morphological parser we will need at least the following:

- **Lexicon:** the list of stems and affixes, together with basic information about them (whether a stem is a noun stem or a verb stem , etc.)
- **Morphotactics:** the model of morpheme ordering that explains which classes of morpheme can follow other class of morpheme inside a word.
- **Orthographic rules:** these spelling rules are used to model the changes that occur in a word, usually when when two morphemes combine.

Different Approaches

Different Approaches for Morphological Analysis

Based on:

- Corpus
- Paradigm
- Finite-state automata
- Finite-state transducer

Corpus Based Morphological Analysis

- Corpus (plural corpora) or text corpus is a large and structured set of annotated texts.
- used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory.
- require a large amount of human intervention to annotate the data.

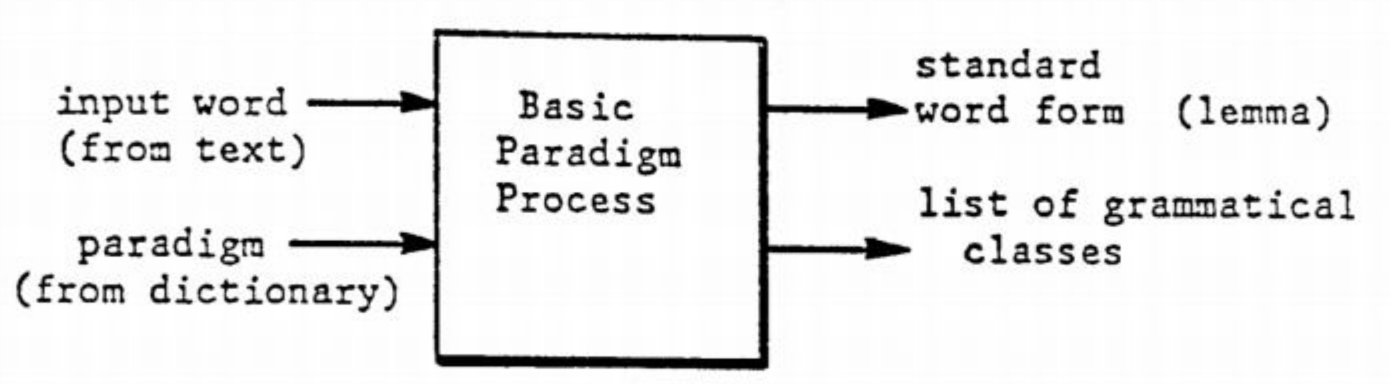
Paradigm based Morphological Analysis

paradigm is the complete set of related word-forms associated with a given lexeme.

- provides all the inflectional forms of a word.
- An input data stream of natural language words can then be processed by generating a lemma for each input word.
- matching the input word against the dictionary and using the resulting paradigm references to access a set of paradigms.

Basic Paradigm Process

- **Basic paradigm Process**



Basic Paradigm Process

- **English Regular Verb Paradigm**
  - Example: Park
    Affixes for
    - Present Participle: -ing
    - Past Participle: -ed
    - Present Tense: -s
    - Past Tense: -ed

- **English Regular Noun Paradigm**
  - Example: book
  - Affixes for
    - singular: -
    - plural: -s

Basic Paradigm Process

- **English Irregular Verb Paradigm**
  - Example: Find
  - Affixes for
    - Present Participle: -ing
    - Past Participle: -ound
    - Present Tense: -s
    - Past Tense: -ed

Morphological Analysis using Finite State Automata (FSA)

Finite State Morphology

- Finite state systems are mathematically well understood.
- Finite state systems are computationally efficient
  (fast and little memory usage)
- Finite state systems provide compact representations for many NLP tasks.
- Finite State systems can be used for
  - Tokenization: divide text into tokens (= words)
  - **Morphological analysis/generation**
  - Part-of-speech tagging: assign a single tag such as VERB or NOUN

FSA Morphology

- **Alphabet:** set of valid symbols
- **Words:** sequence of accepted symbols
- **Language:** set of accepted words
- The description of a finite state acceptor is finite
  - Finite number of states
  - Finite number of alphabet symbols
  - Finite number of transitions
  - Number of accepted strings can be infinite

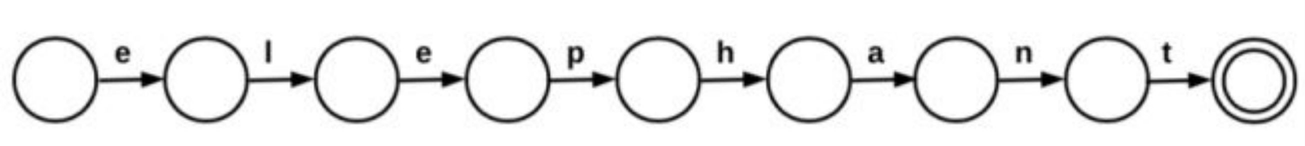
Finite State Automata

- FSAutomata have Input Labels.

Q	a finite set of N states $q_0, q_1, \ldots, q_{N-1}$
Σ	a finite set corresponding to the input alphabet
$q_0 \in Q$	the start state
$F \subseteq Q$	the set of final states
$\delta(q, w)$	$Q \times \Sigma^* \rightarrow 2^Q$

Morphological Analysis using Finite State Automata (FSA)

Example: Small Finite State Acceptor

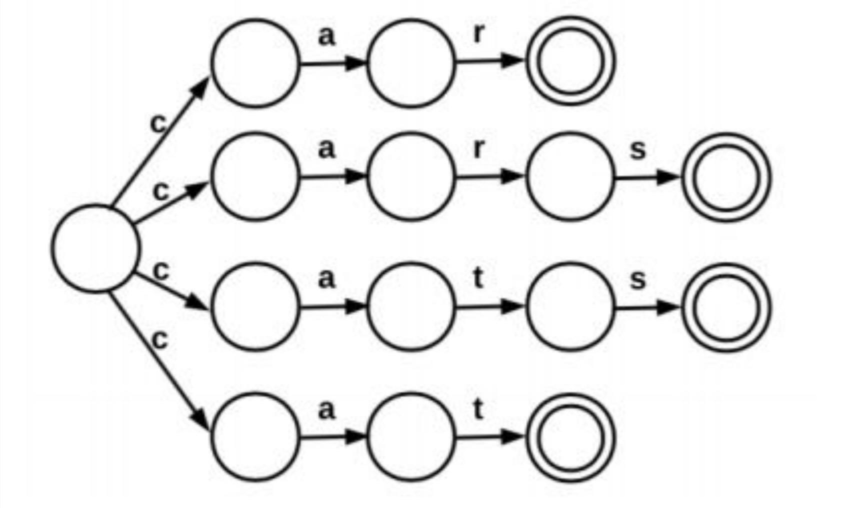


- Network accepts the single word "**elephant**".
- alphabet (set of valid symbols): **e,l,p,h,a,n,t**
- When entering the input sequence **e,l,e,p,h,a,n,t**, the machine transitions through a series of states until the final state and the input word will be accepted.
- No other words (e.g. "elephants" or "ant") are accepted by this network.

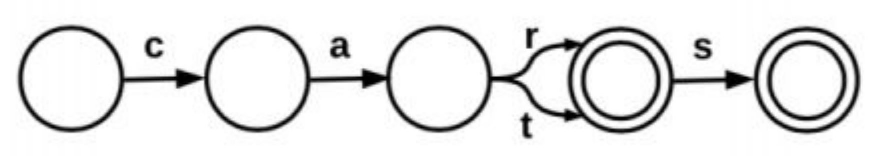
- **IMPORTANT NOTE:** In this case there will always be a single start state (which is the leftmost state on the slide)

Example: Small Finite State Acceptor

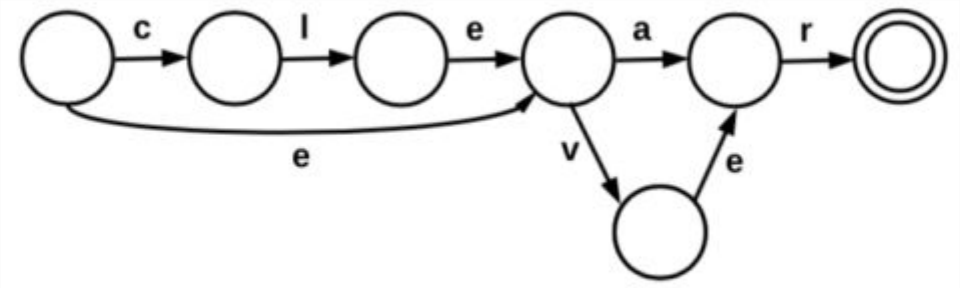
- **Network for the forms “cat”, “cats”, “car”, “cars”**



- States and transitions can be shared

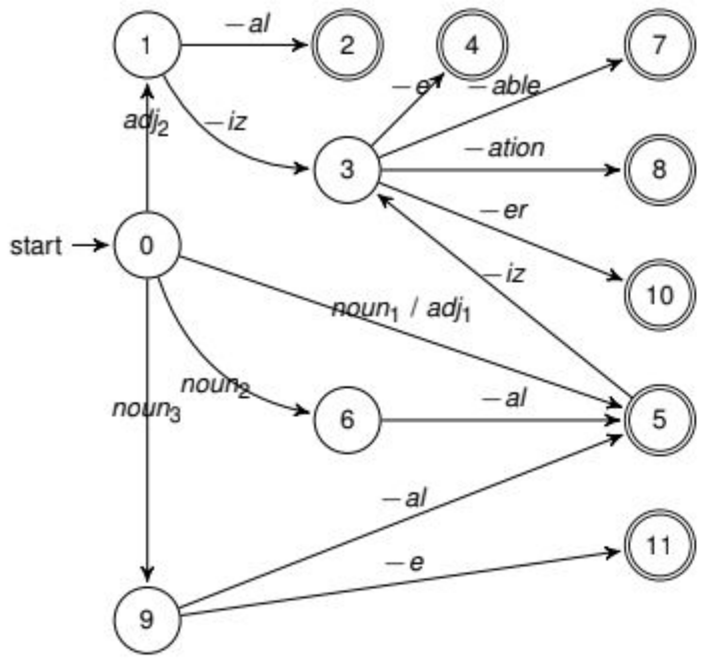


- **Which word forms are recognized by this network?**



- **“clear”, “ear”, “clever”, “ever”**

FSAs for derivational morphology



$$
\begin{aligned}
noun_1 &= \{fossil, mineral, \ldots\},\\
adj_1 &= \{equal, neutral\},\\
adj_2 &= \{minim, maxim\},\\
noun2 &= \{nation, form, \ldots\},\\
noun_3 &= \{natur, structur, \ldots\}
\end{aligned}
$$

References

References

- Diana Maynard, Kalina Bontcheva, Isabelle Augenstein, “*Natural Language Processing for the Semantic Web*”, A Publication in the Morgan & Claypool Publishers series.
- Steven Bird, Ewan Klein and Edward Loper, “*Natural Language Processing with Python*”, By O‘Reilly.
- Christopher Manning and Hinrich Schtze, “*Foundations of Statistical Natural Language Processing*”, The MIT Press.
- Daniel Jurafsky & James H. Martin, “*Speech and Language Processing*”, Prentice Hall

Thank You

Contacts:

Dr. Samit Biswas
Department of CST,
IIEST, Shibpur

samit@cs.iiests.ac.in
https://www.iiests.ac.in/IIEST/Faculty/cs-samit