English Morphology

- Morphological analysis is a field of linguistics that studies the structure of words.
- It identifies how a word is produced through the use of morphemes.
- A morpheme is a basic unit of the English language.
- The morpheme is the smallest element of a word that has grammatical function and meaning.
- Types of Morpheme
- 1. Free Morpheme (Stem)
- 2. Bound Morpheme (affix)

- A single free morpheme can become a complete word.
- For eg: a bus, a bicycle, pen and so forth.
- A bound morpheme, on the other hand, cannot stand alone and must be joined to a free morpheme to produce a word. ing, un etc.
- Affixes are further divided into
- Prefixes- precede the stem-unhappy
- Suffixes- follow the stem- happily
- Infixes- inserted inside stem-passersby

Types of Morphology:

There are two types of Morphology

Inflectional Morphology- It is the combination of a word stem with grammatical morpheme which results in a word of the same class as the original stem.

Eg. Cat□ cats, Mouse□ Mice, walk□ walking

Derivational Morphology- It is the combination of a word stem with grammatical morpheme which results in a word of the different class as the original stem.

Eg. Appoint□ Appointee, clue□ clueless, kill□ killer

Inflectional Morphology

Inflection Morphology conveys grammatical information such as number, tense, comparison, possession.

Due to inflectional morphological process, the meaning and categories of the new inflected words usually do not change.

That is noun can be inflected to a noun while adding affixes, a verb can be inflected to verb in different tense.

Inflection creates different forms of the same word.

Example

Category	Stem	Affixes	Inflected Word
Noun	Word	-s	Words
	Box	-es	Boxes
Verb	Treat	-s	Treats
		-ing	Treating
		-ed	Treated

- Possessive- Boy's, Girl's, Teacher's……
- Tense- Cooked, Played, Waited……
- Comparison- Faster, Slower, Taller……
- Superlative- Fastest, Slowest, Highest..

Derivational Morphology

Derivation is the process of creating new words from a stem/base form of the word.

It Creates new words with new meanings.

The new words are formed through derivational morphology may be a stem for another affix.

Category	Stem	Affixes	Inflected Word	Target Category
Noun	Vapour	-ize	Vaporize	**Verb**
Verb	Read	-er	Reader	**Noun**
Adjectives	Real	-ize	Realize	**Verb**
Noun	Mouth	-ful	Mouthful	**Adjective**

Inflection Vs. Derivation

Inflection	Derivation
It is a morphological process that adopts existing words so that they function efficiently in sentence without changing class of base morpheme.	It is connected with the way morphemes are connected to existing lexical forms as affixes.
More Regular	Less Regular
It can only be use with suffix or infix and not with prefix.	It can use with prefix and suffix.
It never change the grammatical category of base morpheme.	It can change the grammatical category of base morpheme.
Ex. Cat-> Cats Watch-> Watched	Ex. Employee-> Employable Employ->Employee

REGULAR EXPRESSION WITH ITS TYPES

- Regular expression is a sequence of pattern that defines a string. It is used to denote regular languages.
- It is also used to match character combinations in strings. String searching algorithm used this pattern to find the operations on string.
- In regular expression, x* means zero or more occurrence of x. It can generate {e, x, xx, xxx, xxxx,.....}
- In regular expression, x+ means one or more occurrence of x. It can generate {x, xx, xxx, xxxx,.....}

A **Regular Expression** can be recursively defined as follows −

- **ε** is a Regular Expression indicates the language containing an empty string. **(L (ε) = {ε})**
- **φ** is a Regular Expression denoting an empty language. **(L (φ) = { })**
- **x** is a Regular Expression where **L = {x}**
- If **X** is a Regular Expression denoting the language **L(X)** and **Y** is a Regular Expression denoting the language **L(Y)**, then
  - **X + Y** is a Regular Expression corresponding to the language **L(X) ∪ L(Y)** where **L(X+Y) = L(X) ∪ L(Y)**.
  - **X . Y** is a Regular Expression corresponding to the language **L(X) . L(Y)** where **L(X.Y) = L(X) . L(Y)**
  - **R*** is a Regular Expression corresponding to the language **L(R*)**where **L(R*) = (L(R))***

Some RE Examples

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(a+b)*	Set of strings of a's and b's of any length including the null string. So L = { ε, a, b, aa , ab , bb , ba, aaa…….}
(a+b)*abb	Set of strings of a's and b's ending with the string abb. So L = {abb, aabb, babb, aaabb, ababb, ……..}
(11)*	Set consisting of even number of 1's including empty string, So L= {ε, 11, 1111, 111111, …….}