

Parts-of-Speech Tagging



Rajat Kumar Mohanty
rkm at cse dot iitb dot ac dot in

Center for Indian Language Technology
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
<http://www.cfilt.iitb.ac.in>

2nd Asian Regional Training on Local Language Computing 2005
Siem Reap, Cambodia



Introduction



An Ubiquitous Task

- Sequence labeling task can be at different levels.
- In written text
 - *Words*
 - *Phrases*
 - *Sentences*
 - *Paragraphs*



Names for Labeling Tasks

Words: Part-of-Speech tagging / Named Entity tagging / Sense marking

Phrases: Chunking

Sentences: Parsing

Paragraphs: Co-reference annotating

Example (Words: Parts-of-Speech Tagging)

<s>

But they have to attract good senior bankers who can bring in the business from day one.

</s>

<s>

But/CC [they/PRP] have/VBP to/TO attract/VB [good/JJ senior/JJ bankers/NNS] [who/WP] can/MD bring/VB in/RP [the/DT business/NN] from/IN [day/NN one/CD] ./ . </s>



Roadmap

- Basic Idea of Parts-of-Speech Tagging
- How hard is the Problem?
- Linguistic Input
- Issues in POS Tagging
- Tagset Design
- Penn Treebank POS Tags
- Methods used to assign Tags
 - Rule-based
 - Stochastic
- Hindi POS Tagger
 - Morphological Analysis
 - POS Tagging
- Conclusion



Basic Idea of POS Tagging

- To assign an appropriate part of speech to a word in a sentence automatically (*e.g., Noun, Verb, Adjective, Preposition, etc.*)
 - *INPUT: a string of words and a specified Tagset*
 - *OUTPUT: a single best tag for each word*
- The first step in NLP at the sentence level
- Useful for many NLP tasks (*such as parsing, WSD, etc.*)
- Corpora with POS tagging: useful for linguistic research



How hard is the task?

- Is it possible to just look up the words in the dictionary?
 - Consider the English word ‘*can*’
- Is it really hard?
 - Ambiguity
 - Does *that* flight serve dinner.
 - I thought *that* your flight was late.
- Reported accuracy: about 99%



Linguistic Input: Parts of Speech (POS)

- Words are grouped together into classes (sets) which show similar syntactic behavior, often typical semantic type
- Syntactic or Grammatical categories
- Word categories are systematically related by morphological process
- Significant amount of information about the word and its neighbors



Significance of POS

- Information about the word and its neighbors
- Many finer distinctions
 - Possessive pronouns (*my, your, his, her, etc.*)
 - Personal pronouns (*I, you, he, she, etc.*)
- Two broad subcategories
 - Closed class types (relatively fixed membership)
 - Open class types (N, V, Adj, Adv)



Closed Class Types

- Determiners
- Pronouns
- Prepositions
- Conjunctions
- Auxiliaries
- Particles
- Numerals



Some Properties of the Closed Class Types

□ Prepositions

- They occur before NPs
- Semantically relational
- indicate spatial / temporal relations (other relation as well)

□ Conjunctions

- Join two phrases, clauses or sentences
- Join two elements of equal status



Open Class Types

- Nouns (Common nouns, Eventive nouns, etc)
- Verbs (Intransitive, Transitive, Ditransitive, etc)
- Adjectives (attributive, predicative, etc)
- Adverbs (manner, degree, sentential, degree, etc)



Issues in POS Tagging

- ❑ Tagset: The fundamental component
- ❑ Classification of words based on grammatical functions
- ❑ Multiple Tags (*i.e., one morpheme indicates more than one functions*)
- ❑ Multiple Words (*i.e., two words indicate one function*)
- ❑ Unknown Words



Tagset Design

- An appropriate set of Tags has direct influence on the accuracy and the usefulness of the tagging system.
- How much linguistic information expected from the Tagset?
- The larger the *Tagset*, the lower the accuracy
- The smaller the Tagset, the higher the accuracy
- The smallest *Tagset*, the least the usefulness
- Consistency



Some Tagsets

- Range from 40-200
 - London-Lund Corpus : 197 tags
 - Lancaster UCREL : 165 tags
 - LOB Corpus : 135 tags
 - Brown Corpus : 87 tags
 - CLAWS (BNC) : 61 (constituent likelihood automatic word-tagging system)
 - Penn POS Tagset: 48 (originally 87)

Penn Treebank POS Tags

□ 36 + 12

- | | |
|--|--------------------------------------|
| 1. CC Coordinating conjunction | 9. JJS Adjective, superlative |
| 2. CD Cardinal number | 10. LS List item marker |
| 3. DT Determiner | 11. MD Modal |
| 4. EX Existential <i>there</i> | 12. NN Noun, singular or mass |
| 5. FW Foreign word | 13. NNS Noun, plural |
| 6. IN Preposition or
subordinating conjunction | 14. NNP Proper noun, singular |
| 7. JJ Adjective | 15. NNPS Proper noun, plural |
| 8. JJR Adjective, comparative | 16. PDT Predeterminer |

Penn Treebank POS Tags

- | | |
|-------------------------------------|---|
| 17. POS Possessive ending | 28. VBD Verb, past tense |
| 18. PRP Personal pronoun | 29. VBG Verb, gerund or
present participle |
| 19. PRP\$ Possessive pronoun | 30. VBN Verb, past participle |
| 20. RB Adverb | 31. VBP Verb, non-3rd
person singular present |
| 21. RBR Adverb, comparative | 32. VBZ Verb, 3rd person
singular present |
| 22. RBS Adverb, superlative | 33. WDT Wh-determiner |
| 23. RP Particle | 34. WP Wh-pronoun |
| 24. SYM Symbol | 35. WP\$ Possessive wh-pronoun |
| 25. TO <i>to</i> | 36. WRB Wh-adverb |
| 26. UH Interjection | |
| 27. VB Verb, base form | |



Penn Treebank POS Tags

- 37. # Pound sign
- 38. \$ Dollar sign
- 39. . Sentence-final punctuation
- 40. , Comma
- 41. : Colon, semi-colon
- 42. (Left bracket character
- 43.) Right bracket character
- 44. " Straight double quote
- 45. ` Left open single quote
- 46. " Left open double quote
- 47. ' Right close single quote
- 48. " Right close double quote

Penn POS Tags

- Official trading in the shares will start in Paris on Nov 6.

[Official/**JJ** trading/**NN**]

in/**IN**

[the/**DT** shares/**NNS**]

will/**MD** start/**VB** in/**IN**

[Paris/**NNP**]

on/**IN**

[Nov./**NNP** 6/**CD**]

[./.]



Verb forms in English

Base	-S	Pres	Past	Pres_participle	Past_participle
VB	VBZ	VBP	VBD	VBG	VBN
Write	writes	write	wrote	written	writing
Be	is	am, are	was, were	been	being
Do	does	do	did	done	doing
Have	has	have	had	had	having



Methods to Assign Tags



Methods to assign Tags

- Two common approaches
 - Rule-based
 - Stochastic

- Objectives
 - Need to be *fast* in order to process large corpora
 - To assign correct tags without actually parsing the sentence



Rule-Based Tagger

- Generally involves a large database of hand-crafted disambiguation rules (Harris, 1962; Klein and Simmons, 1963; Green and Robin, 1971)
- Based on a two-stage architecture
 - The 1st stage uses a dictionary to assign each word a list of potential POS
 - The 2nd stage uses a large list of hand-crafted disambiguation rules to winnow down this list to a single POS for each word



The ENGTWOL Tagger (Voutilainen, 1995)

- Based on the same two-stage architecture
- More sophisticated than the early algorithm
- Based on Two-level morphology
- 1st stage
 - each word is run through a two-level lexicon transducer
 - the entries for all possible POS are returned
- 2nd stage
 - about 1100 constraints are applied to the input sentences to rule out incorrect POS

Examples

- *The event was not that bad.*
 - 1st stage
 - that **ADV** | PRON | DET | CS
 - 2nd stage
 - that **ADV**
- *John had shown **that** salvation...*
 - 1st stage
 - that ADV | PRON | DET | **CS**
 - 2nd stage
 - that **CS**
- *I consider **that** odd.*
 - 1st stage
 - that ADV | **PRON** | DET | CS
 - 2nd stage
 - that **PRON**

Adverbial-THAT Rule (*informal*)

- Given input “*that*”

if

- the next word is an adjective/adverb #*the event was not **that** bad.*
- and the following is a sentence boundary
- the previous word is not a verb like *believe* or *consider* which allows a small clause

then

- eliminate non-ADV tags

else

- eliminate ADV tag

Complementizer-THAT Rule (*informal*)

- Given input “*that*”

If

- the previous word is a verb which expects a complement (like *think, show, know, etc*)
- “*that*” is followed by the beginning of a noun phrase, and a finite verb

then

- eliminate non-CS tags

else

- eliminate CS tag



Stochastic Taggers

- Produce an often-reasonable output without seeming to know anything about the rules of a language
- Resolve Tagging ambiguities by using a trained corpus to compute the probability of a given word having a given tag in a given context
- Intuition behind all Stochastic taggers
 - *Pick the most likely tag for this word*
- A stochastic tagging algorithm: Hidden Markov Model (HMM)

Example

- Secretariat/NNP is/VBZ expected/VBN to/TO **race/VB** tomorrow/NN
- People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT **race/NN** for/IN outer/JJ space/NN
- Consider the problem of assigning the appropriate tag to *race*
- Choose between NN and VB
 - to/TO race/???
 - The/DT race/???

Example

- Probabilities
 - $P(\text{VB}|\text{TO})P(\text{race}|\text{VB})$
 - $P(\text{NN}|\text{TO})P(\text{race}|\text{NN})$
- *How likely are we to expect a verb or noun given the previous tag?*
- They can just be computed from a corpus by counting and normalizing.
- A verb is more likely to follow *TO* than noun is.
- *To run, to race, to eat*
- Less common: *walk to school*



Brill's Transformation-based Tagger

- Shares features of both tagging architectures
- Based on rules like a Rule-based Tagger
- Has a machine learning component, like a Stochastic Tagger
 - The rules are automatically induced from a previously tagged training corpus
 - Then the words are tagged following these rules
- Works by automatically recognizing and remedying its weaknesses, thereby incrementally improving its performance.

Transformation-based Rules

- Initially, the tagger assigns each word its the most-likely tag
- The most-likely tag for *race* (*in the Brown Corpus*)
 - $P(\text{NN} \mid \text{race}) = .98$
 - $P(\text{VB} \mid \text{race}) = .02$
- Examples
 - ... is/VBZ expected/VBN to/TO **race/NN** tomorrow/NN
 - the/DT **race/NN** for/IN outer/JJ space/NN
- After selecting the most-likely tag, the *transformational rules* are applied
 - *Change NN to VB when the previous tag is TO*



The Lexicon

- The most-likely tag is estimated by examining a large tagged corpus.
- In the Transformation-based tagger, the lexicon is simply a list of all tags seen for a word in the training corpus, with one tag labeled as the most-likely.
- *half*: CD DT JJ NN PDT RB VB

Two Kinds of Rules

□ Contextual Rules

- Revise the tag of a particular word based on the context
- **NN VB PREVTAG TO**
- *Change NN to VB when the previous tag is TO*
- to/TO race/NN → to/TO race/VB

□ Lexical Rules

- Used to tag unknown words or words not found in the lexicon
- **NN s fhassuf NNS**
- *Change NN to NNS if it has suffix –s*
- *E.g., grapes*



Patch Templates

- The tagger acquires patches to improve its performance.
- Patch templates are of the form:
 - If a word is tagged a and it is in context C , then change that tag to b , *or*
 - If a word is tagged a and it has lexical property P , then change that tag to b , *or*
 - If a word is tagged a and a word in region R has lexical property P , then change that tag to b .



The Transformation Templates (non-lexicalized)

- Change tag a to tag b when:
 - (where a , b , z and w are variables over parts-of-speech)
 - The preceding (following) word is tagged z
 - The word two before (after) is tagged z
 - One of the two preceding (following) words is tagged z
 - One of the three preceding (following) words is tagged z
 - The preceding word is tagged z and the following word is tagged w
 - The preceding (following) word is tagged z and the word two before (after) is tagged z

Non-lexicalized Transformations

□ Change tag

From	To	Condition	Example
NN	VB	Previous tag is <i>TO</i>	to race
VBP	VB	One of the previous three tags is MD	might not vanish
NN	VB	One of the previous two tags is MD	might not reply
VB	NN	One of the previous three tags is DT	the most recent match
VBD	VBN	One of the previous three tags is VBZ	...is frequently attacked

The Transformation Templates (lexicalized)

- Change tag a to tag b when:
 - The preceding (following) word is w
 - The word two before (after) is w
 - One of the two preceding (following) words is w
 - The current word is w and the preceding (following) words is x
 - The current word is w and the preceding (following) words is tagged z
 - The preceding (following) word is w and the preceding (following) tag is t
 - The current word is w , the preceding (following) word is w_2 and the preceding (following) tag is t

(where w and x are variables over all words in the training corpus, and a , b , and z are variables over parts-of-speech)



Lexicalized Transformations

- Learned from the trained WSJ Corpus
- Change the tag:
 - From IN to RB if the word two positions to the right is *as*
 - From VBP to VB if one of the previous two words is *n't*
- Examples:
 - *as tall as*
 - *We didn't usually drink*



Transformations for Unknown Words

- Lexical Rules are used to tag unknown words
- *Change the tag of an unknown word (from X) to Y if*
 - Deleting the prefix (suffix) x , $|x| = < 4$, results in word (x is any string of length 1 to 4)
 - The first (last) (1,2,3,4) characters of the word are x
 - Adding the character string x as a prefix (suffix) results in a word ($|x| \leq 4$)
 - Word w ever appears immediately to the left (right) of the word
 - Character z appears in the word

Transformations for Unknown Words

□ Change Tag

From	To	Condition	Example
NN	NNS	Has suffix -s	webpages
NN	CD	Has character . or 0	3.5 or 30
NN	JJ	Has suffix -al	morphological
NN	VBG	Has suffix -ing	tagging
NNS	NN	Has suffix -ss	mass
NN	JJ	Has suffix -ive	constructive
NN	JJ	Has suffix -ble	readable



Transformation-Based Learning (TBL)

- Three major stages (of Brill's TBL algorithm):
 - It labels every word with its most-likely tag.
 - It examines every possible transformation and selects the one that results in the most improved tagging.
 - It then re-tags the data according to the rules.
- TBL is a supervised learning technique; it assumes a pre-tagged training corpus.



Hindi POS Tagger



A Rule-based POS Tagger for Hindi

- Hindi: A morphologically rich Indian language
- A sophisticated Morphological Analyzer is embedded in the POS tagger
- At present 85 Tagsets
- The Tagsets are designed on the basis of morphological information, such as *Person, Number, Gender, Tense, Aspect, Modality, Case, etc*



Morphs and Morphemes

- Consider the word “*accept*”. If “*accept*” is a word, the following questions arise:

Is “*unacceptable*” also a single word?

Is “*accept*” in “*unacceptable*” a word within a word?

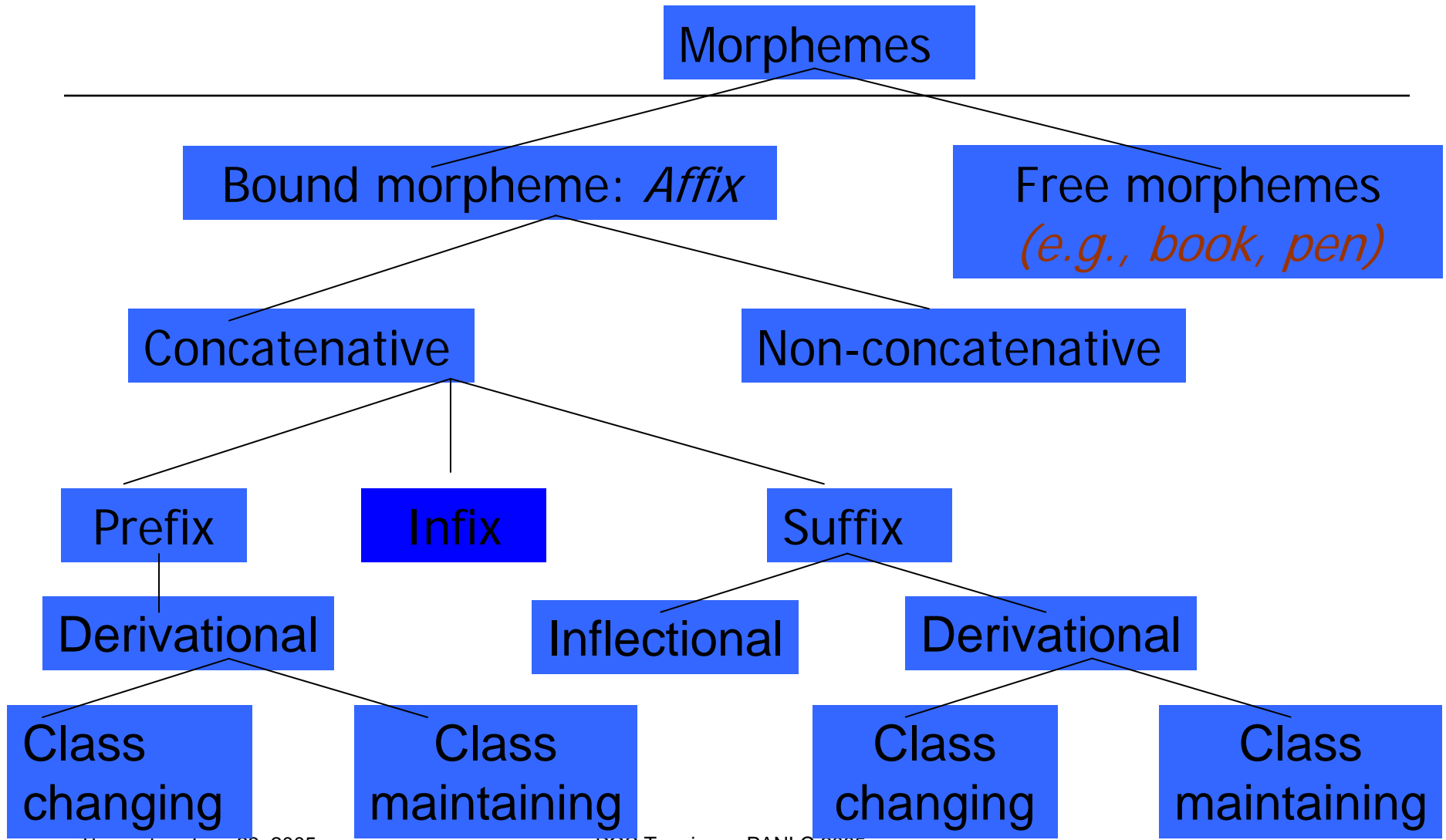
- The word *unacceptable* can be decomposed into three constituents:
 - *un-*, *accept*, *-able*.



Morphs and Morphemes

- The decomposed units of a word
- The minimal recurring units in the grammatical system of a language
- May not always be meaningful
- Examples:
 - Morph: /s, z, ɪz/
 - Morphemes: -s, -es (plural suffixes)

Types of Morphemes



Examples

□ Free Morphemes

- morphemes like “*happy*”, “*regard*”, *etc* can stand on their own as independent words

□ Bound morphemes

- morphemes like *un-*, *dis-*, *etc* cannot stand on their own as independent words

□ Infix

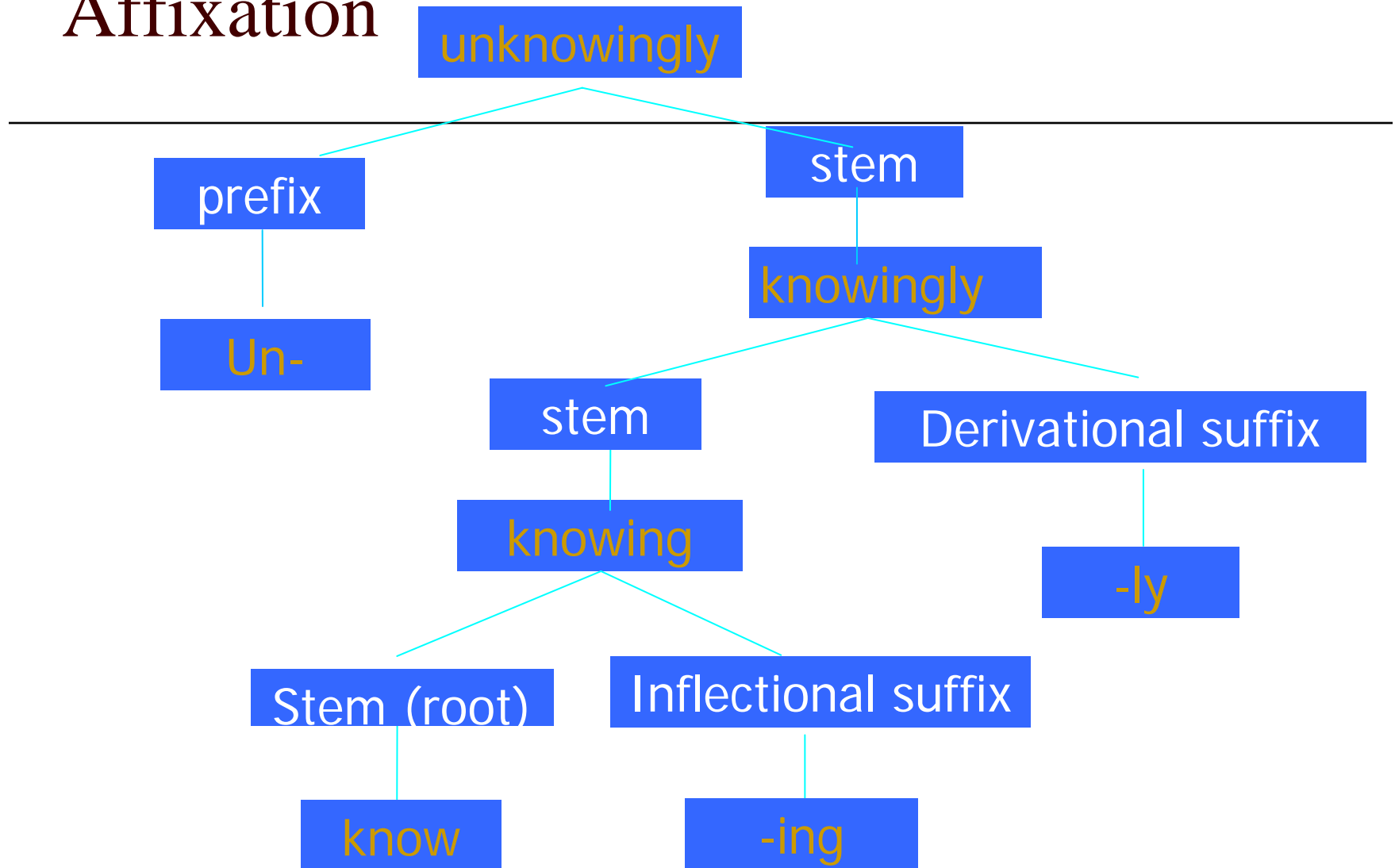
- lakad → lumakad (*walk* → *walked*)
 ↑



Stems and Affixes

- Morphemes like *-s*, *un-*, *dis-*, *etc* are called affixes (*e.g.*, *books*, *undesirable*, *dishonour*)
- The form to which an affix is attached is called a *stem*. In the word “*unhappy*”, the affix is “*un-*” and the stem is “*happy*”.
- Stems may also consist of a number of morphemes.

Affixation



Non-concatenative

- /kʌb/ ‘write’ (no category in Arabic)
- /kaʌb/ ‘to write’
- /kaʌib/ ‘wrote’
- /kaʌʌib/ ‘has written’
- /akʌb/ ‘will write’

Allomorphy

- Phonetically Conditioned /s, z, iz; d, t, id/
(*e.g., books, dogs, houses*)
- Phonologically Conditioned
 - The choice of plural suffix solely depends on the pronunciation of the stem
- Lexically Conditioned
 - -en in *oxen*, -ren in *children*, -im in *seraphim*
- Grammatically Conditioned
 - Adjectives in German change their form depending on the gender of the noun they modify



Morphological Operations

- Inflection
- Derivation
- Compounding
- Feature Percolation



Inflection

- Inflectional morphology has certain characteristics
 - Systematic
 - Adding a particular affix to a stem has the same grammatical or semantic effect for all stems
 - Productive
 - New addition to the language automatically conform to the rules for affixation
 - Preservative
 - The broad grammatical category of the word is not altered in the inflectional process
- *Fix* → *fixed*
- *Laugh* → *laughing*

Derivation

- Derivational Morphology has certain characteristics
 - Unsystematic
 - Adding the same affix two different stems may have radically different semantic effects (e.g., criticise / localise)
 - Partly productive
 - It would be misleading to say that new words in a language automatically undergo derivational process
 - Category alteration
 - The category may or may not be unaltered (e.g., communism, brotherhood)
- Examples

Form → *form* + *al*

Fix → *fix* + *able*

Formal → *formal* + *ise*

Organ → *organ* + *ic*

Derivational Affixes

□ Some examples from English

■ Nouns to nouns

[_{pref} auto [_{stem} biography]]

■ Verb to verbs [_{pref} re [_{stem} try]]

■ Adjective to adjectives [_{pref} sub [_{stem} human]]

■ Nouns to adjectives [_{stem} nation [_{suff} al]]

■ Verbs to nouns [_{stem} work [_{suff} er]]

■ Adjective to adverbs [_{stem} efficient [_{suff} ly]]

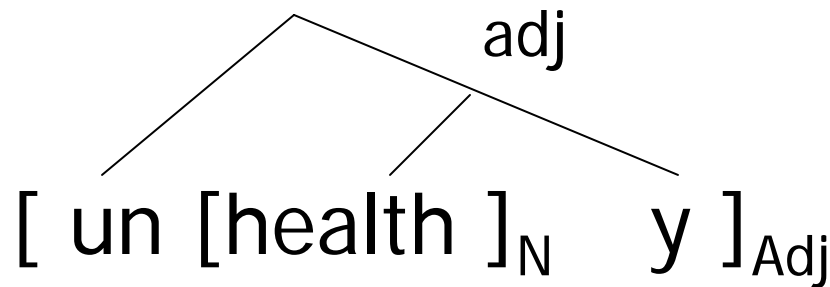


Compounding

- Some examples
 - Houseboat
 - Housewife
 - Blackboard
 - overflow

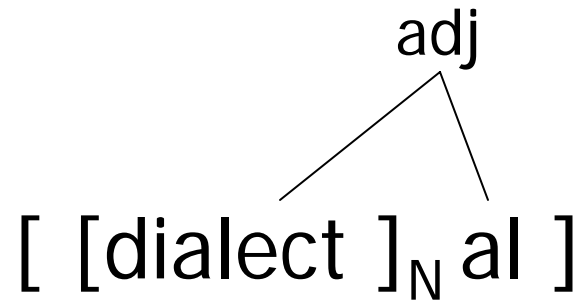
Feature Percolation Convention (FPC)

- Morphology is not a linear ordering of morphemes. Like syntax it has hierarchical organisation.
- The grammatical category of the word comes from the grammatical feature of the rightmost morpheme.



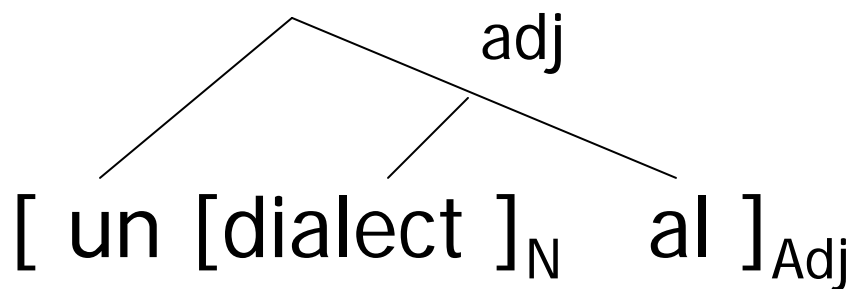
Feature Percolation Convention -1

- If a word is derived by an affix, the grammatical category of the affix is carried over to the derived word. E.g., *dialectal*



Feature Percolation Convention -2

- If a prefix does not have a grammatical feature of its own, then the derived word takes the feature of the immediate lower level. E.g., *undialectal*





Productive Morphology

- Xerox_n
- Xerox_v
- Xeroxable_{adj}
- unxeroxable_{adj}



The Morphology-Syntax Interaction

- The form of a word gets affected by the syntactic construction
- Example: walk {*walks, walk, walked*}
- The selection of a particular form of this verb on a given occasion is dependent on the syntactic construction in which it appears.



Morphology and the Lexicon

- The relation between a word and its meaning is arbitrary
- The role of the lexicon is to list the meanings of words
- Morphosyntactic properties of a word must be stored in the lexicon (*e.g.*, syntactic category, countable features, *etc.*)
- Many morphological problems involve the interaction between morphology and other modules of grammar



Morphological Analysis

- Two basic approaches: *analysis and synthesis*
- The *analytic* approach has to do with breaking words down
- The *synthetic* approach is how to put the pieces together
- In a sense, *Analysis* precedes *synthesis*
- A linguist needs both



Analytic Principles

□ Principle 1

- Forms with the same meaning and the same sound shape in all their occurrences are instances of the same morpheme

- Computerize (-ize)

- Nationalize (-ize)

- Modernize (-ize)

- Singing (-ing)

- Writing (-ing)

- Playing (-ing)



Analytic Principles

□ Principle 2

- Forms with the same meaning but different sound shapes may be instances of the same morpheme if their distributions do not overlap

Seats	/s/
Shades	/z/
Houses	/iz/
Oxen	/en/



Analytic Principles

- Principle 3

- Not all morphemes are segmental

- Example (*ablaut*)

- Run ran
- Speak spoke
- Eat ate



Analytic Principles

□ Principle 4

- A morpheme may have zero as one of its allomorphs provided it has a non-zero allomorph

□ Fish (sg) fish (pl)



Morpheme Recognition

- Meaning of the affixes
- Function of the affixes
- Category of the root / stem
- Restriction of the root / stem
- Range of allomorphs



Morpheme Analysis

□ DOs

- Identify recurring forms
- Match them with recurring meaning
- Remember that a morpheme can have more than one form (allomorphs)

□ DON'T assume that

- The order of the morphemes are same as that of English
- The semantic contrast is same as that of English



Morpheme Analysis: Examples

- Identify recurring forms and match them with recurring meanings. Consider the data from Turkish:

[mumlar] “candles”

[toplara] “gums”

[adamlar] “men”

[kitaplar] “books”



Morpheme Analysis: Examples

- The form /lar/ occurs in all the four items in our sample data.
- From the English gloss of this data, we can see that the feature “plurality” is present in the meaning of all the four items.
- We can hypothesize that /lar/ is the morpheme making “plurality” in Turkish.
- Now we can infer that /mum/ in /mumlar/ is a morpheme which means “*candle*”, and so on.



Morphology: Its Practical Relevance in NLP

- Smaller Dictionary
- Ease of entering data
- Neologism: Even if a word has not been seen before, it can be recognized
- Look-up: Simpler and faster look up process



Computational Issues

□ Segmentation and Grapheme

- Segmentation cannot be done simply by spotting a familiar affix and detaching it (*e.g., thing, read, etc*)

□ Graphotactics

- Deletion (*e.g., love + ed = loved*)
- Insertion (*e.g., church + s = churches*)
- Transformation (*e.g., fly + s = flies*)



Morphological Parsing with *Shoebox*

Lexicon

/ lx	un-	/ lx	success	/ lx	-ful
/ ps	OPPOS-	/ ps	achievement	/ ps	ADJVZR
/ ge	neg	/ ge	n	/ ge	-nadjzr

Parse

/ t	unsuccessful		
/ m	un-	success	-ful
/ ge	OPPOS	achievement	ADJVZR
/ ps	neg	n	-nadjzr

Sample Hindi Morphological Analysis

- /a:/ S-type Aspect | GNR<M> | Causative | deverbal
- /i:/ GNR<F>
- /cuk/ V-type Aspect <PERF>
- /kar/ Habitual Aspect
- /lag/ Inceptive Aspect
- /rah/ Imperfective Aspect
- /h/ Copula (non-past) /t^h/ Copula (past)
- /pɒd/ Modal<deontic>
- /wal/ Nominalizer
- /ẽ/ | /o/ | /ĩ/ Plural number
- /ɛ/ | /ẽ/ | /ũ/ Person Number
- /sɒk/ Modal <probabilitive>
- /jie/ | /ie/ Modal <requestive>
- /g/ Tense<FUT>

Underlying Representation of Hindi VGs

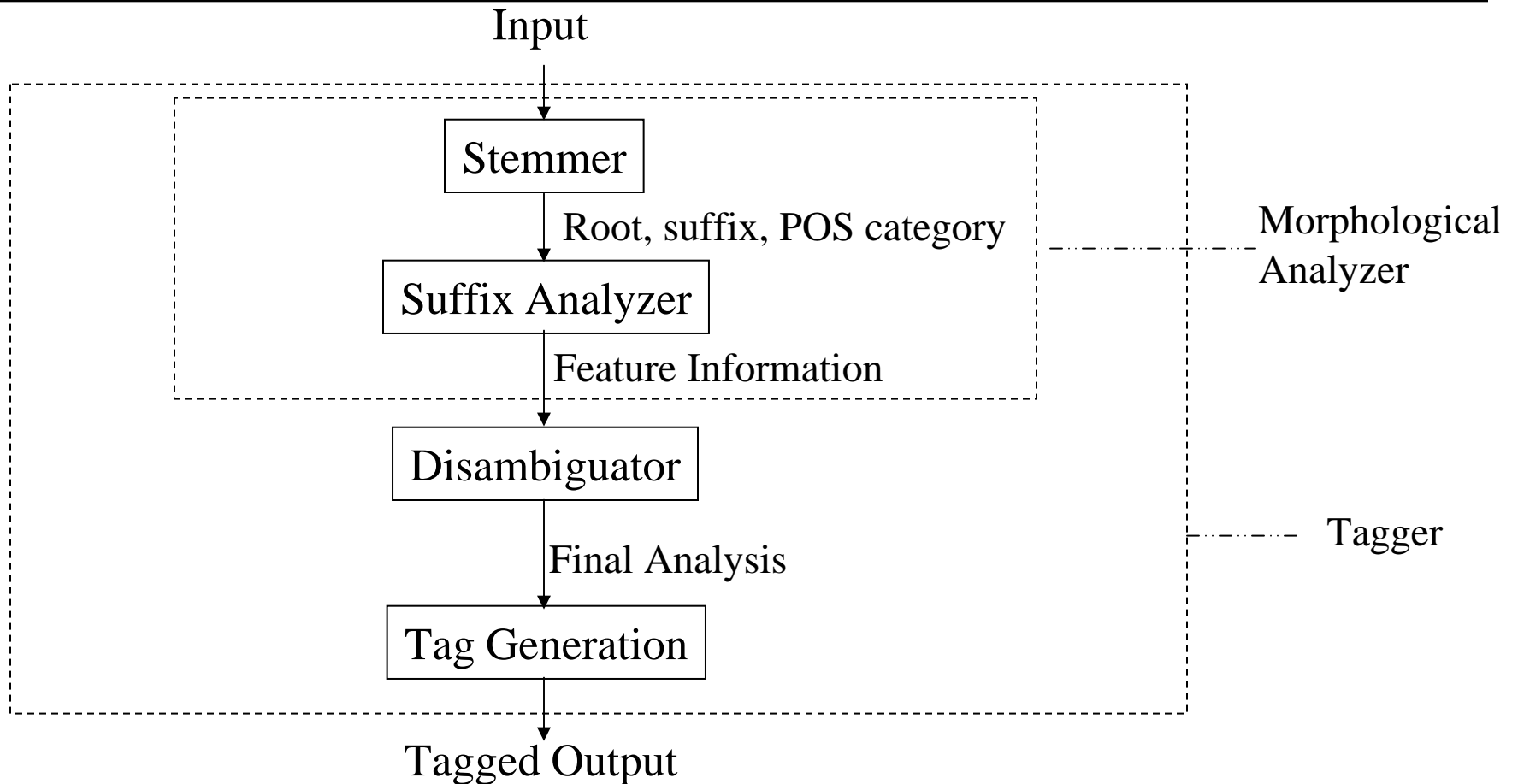
V >MOD >ASP >GNR >ASP >GNR >MOD >ASP >GNR >COP >SJ>PN >F >GNR
b^hag> [] >n >e >[] >[] >[] >lɔg >a: >h >[] >ɛ >[] >[]
b^hag> pa:>[] >[] >rɔh >a: >[] >[] >[] >h >[] >ũ >[] >[]
b^hag> pa:>t >a: >rɔh >a: >[] >[] >[] >h >o > >g >a:
k^ha:> [] >n >i: >[] >[] > pɔd̪>[] >ɽi >h >o > >g >i:



Shoebox

- Demo

Hindi POS Tagger (Shrivastava et.al., 2005)





PNGTAM information in Hindi Verbs

- PNG Features
- Tense: Past, Present, Future
- Aspect
 - Situation type Aspects: *Perfective, Imperfective, etc*
 - View-point type Aspects: *Durative, Inceptive, etc*
- Modality: *deontic, abilitive, permissive, requestive, etc.*

Examples

ghɔrmɛ /N_LOC rɔh̄t̄ahɛ /VB_PRS_3_SG_M
house-LOC stay-M-COP-PRES-3sg

skul /N ja: /VB rɔha:hɛ /PRS_DUR_3_SG_M
school go ASP_{DUR}-M-COP-PRES-3sg

t̄ɔk^hã /N pa:t̄a:hũ /VB_PRS_1_SG_M
money get-M-COP-PRES-1sg

b^ha:g /VB pa:t̄ahũ /PRS_ABL_1_SG_M
run ASP_{ABL}-M-COP-PRES-1sg



Conclusion

- ❑ A Rule-based POS Tagger for Hindi is under development at CFILT, IIT Bombay.
- ❑ Much importance is being given to Morphological Analysis.
- ❑ We started our work with the use of *The Linguist's Shoebox/Toolbox* (<http://www.sil.org/computing/shoebox>) to acquire the necessary morphological information from the corpus.
- ❑ Morphological Information is being used for disambiguation.
- ❑ We are also working on Marathi and Oriya POS Tagger.
- ❑ The same model is likely to be applied to other Indian Languages shortly.



Sources and Suggested Readings

- Allen, J. 2004. *Natural Language Understanding*. Person Education, Singapore.
- Brill, E. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing, Italy*.
- Brill, E. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Parts-of-Speech Tagging. *Computational Linguistics*, 21-4.
- Brill, E. 1997. Unsupervised learning of Disambiguation Rules for Parts-of-Speech Tagging. In *Natural Language Processing Using very Large Corpora*. Kluwer Academic Press.
- Charniak, E. 1993. *Statistical Language Learning*. The MIT Press, Cambridge.



Sources and Suggested Readings

- Harris, Z. 1962. *String Analysis of Language Structure*. Mouton and Co, The Hague.
- Jurafsky, D. and J. H. Martin. 2000. *Speech and Language Processing*. Prentice-Hall, New Jersey.
- Mahapatra, B. B. 2004. *Hindi Verb Morphology*. Ms. CFILT, IITB, INDIA.
- Manning, C. D. and H. Schütze. 2002. *Foundations of Statistical Natural Language Processing*. The MIT Press.
- Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz. 1993. *Building a large annotated corpus in English: The Penn Treebank*. *Computational Linguistics*, 19-2.



Sources and Suggested Readings

- ❑ Shrivastava, M, B. B. Mahaptra, N. Agarwal, S. Sing, P. Bhattacharya. 2005. Morphology-based Natural Language Processing Tools for Indian Languages. Morphology Workshop, CFILT, IIT Bombay, INDIA.
- ❑ Voutilainen, A. 1995. The ENGTWOL Tagger.
<http://www.lingsoft.fi/doc/engcg/intro/>
- ❑ Aronoff, Mark. 2005. *What is Morphology?*. Blackwell. UK.
- ❑ Katamba, Francis. 1993. *Morphology*. Macmillan.
- ❑ Sengupta, Gautam. 1997. Three Models of Morphological Processing. *South Asian Language Review, Vol-vii, 1997*.
- ❑ Nida, Eugene. 1965. *Morphology*.



Thank You