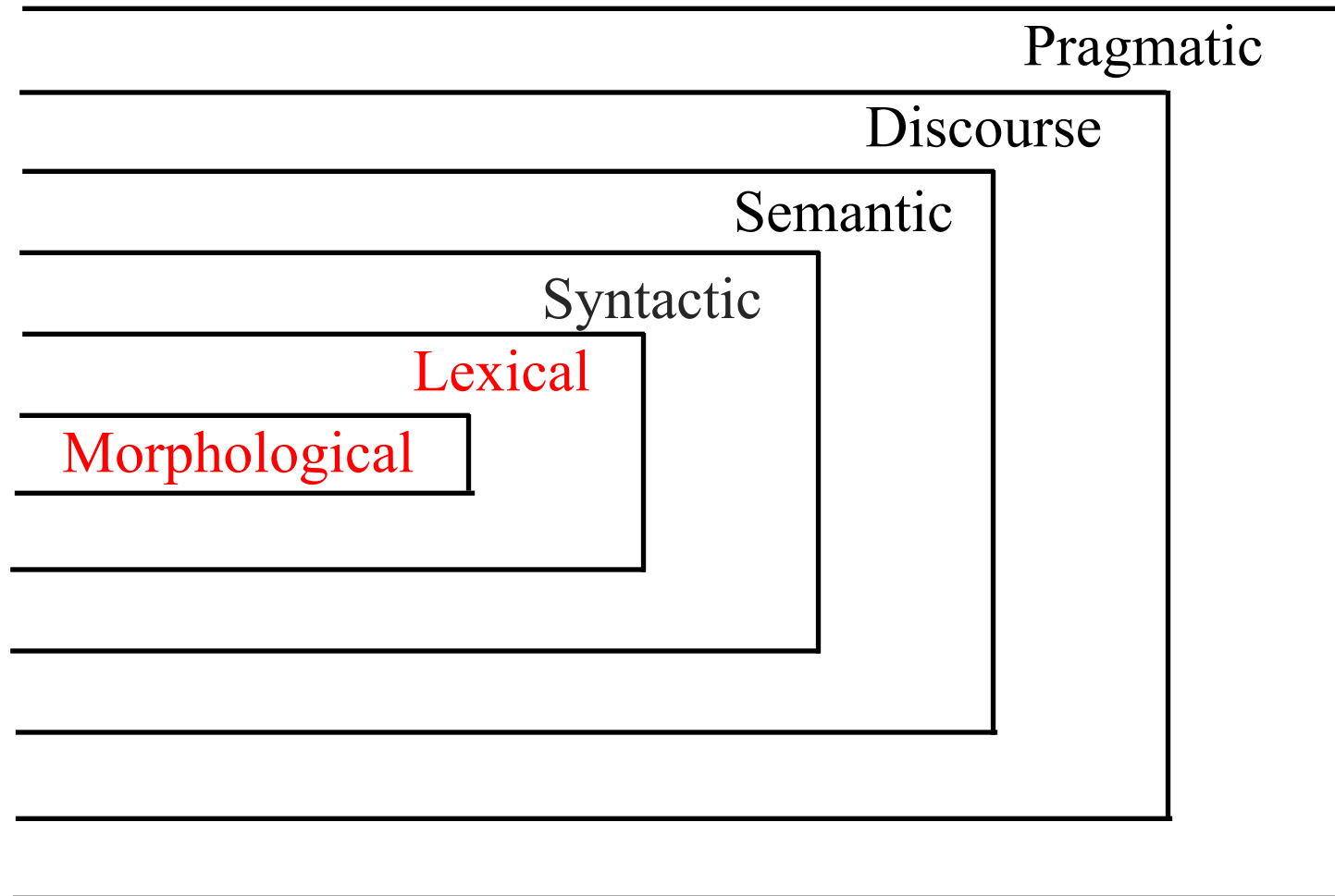

Basic Text Processing:
Morphology
Sentence Segmentation

Basic Text Processing

- Every NLP task needs to do text normalization to determine what are the words of the document:
 - Segmenting/tokenizing words in running text
 - Special characters like hyphen “-” and apostrophe ‘
 - Normalizing word formats
 - (Non) capitalization of words
 - Reducing words to stems or lemmas
 - Segmenting sentences in running text
- To do these tasks, we need to use morphology

Synchronic Model of Language



Morphology

- Morphology is the level of language that deals with the internal structure of words
- General morphological theory applies to all languages as all natural human languages have systematic ways of structuring words (even sign language)
- Must be distinguished from morphology of a specific language
 - English words are structured differently from German words, although both languages are historically related
 - Both are vastly different from Arabic

Minimal Units of Meaning

- **Morpheme** = the minimal unit of meaning in a word
 - *walk*
 - *-ed*
- **Simple words** cannot be broken down into smaller units of meaning
 - Monomorphemes
 - Called base words, roots or **stems**
- **Affixes** are attached to free or bound forms
 - prefixes, infixes, suffixes, circumfixes

Affixes

- **Prefixes** appear in front of the stem to which they attach
 - **un-** + happy = *unhappy*
- **Infixes** appear inside the stem to which they attach
 - **-blooming-** + absolutely = *absobloominglutely*
- **Suffixes** appear at the end of the stem to which they attach
 - *emotion* = emote + -ion
 - English may stack up to 4 or 5 suffixes to a word
 - Agglutinative languages like Turkish may have up to 10
- **Circumfixes** appear at both the beginning and end of stem
 - German past participle of *sagen* is *gesagt*: ge- + sag + -t
- Spelling and sound changes often occur at the boundary
 - Very important for NLP

Inflection

- Inflection modifies a word's form in order to mark the grammatical subclass to which it belongs
 - *apple* (singular) > *apples* (plural)
- Inflection does not change the grammatical category (part of speech)
 - *apple* – noun; *apples* – still a noun
- Inflection does not change the overall meaning
 - both *apple* and *apples* refer to the fruit

Derivation

- Derivation creates a new word by changing the category and/or meaning of the base to which it applies
- Derivation can change the grammatical category (part of speech)
 - sing (verb) > singer (noun)
- Derivation can change the meaning
 - act of singing > one who sings
- Derivation is often limited to a certain group of words
 - You can **Clintonize** the government, but you can't **Bushize** the government
 - This restriction is partially phonological

Inflection & Derivation: Order

- **Order is important** when it comes to inflections and derivations
 - **Derivational suffixes must precede inflectional suffixes**
 - sing + -er + -s is ok
 - sing + -s + -er is not
 - This order may be used as a clue when working with natural language text

Inflection & Derivation in English

- English has few inflections
 - Many other languages use inflections to indicate the role of a word in the sentence
 - Use of case endings allows fairly free word order
 - English instead has a fixed word order
 - Position in the sentence indicates the role of a word, so case endings are not necessary
 - This was not always true; Old English had many inflections
- English has many derivational affixes, and they are regularly used to form new words
 - Part of this is cultural -- English speakers readily accept newly introduced terms
- Look at examples from J&M, sections 3.1 – 3.3 (2nd ed.)

Classes of Words

- **Closed** classes are fixed – new words cannot be added
 - Pronouns, prepositions, comparatives, conjunctions, determiners (articles and demonstratives)
 - Function words
- **Open** classes are not fixed – new words can be added
 - Nouns, Verbs, Adjectives, Adverbs
 - Content words
 - New content words are a constant issue for NLP

Creation of New Words

- **Derivation** - adding prefixes or suffixes to form a new word
 - Clinton → Clintonize
- **Compounding** - combining two existing words
 - home + page → homepage
- **Clipping** - shortening a polysyllabic word
 - Internet → net
- **Acronyms** - take initial sounds or letters to form new word
 - Scuba → Self Contained Underwater Breathing Apparatus
- **Blending** - combine parts of two words
 - motor + hotel → motel
 - smoke + fog → smog
- **Backformation**
 - resurrection → resurrect

Word Formation Rules: Agreement

- Plurals
 - In English, the morpheme *s* is often used to indicate plurals in nouns
 - Nouns and verbs must agree in plurality
- Gender – nouns, adjectives and sometimes verbs in many languages are marked for gender
 - 2 genders (masculine and feminine) in Romance languages like French, Spanish, Italian
 - 3 genders (masc, fem, and neuter) in Germanic and Slavic languages
 - More are called noun classes – Bantu has up to 20 genders
 - Gender is sometimes explicitly marked on the word as a morpheme, but sometimes is just a property of the word

How does NLP make use of morphology?

- Stemming
 - Strip prefixes and / or suffixes to find the base root, which may or may not be an actual word
 - Spelling corrections are not made
- Lemmatization
 - Strip prefixes and / or suffixes to find the base root, which will always be an actual word
 - Spelling corrections are crucial
 - Often based on a word list, such as that available at WordNet
- Part of speech guessing
 - Knowledge of morphemes for a particular language can be a powerful aid in guessing the part of speech for an unknown term

Stemming

- Removal of affixes (usually suffixes) to arrive at a base form that may or may not necessarily constitute an actual word
- Continuum from very conservative to very liberal modes of stemming
 - Very Conservative
 - Remove only plural *-s*
 - Very Liberal
 - Remove all recognized prefixes and suffixes
- Good resource:
 - <http://www.comp.lancs.ac.uk/computing/research/stemming/>

for example compressed and compression are both accepted as equivalent to compress.



for exampl compress and compress ar both accept as equal to compress

Porter Stemmer

- Popular stemmer based on work done by Martin Porter
 - M.F. Porter. An algorithm for suffix stripping. 1980, Program 14(3), pp. 130-137.
- Very liberal step stemmer with five steps applied in sequence
 - See example rules on next slide
- Probably the most widely used stemmer
 - Has been incorporated into a number of Information Retrieval systems
- Does not require a lexicon.
- Open source software available for almost all programming languages.

Examples of Porter Stemmer Rules

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ ∅	cats	→ cat

Step 1b

(*v*)ing	→ ∅	walking	→ walk
		sing	→ sing
(*v*)ed	→ ∅	plastered	→ plaster

...

Where *v* is the occurrence of any vowel.

From Dan Jurafsky

Step 2 (for long stems)

ational	→ ate	relational	→ relate
izer	→ ize	digitizer	→ digitize
ator	→ ate	operator	→ operate

...

Step 3 (for longer stems)

al	→ ∅	revival	→ reviv
able	→ ∅	adjustable	→ adjust
ate	→ ∅	activate	→ activ

...

Some other Stemmers for English

- Paice-Husk Stemmer
 - Simple iterative stemmer; rather heavy when used with standard rule set
- Krovetz Stemmer
 - Light stemmer; removes inflections only; removal of inflections is very accurate (actually a lemmatizer)
 - Often used as a first step before using another stemmer for increased compression
- Lovins Stemmer
 - Single-pass, context-sensitive, longest match stemmer; not widely used
- Dawson Stemmer
 - Complex linguistically targeted stemmer based on Lovins; not widely used

Lemmatization

- Removal of affixes (typically suffixes),
- But the goal is to find a base form that does **constitute an actual word**
- Example:
 - *parties* → remove *-es*, correct spelling of remaining form
→ *party*
- Spelling corrections are often rule-based
- May use a lexicon to find actual words

Guessing the Part of Speech

- English is continuously gaining new words on a daily basis
- And new words are a problem for many NLP systems
 - New words won't be found in the MRD or lexicon, if one is used
- How might morphology be used to help solve this problem?
- What part of speech are:
 - clemness
 - foramtion
 - depickleated
 - outtakeable

Ambiguous Affixes

- Some affixes are ambiguous:
 - -er
 - Derivational: Agentive –er Verb + -er > Noun
 - Inflectional: Comparative –er Adjective + -er > Adjective
 - -s or –es
 - Inflectional: Plural Noun + -(e)s > Noun
 - Inflectional: 3rd person sing. Verb + -(e)s > Verb
 - -ing
 - Inflectional Progressive Verb + -ing > Verb
 - Derivational “act of” Verb + -ing > Noun
 - Derivational “in process of” Verb + -ing > Adjective
- As with all other ambiguity in language, this morphological ambiguity creates a problem for NLP

Complex Morphology

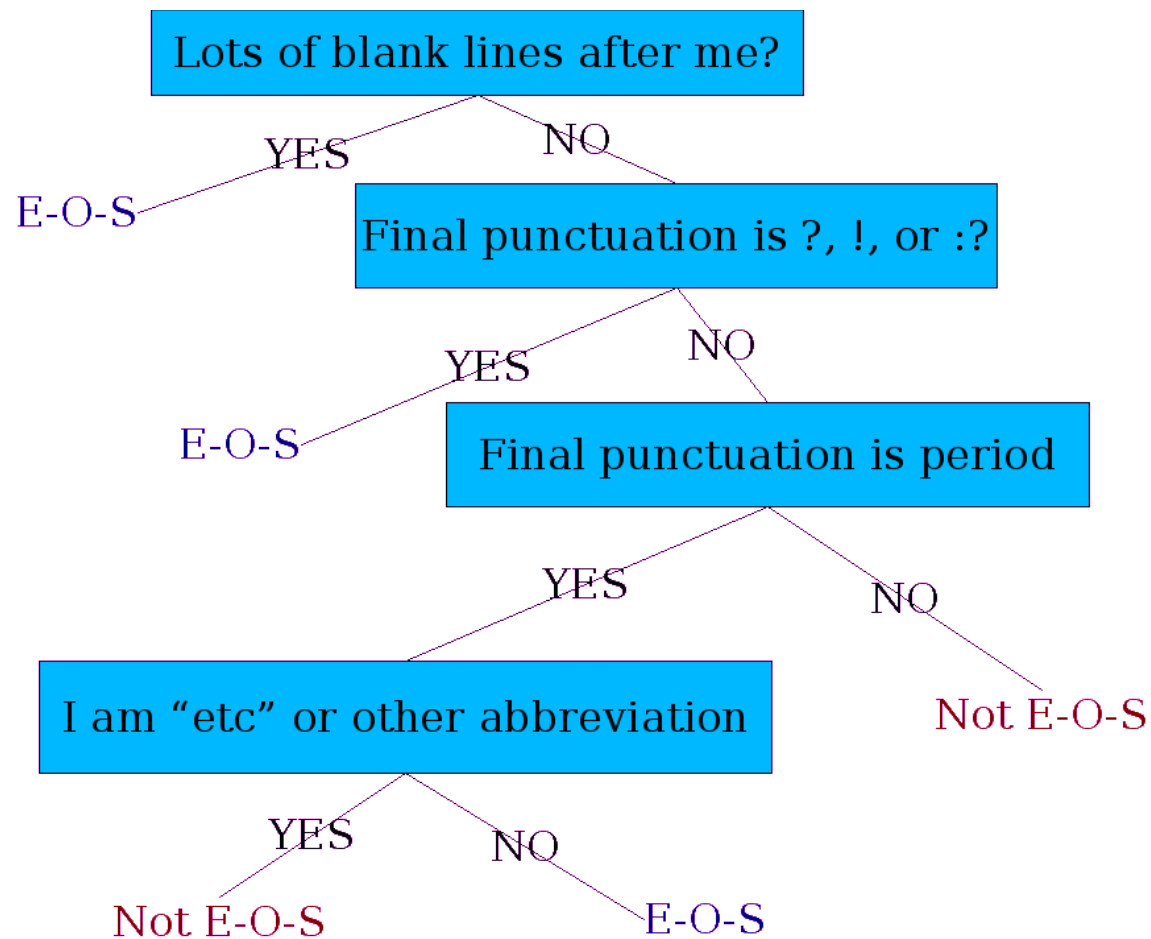
- Some languages requires complex morpheme segmentation
 - Turkish
 - **Uygarlastiramadiklarimizdanmissinizcasina**
 - ‘(behaving) as if you are among those whom we could not civilize’
 - **Uygar** ‘civilized’ + **las** ‘become’
 - + **tir** ‘cause’ + **ama** ‘not able’
 - + **dik** ‘past’ + **lar** ‘plural’
 - + **imiz** ‘p1pl’ + **dan** ‘abl’
 - + **mis** ‘past’ + **siniz** ‘2pl’ + **casina** ‘as if’

Sentence Segmentation

- !, ? are relatively unambiguous
- Period “.” is quite ambiguous
 - Sentence boundary
 - Abbreviations like Inc. or Dr.
 - Numbers like .02% or 4.3
- Treat this as a **classification problem**
 - Looks at a “.” (or the word preceding the “.”)
 - Decides EndOfSentence/NotEndOfSentence
 - Classifiers: hand-written rules, regular expressions, or machine-learning

Classify whether a word is End-of-Sentence

- An example of one way to classify is a Decision Tree:



Classification Problem Features

- Each property used in the decision tree to decide which branch to take is usually called a **feature** of the word
- More sophisticated features for end-of-sentence decision
 - Word shape features
 - Case of word with “.”: Upper, Lower, Cap, Number
 - Look at word after “.” to see if it begins a new sentence:
 - Case of word after “.”: Upper, Lower, Cap, Number
 - Numeric features
 - Length of word with “.”
 - Probability(word with “.” occurs at end-of-s)
 - Probability(word after “.” occurs at beginning-of-s)