CSME 206A – Natural Language & Speech Processing **Spring Semester**

Lecture 11: English language Morphology

Lecturer: K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

11.1 English Language Morphology

Morphology is study of, how the words are constructed. Construction of English language words through attachment of prefixes and suffixes (both together called *affix*) are called *concatenative morphology*, because a word is composed of number of morphemes concatenated together. A word may have more than one affix, for example rewrites (*re+write+s*), unlikely (*un+like+ly*), etc. There are broadly two ways to form words using morphemes:

1. *Inflection:* Inflectional morphology forms words using the same group word stem, e.g., write+*s*, word+*ed*, etc. The Table 11.1 shows the words constructed using inflective morphology.

   It is modification of a word to express different grammatical categories. Inflectional morphology is the study of processes, including affixation and vowel change, that distinguish word forms in certain grammatical categories. Inflectional morphology consists of at least five categories, from *language typology* and *syntactic description*, i.e., *grammatical categories* and the *lexicon.* However, the derivational morphology cannot be so easily categorized because derivation is not as predictable as inflection.

   Table 11.1: Inflectional Morphology

Type	Regular nouns	Irregular nouns
Singular	cat, thrush	mouse, ox
Plural	cats, thrushes	mice, oxen

2. *Derivation:* Derivations morphology produce a words of different stem, for example computerization (noun) from computerize (verb) – the words belong to different groups.

   This morphology creates new lexemes, either by changing the syntactic category (part-of-speech) of a base or by adding substantial, nongrammatical meaning or both. On the one hand, derivation may be distinguished from inflectional morphology, which typically does not change category but rather modifies lexemes to fit into various syntactic contexts; inflection typically expresses distinctions like number, case, tense, aspect, person, among others. On the other hand, derivation may be distinguished from compounding, which also creates new lexemes, but by combining two or more bases rather than by affixation, reduplication, subtraction, or internal modification of various sorts. Although the distinctions are generally useful, in practice applying them is not always easy.

Table 11.2: Derivational Morphology

Suffix	Base	Derived
	verb/adjective	Noun
-ation	computerize (V)	Computerization
-ee	appoint (V)	appointee
-er	kill (V)	killer
-ness	fuzzy (A)	fuzziness

The examples of *regular verbs* are walk, walks, walking, walked. Similarly, *irregularly inflected* verbs are: “eat, eats, eating, ate, eaten, catch, catches, cut, cuts, cutting, caught,” etc.

The derivation is a combination of word stem with *grammatical morpheme*, usually resulting in a word of different class. For example, formation of nouns from verbs and adjectives. The Table 11.2 shows the examples of derivational morphology.

Before we can do the morphological analysis, the text sentence need to be tokenized, which through NLTK (Natural Language Toolkit) and Python language, as shown in the following example.

Example: Sentence tokenization using Python.

```
$ python
python 3.7.6 (default, Jan 8 2020)
>>> from nltk import tokenize
>>> text="This is a simple sentence, that any one can
     write."
>>> tokens=nltk.word_tokenize(text)
>>> print(tokens)
[’This’, ’is’, ’simple’, ’sentence’, ’,’, ’that’, ’any’,
  ’one’, ’can’, write’, ’.’]
>>>
```

□

There are various approaches to morphology, as follows:

Morpheme Based Morphology Word-based morphology is often a word-and-paradigm approach. The theory takes paradigms as a central notion. Instead of stating rules to combine morphemes into word forms or to generate word forms from stems, word-based morphology states generalizations that hold between the forms of inflectional paradigms.

Lexeme Based Morphology Lexeme-based morphology usually takes what it is called an “item-and process” approach. Instead of analyzing a word form as a set of morphemes arranged in sequence, a word form is said to be the result of applying rules that alter a word-form or stream in order to produce a new one.

Word based Morphology Word-based morphology is usually a word-and -paradigm approach instead of stating rules to combine morphemes into word forms.

11.2 Application of Morphological Analysis

One application is in Text to Speech Synthesis. The morphological analysis can be used to reduce the size of lexicon and also plays an important role in determining the pronunciation of a homograph.

Other application is in Machine translation. Machine translation mainly helps the people who are belonging to the different communities and want to interact with the data present in the different languages. In lack of Morphological analysis, we need to store all the word forms, this will increase the size of database and will take more time to search. One more benefit of this analyzer is it provides the information of the word such as number, gender. This information can be used in target language to generate the correct form of the word.

Other application is in Spell Checker. A Spell checker is an application that is used to identify whether a word has been spelled correctly or not. Spell checker functionality can be divided into two parts: Spell check error detection and Spell check error correction. Spell check error detection phase only detects the error while Spell check error correction will provide some suggestions also to correct the error detected by Spell check error detection phase. One more advantage of using morphology based spell checker is that it can handle the name entity problem. If any word is not included in the lexicon, can be added easily.

Other application is in Search Engine. Morphological Analysis and Generation improves the result of the search engine. Suppose if a word is provided as a input but this word is not present in the lexicon, in that case Morphological analysis of that word is done.

11.3 Morphology and Finite-state Transducers

To know the structure about a word when we perform the *morphological parsing* for that word, given a *surface form* (input form), e.g., "going" we might produce the parsed form: *verb-go + gerund-ing*. Morphological parsing can be done with the help of finite-state transducer. A *morpheme* is a meaning bearing unit of any language. For example,

> *fox*: has single morpheme, *fox*,
>
> *cats*: has two morphemes, *cat, -s*,
>
> Similarly, eat, eats, eating, ate, eaten have different morphemes.

Some examples of mapping of certain words and corresponding morphemes are given in the Table 11.3, these mapping of input and output corresponds to the input and output of finite state transducers.

In speech recognition, when a word has been identified, like cats, dogs, it becomes necessary to produce its morphological parsing, to find out its true meaning, in the form of its structure, as well to know how it is organized. These include the features, like N (Noun), V (Verb), specify additional information about the word stem, e.g., *+N* means that word is noun, *+SG* means singular, +PL for plural, etc.

11.3.1 Databases for Morphological Parsing

The following databases are needed for building a morphological parser:

1. *Lexicon.* List of stems, and affixes, plus additional information about them, like *+N, +V*.

Table 11.3: Mapping of input word to Morphemes.

Input Words	Morphological parsed output
cats	cat *+N +PL*
cat	cat *+N +SG*
cities	city *+N +PL*
geese	goose *+N +PL*
goose	goose *+V +3SG*
caught	catch *+V +PAST-Part*

2. *Morphotactics rules.* Rules about ordering of morphemes in a word, e.g. *-ed* is followed after a verb (e.g., worked, studied), *un* (undo) precede a verb, for example, unlock, untie, etc.

3. *Orthographic rules* (spelling rules). For combining morphemes, e.g., city+ *-s* gives cities and not citys.

We can use the *lexicons* together with *morphotactics* (rules) to recognize the words with the help of finite automata in the form of stem+affix+part-of-speech (N, V, etc).

11.3.2 Morphological Parsing

The Table 11.4 shows some examples of regular and irregular nouns, and the Fig. 11.1 shows the basic idea of parsing of nouns using morphological parsing. The regular-noun, e.g., "cat" has corresponding three transitions c-a-t between q_0 and q_1 states.

Table 11.4: Regular and Irregular nouns.

Reg-noun	Plural	Irreg-noun	Irreg-sg-noun
fox	-es	goose	geese
cat	-s	sheep	sheep
dog	-s	mouse	mice

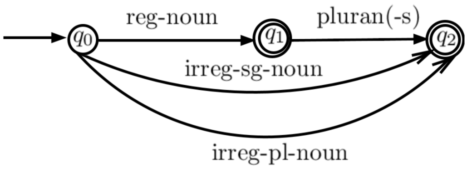

Figure 11.1: Morphological Parsing of nouns

Similarly, what is shown as a single transition for irregular-noun between q_0 and q_2 states, actually consists sequence of transitions, e.g., for the word "goose" there are five transitions, and for plural noun "geese" there are also five transitions. Such detailed transitions for parsing of noun words are shown in Fig. 11.2.

At the next stage, the lexicon can be expanded to sub-lexicons, i.e, individual letters, to be recognized by the finite automata. For example, regular-noun in Fig. 11.1 can be expanded to letters "*f o x*" connected

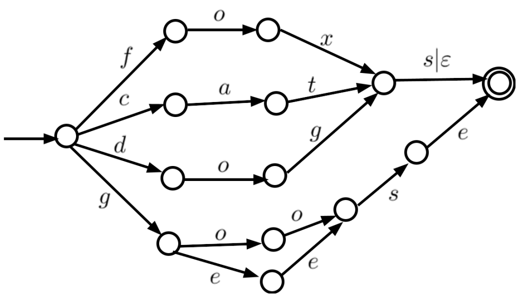

Figure 11.2: Morphological Parsing for noun words in details

by three states in a transition diagram. The three transitions are $f : f, o : o, x : x$. Similarly, the regular verb stem in Fig. 11.3 can be expanded by letters “*w a l k*”, and so on, as shown in Fig. 11.2. Note that in the parsing of N, V, ADJ, and ADV discussed above, for the sake of simplicity we have not shown the transitions separated by colon (‘:”), however, the FST has two tapes as usual, for input and output.

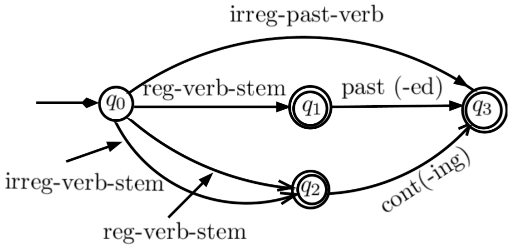

Figure 11.3: Morphological Parsing of verbs

A similar arrangement is possible for *verbs*' morphological parsing (see Fig. 11.3, and Table 11.5). The lexicon for verbal inflection have three stem classes (*reg-verb stem, irreg-verb stem,* and *irreg-past-verb*), with affix classes as: *-ed* for past and participle, *-ing* for continuous, and 3rd person singular has *-s*.

Table 11.5: Regular and Irregular verbs

Reg-verb	Past	Irreg-verb	Irreg-past-v	Cont.	3sg
walk	-ed	catch	caught	-ing	-s
fry	-ed	eat	ate	-ing	-s
talk	-ed	sing	sang	-ing	-s

Adjectives can be parsed in the similar manner like, the nouns and verbs. Some of the adjectives of English language are: big, bigger, biggest, clean, cleaner, cleanest, happy, unhappy, happier, happiest, real, really, unreal, etc. The finite automata in Fig. 11.4 is showing the morphological parsing for adjective words.

11.4 Morphological Analysis using Finite State Transducers

The objective of the morphological parsing is to produce output lexicons for a single input lexicon, e.g., like it is given in table 11.6. The second column in the table contains the stem of the corresponding word

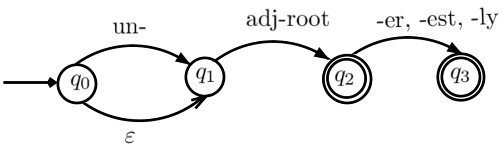

Figure 11.4: Morphological Parsing for adjectives

(lexicon) in first column, along with its morphological features, like, $+N$ means word is noun, $+SG$ means it is singular, $+PL$ means it is plural, $+V$ for verb, and *pres-part* for present participle. We achieve it through two level morphology, which represents a word as a correspondence between lexical level – a simple concatenation of lexicons, as shown in column 2 of Table 11.6, and a surface level as shown in column 1. The two columns corresponds to two tapes of a finite state transducer 11.5.

Table 11.6: Lexical Transformation table.

Input	**Parsed output**
cat	cat $+N$ $+SG$
cats	cat +N $+PL$
geese	goose $+N$ $+PL$
reading	read $+V$ $+$*Pres-part*

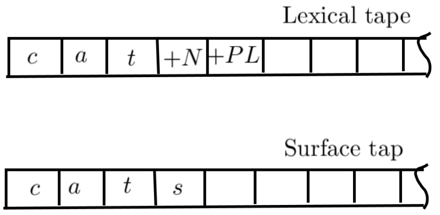

Figure 11.5: A FST

The FST is a multi-function device, and can be viewed in the following ways:

- *Translator*: It reads one string on one tape and outputs another string on other tape, it may receive input "cats" on surface tape, and produce parsed output "cat +N +PL" on lexical tape. Alternatively, the role of input and output tape can be interchanged.

- *Recognizer*: It takes a pair of strings as two tapes and accepts/rejects based on their matching. Foe example, when both the contents are as shown in Fig. 11.5, then it accepts translation, if one of the tape is having different contents, then the FST rejects (no match).

- *Generator*: It outputs a pair of strings of that language, on two tapes along with yes/no result based on whether they are matching or not. Hence, acts as generator.

- *Relater*: It compares the relation between two sets of strings available on two tapes.

11.4.1 Closure properties of FSTs

Like FSA (Finite State Automata) are isomorphic to regular expressions, the FSTs are isomorphic to *regular relations*. The FSTs are closed on the following relations:

1. *Union*: If R_1 and R_2 are relations on FST, then $R_1 \cup R_2$ is also a relation on FST.

2. *Composition:* If T_1 is FST from I_1 to O_1, and T_2 is FST from I_2 to O_2, then $T_2 \circ T_1$ is FST from I_1 to O_2.

3. *Inversion:* The FSTs are closed on *inversion.* A transducer T^{-1} simply switches the input and output labels on each transition.

The composition operation is useful because it replaces two FST running in series by a single FST. The composition works as in algebra. Applying $T_2 \circ T_1$ to input sequence S is equal to applying T_1 to S, and then T_2 to result $T_1(S)$, i.e.,

$$T_2 \circ T_1(S) = T_2(T_1(S)) \tag{11.1}$$

Similarly, the composition is useful to convert a FST as *parser* to FST as a *generator*[1],[2].

11.4.2 Morphological parsing using FST

In two level morphology, the lexical tape is composed of symbols from a in $a : b$ pairs, and the surface tape comprises the symbols from b in this pair. Hence, each symbol pair $a : b$ gives mapping from one tape to other tape. The symbols $a : a$ are called *default pairs* and written simply as a, as Fig. 11.6 shows transitions: $q_0 - q_1$.

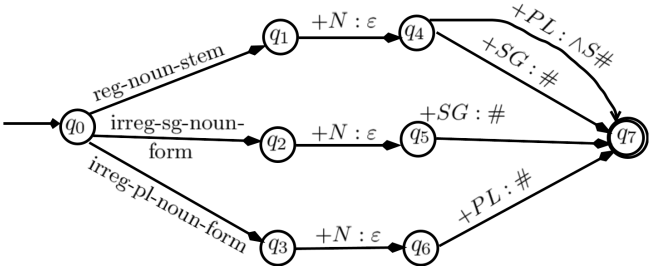

Figure 11.6: Morphological Parsing using FST

The Fig. 11.6 shows the transition diagram for FST with additional symbols $+SG$ (singular), $+PL$ (plural), corresponding to each morpheme. These symbols map to empty string (ε), as there are no corresponding symbols on output (surface) tape. The word, *stem* is basic word, for example, “cat” is stem for both its singular and plural forms.

The symbol # stands for boundary symbol. Typical example of mapping, e.g., in case of word “geese” (irregular noun) on surface tape, will be parsed into *goose +N +PL* on lexical tape, and symbols on the arc joining states q_0 - q_2 are “$g : g\ o : e\ o : e\ s : s\ e : e$”, which is written as “$g\ o : e\ o : e\ s\ e$”. Since, there are five letters in the word, there will be five state transitions between q_0 - q_2. For regular noun, like *fox*, it will be “$f : f\ o : o\ x : x$”. The surface form “geese” is mapped to lexical form “*goose +N +SG*” through *cascading* the FSTs, where two automata are run in series, i.e. output of first becomes input to next. This is what we discussed earlier as P-subsequential transducer.

[1] *Parser*: A parser converts a word into its constituent components, e.g., “cats” is parsed into “cat +N +PL.”

[2] *Generator:* Given “cat” as a lexicon for noun, and that its plural form +PL, use “cat +N +PL”, generate the word “cats.”

Instead of cascading two transducers, we perform this job using *composition* operator. Composing the transducers in this way helps in taking many different levels of input and outputs, and converting them into a single two level transducer with one input and one output tape. A typical FST, which results for morphological parsing of "cat" is shown in Fig. 11.7, producing a mapping $c{:}c\ a{:}a\ t{:}t\ +N{:}\varepsilon\ +PL{:}{}^\wedge S\#$. The symbol sequence $+PL$ maps to ${}^\wedge S$. The symbol $^\wedge$ indicates the morpheme boundary, and # indicates the word boundary. In fact, the fist transducer T_1 will have *surface* tape as input and the *intermediate* tape as output. And, the second transducer (T_2) has intermediate tape as input and lexical tape as output tape. Hence, the composition $L = T_2 \circ T_1(S)$, where S source tape and L is content of lexical tape.

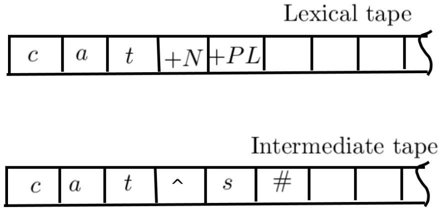

Figure 11.7: Morphological Parsing with Lexical and Intermediate tapes

11.4.3 Implementing Orthographic Rules

We note that concatenating the morphemes can work to parse the words like "dog", "cat", "fox", but this simple method does not work when there is spelling change, like "foxes" is to be parsed into lexicons "fox +N +PL" or "cats" is to be parsed into "cat +N +3SG", etc. This requires introduction of spelling rules (also called orthographic rules).

To account for the spelling rules, we introduce another tape, called *intermediate tape*, which produces the output slightly modified, thus going from 2-level to 3-level morphology. Such a rule maps from intermediate tape to surface tape. For plural nouns, the rule states, "insert *e* on the surface tape just when intermediate tape has a morpheme ending in *x* or *z* or *s* and next morpheme is -*s*". The examples are *ox* to *oxes*, *fox* to *foxes*, *rose* to *roses*, *doss* to *dosses*, etc.

Definition 11.1 *(Chomsky and Hall rule).The rule is stated as in equation (11.2),*

$$\varepsilon \to e/ \left\{\begin{array}{c} x \\ s \\ z \end{array}\right\} {}^\wedge - - - S\# \tag{11.2}$$

and called Chomsky and Hall *notation. A rule of the form $a \to b/c - d$ means rewrite a as b, when it occurs between c and d.*□

Since symbol ε is null, and it occurs between $^\wedge$ and S on intermediate tape, therefore replacing ε (null) by e means inserting e between $^\wedge$ and S. In the symbol $^\wedge$ indicates morpheme boundary. These boundaries are deleted by including the symbol pair $^\wedge : \varepsilon$ in Fig. 11.6, the default pairs for the transducer ($I : O$), i.e., in the graph, the symbol ':' indicates that 1st symbol is on intermediate tape and ε is on surface tape. The mapping of symbols shown in Fig. (11.8), is called *morphological parsing.* There are n number of FSTs, indicating that there are n number of rules encoded.

Using these multi-level FSTs in sequence between different tapes, as well as through parallel transducers for spelling checks, we are able to parse those words whose morphological analysis is simple. However,

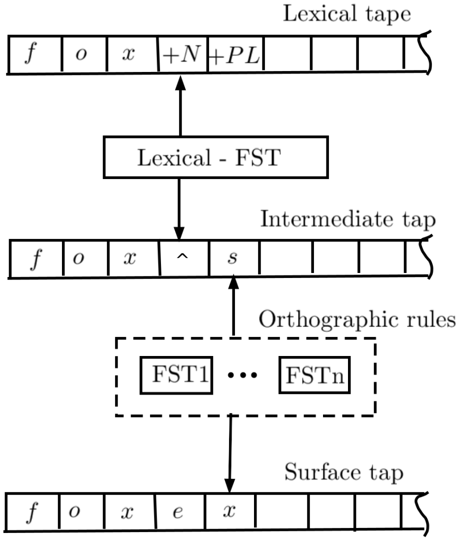

Figure 11.8: Morphological Parsing using 3-tape FSTs

considering the sentence "The police books the right culprit", here it is not clear as per above rules that whether the lexical parser's output is "book +N +PL" or it is"book +V +3SG" ! However, to human it is not difficult to infer that it is the second option. This is due to the ambiguity in the word, which may be a noun or a verb, depending on its position in a sentence. This type of ambiguity is called *lexical ambiguity*, and is the subject of later discussions.