ANISH ADNANI

NLP

(1)

017B/01

EXPERIMENT 4.

Aim: Perform morphological analysis and word generation for any given text

Theory:

Morphology

Morphological analysis is a method of identifying, structuring and investigating the total set of possible relationships contained in a given multidimensional problem complex. It allows small groups of subject specialist to define, link & internally evaluate the parameters of complex problem spaces, creating a solution space and a flexible inference model. It has been applied successfully in stratagical planning and decision support in various domains.

A morphological analyzer or generator supplies information concerning morphosyntactic properties of the words it analyses. Morphological analysis and generation are important concepts for building computational grammar as well as for machine translation.

Examples

- washing = wash + ing
- Browser = browse + er
- Rats = rat + s

Types of morphology

- Inflectional morphology : modification of a word to express different grammatical categories.
  Example - cat, men, etc

- Derivational morphology : creation of a new word from existing word by changing grammatical category
  Example - happiness, brotherhood, etc.

Approaches to morphology

There are three principle approaches to morphology

- morpheme based morphology
- Lexeme based morphology
- word based morphology

Morphological analysis

- Analysis words into their linguistic components
- Ambuiguity : more than one alternatives

Files

fly_{verb} + PROG

fly_{noun} + PLU

Need for morphological Analysis

- wastage of memory in exhaustive lexicon
- Failure to depict linguistic generalization necessary to understand an unknown word.

- morphologically rich & productive languages might be problematic

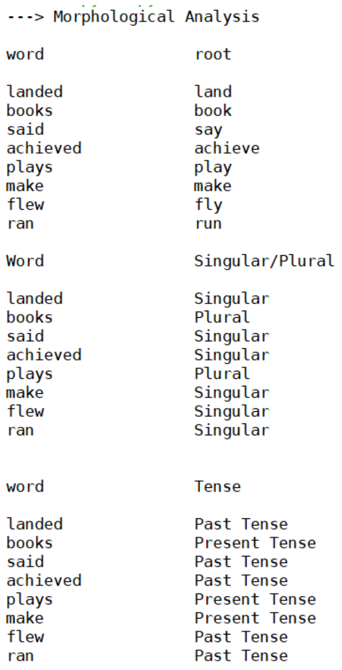
The idea to make a device write on a human's behalf is a practice referred to as text generation or word generation, which is a subfield of Natural Language Processing.

Conclusion: Thus we have seen what morphological analysis & word generation is, and implemented it successfully.

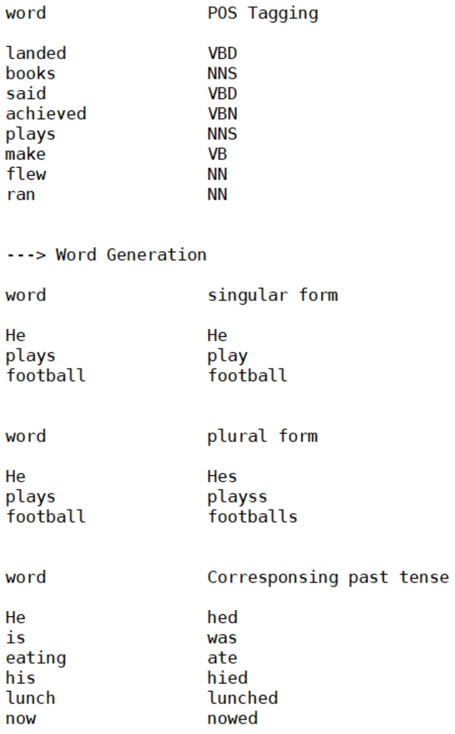
Program:
Morphological Analysis and Word Generation

```
import nltk
from pattern.en import conjugate, lemma, lexeme,PRESENT,SG, PAST, FUTURE
from textblob import TextBlob
from nltk.corpus import wordnet as wn

print("---> Morphological Analysis")
print()

Words = ["landed", "books", "said", "achieved", "plays", "make", "flew", "ran"]

print("{0:20}{1:20}".format("word", "root"))
print()

for word in Words:
    root = lemma(word)
    print("{0:20}{1:20}".format(word, root))

print()
print("{0:20}{1:20}".format("Word", "Singular/Plural"))
print()

for word in Words:
    sgpl = TextBlob(word).words.singularize()[0]
    if sgpl == word:
        sgpl = "Singular"
    else:
        sgpl = "Plural"

    print("{0:20}{1:20}".format(word, sgpl))

print()
print()
print("{0:20}{1:20}".format("word", "Tense"))
print()

for word in Words:
    s1 = conjugate(verb=word,tense=PAST,number=SG)
    s2 = conjugate(verb=word,tense=PRESENT,number=SG)
    if s1 == word.rstrip():
        s = "Past Tense"
    else:
        s = "Present Tense"
    pos = nltk.pos_tag(nltk.word_tokenize(word))[0][1]

    print("{0:20}{1:20}".format(word, s))

print()
print()
print("{0:20}{1:20}".format("word", "POS Tagging"))
print()

for word in Words:
    pos = nltk.pos_tag(nltk.word_tokenize(word))[0][1]
    print("{0:20}{1:20}".format(word, pos))

print()
print()
print("---> Word Generation")
print()

sentence = "He plays football"

print("{0:20}{1:20}".format("word", "singular form"))
print()

for word in sentence.split():
    sg = TextBlob(word).words.singularize()[0]
    print("{0:20}{1:20}".format(word, sg))

print()
print()
print("{0:20}{1:20}".format("word", "plural form"))
print()

for word in sentence.split():
    pl = TextBlob(word).words.pluralize()[0]
    print("{0:20}{1:20}".format(word, pl))

sentence = "He is eating his lunch now"

print()
print()
print("{0:20}{1:20}".format("word", "Corresponsing past tense"))
print()

for word in sentence.split():
    past = conjugate(verb=word,tense=PAST,number=SG)
    print("{0:20}{1:20}".format(word, past))

print()
print()
print("{0:20}{1:20}".format("word", "Different Forms of Word in Multiple Tenses"))
print()

sentence = "He plays football"

for word in sentence.split():
    s = "[" + ', '.join(lexeme(word)) + "]"
    print("{0:20}{1:20}".format(word, s))
```

Output:



```
---> Morphological Analysis

word                root

landed              land
books               book
said                say
achieved            achieve
plays               play
make                make
flew                fly
ran                 run

Word                Singular/Plural

landed              Singular
books               Plural
said                Singular
achieved            Singular
plays               Plural
make                Singular
flew                Singular
ran                 Singular


word                Tense

landed              Past Tense
books               Present Tense
said                Past Tense
achieved            Past Tense
plays               Present Tense
make                Present Tense
flew                Past Tense
ran                 Past Tense
```



```
word                 POS Tagging

landed               VBD
books                NNS
said                 VBD
achieved             VBN
plays                NNS
make                 VB
flew                 NN
ran                  NN


---> Word Generation

word                 singular form

He                   He
plays                play
football             football


word                 plural form

He                   Hes
plays                playss
football             footballs


word                 Corresponsing past tense

He                   hed
is                   was
eating               ate
his                  hied
lunch                lunched
now                  nowed
```

```
word                Different Forms of Word in Multiple Tenses

He                  [he, hes, hing, hed]
plays               [play, plays, playing, played]
football            [football, footballs, footballing, footballed]
```