In [1]:

```
# test tokenizaion
from nltk import sent_tokenize
import nltk
nltk.download('punkt') #download the period, enter ...

# steve jobs' speech
text_sample = 'I am honored to be with you today at your commencement from one of t

sentences = sent_tokenize(text=text_sample)

print(type(sentences),len(sentences))
print(sentences)
```

```
[nltk_data] Downloading package punkt to /home/pict/nltk_data...
```

```
<class 'list'> 7
['I am honored to be with you today at your commencement from one of t
he finest universities in the world.', 'I never graduated from colleg
e.', "Truth be told, this is the closest I've ever gotten to a college
graduation.", 'Today I want to tell you three stories from my life.',
"That's it.", 'No big deal.', 'Just three stories.']
```

```
[nltk_data]   Package punkt is already up-to-date!
```

In [2]:

```
# word tokenization
from nltk import word_tokenize

sentence = 'The first story is about connecting the dots.'
words = word_tokenize(sentence)
print(type(words),len(words))
print(words)
```

```
<class 'list'> 9
['The', 'first', 'story', 'is', 'about', 'connecting', 'the', 'dots',
'.']
```

In [3]:

```
from nltk import word_tokenize, sent_tokenize

# function that break the document into word

def tokenize_text(text):

    # tokenize sentence
    sentences = sent_tokenize(text)

    # tokenize word
    word_tokens = [word_tokenize(sentence) for sentence in sentences]

    return word_tokens

word_tokens = tokenize_text(text_sample)
print(type(word_tokens),len(word_tokens))
print(word_tokens)
```

```
<class 'list'> 7
[['I', 'am', 'honored', 'to', 'be', 'with', 'you', 'today', 'at', 'you
r', 'commencement', 'from', 'one', 'of', 'the', 'finest', 'universitie
s', 'in', 'the', 'world', '.'], ['I', 'never', 'graduated', 'from', 'c
ollege', '.'], ['Truth', 'be', 'told', ',', 'this', 'is', 'the', 'clos
est', 'I', "'ve", 'ever', 'gotten', 'to', 'a', 'college', 'graduatio
n', '.'], ['Today', 'I', 'want', 'to', 'tell', 'you', 'three', 'storie
s', 'from', 'my', 'life', '.'], ['That', "'s", 'it', '.'], ['No', 'bi
g', 'deal', '.'], ['Just', 'three', 'stories', '.']]
```

In [4]:

```
import nltk
nltk.download('stopwords')

# available language
from nltk.corpus import stopwords
print('-----what languages do it have?-----')
print(stopwords.fileids(),'\n\n')

# check the english stop word
print('-----stop words of english-----')
print('the number of english stop word: ',len(nltk.corpus.stopwords.words('english'
print(nltk.corpus.stopwords.words('english')[:10])
```

```
-----what languages do it have?-----
['arabic', 'azerbaijani', 'bengali', 'danish', 'dutch', 'english', 'fi
nnish', 'french', 'german', 'greek', 'hungarian', 'indonesian', 'itali
an', 'kazakh', 'nepali', 'norwegian', 'portuguese', 'romanian', 'russi
an', 'slovene', 'spanish', 'swedish', 'tajik', 'turkish']


-----stop words of english-----
the number of english stop word:  179
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
"you're"]
```

```
[nltk_data] Downloading package stopwords to /home/pict/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [5]:

```
# let's eliminate stop word in word_tokens
stopwords = nltk.corpus.stopwords.words('english')
all_tokens = []

# eliminate stop word of sentence
for sentence in word_tokens:
    filtered_words = []

    #eliminate stop word of word
    for word in sentence:
        #make all of word into small letter
        word = word.lower()
        # check whether word is stopword or not
        if word not in stopwords:
            filtered_words.append(word)

    all_tokens.append(filtered_words)

print(all_tokens)
```

```
[['honored', 'today', 'commencement', 'one', 'finest', 'universities',
'world', '.'], ['never', 'graduated', 'college', '.'], ['truth', 'tol
d', ',', 'closest', "'ve", 'ever', 'gotten', 'college', 'graduation',
'.'], ['today', 'want', 'tell', 'three', 'stories', 'life', '.'],
["'s", '.'], ['big', 'deal', '.'], ['three', 'stories', '.']]
```

In [6]:

```
# stemming
from nltk.stem import LancasterStemmer
stemmer = LancasterStemmer()

print(stemmer.stem('working'),stemmer.stem('works'),stemmer.stem('worked'))
print(stemmer.stem('amusing'),stemmer.stem('amuses'),stemmer.stem('amused'))
print('You can see \'amuse\' recongnized into \'amus\' ')
```

```
work work work
amus amus amus
You can see 'amuse' recongnized into 'amus'
```

In [ ]:

```
# lemmatization
from nltk.stem import WordNetLemmatizer
import nltk
nltk.download('wordnet')

lemma = WordNetLemmatizer()

print(lemma.lemmatize('amusing', 'v'),lemma.lemmatize('amuses', 'v'),lemma.lemmatiz
print(lemma.lemmatize('happier', 'a'),lemma.lemmatize('happiest', 'a'))

print('It is more accurate !')
```

In [ ]:

```
```