

**ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ**

**ΕΡΓΑΣΤΗΡΙΟ 2**

**ΜΟΡΦΟΛΟΓΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ**

**ΦΟΡΤΩΣΗ ΑΠΑΡΑΙΤΗΤΩΝ ΕΡΓΑΛΕΙΩΝ**

1. Τρέξτε την Python μέσω του IDLE (Interactive Development Environment).

Start -> All Programs -> Python 2.5 -> IDLE.

2. Φορτώστε τα ακόλουθα εργαλεία (modules) του ΝLΤΚ και της Python:

>>>import nltk, re, pprint

**ΑΝΑΓΝΩΡΙΣΗ ΛΕΞΕΩΝ ΣΕ ΜΙΑ ΠΡΟΤΑΣΗ (TOKENIZATION)**

3. Μια πρόταση μπορεί να διασπαστεί σε λίστα λέξεων με την συνάρτηση *split()*.

>>>sent= ‘Hello world’

>>> words=sent.split()

>>>words[0]

‘hello’

>>>words[1]

‘world’

**ΜΟΡΦΟΛΟΓΙΚΗ ΕΠΙΣΗΜΕΙΩΣΗ (TAGGING)**

1. Η πιο συνηθισμένη αναπαράσταση μορφολογικής επισημείωσης (επισημείωση μερών του λόγου) είναι η εξής:

*Time/NN flies/VB very/ADB quickly/ADB.*

To NLTK διαθέτει μια συνάρτηση *str2tuple()* που μετασχηματίζει την παραπάνω δομή αλφαριθμητικών σε δυάδα λέξης/επισημείωσης (tuple). Η δυάδα είναι ένας πίνακας δύο στοιχείων.

>>> tagged\_token=nltk.tag.str2tuple(‘Time/NN’)

>>> tagged\_token

(‘Time’, ‘NN’)

>>>tagged\_token[0]

‘Time’

>>>tagged\_token[1]

‘NN’

1. Έστω ότι έχετε την πρόταση *Time/NN flies/VB very/ADB quickly/ADB.* Μετατρέψτε την σε δυάδες λέξης/επισημείωσης.

**Σημείωση:**

Στην Python υπάρχει σχήμα επανάληψης, ως εξής:

>>> for num in [1, 2, 3]:

print ‘The number is’, num

The number is 1

The number is 2

The number is 3

Παρατήρηση: η λίστα σας (αντίστοιχη με την λίστα [1, 2, 3] του παραδείγματος) είναι η διασπασμένη πρόταση.

------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------

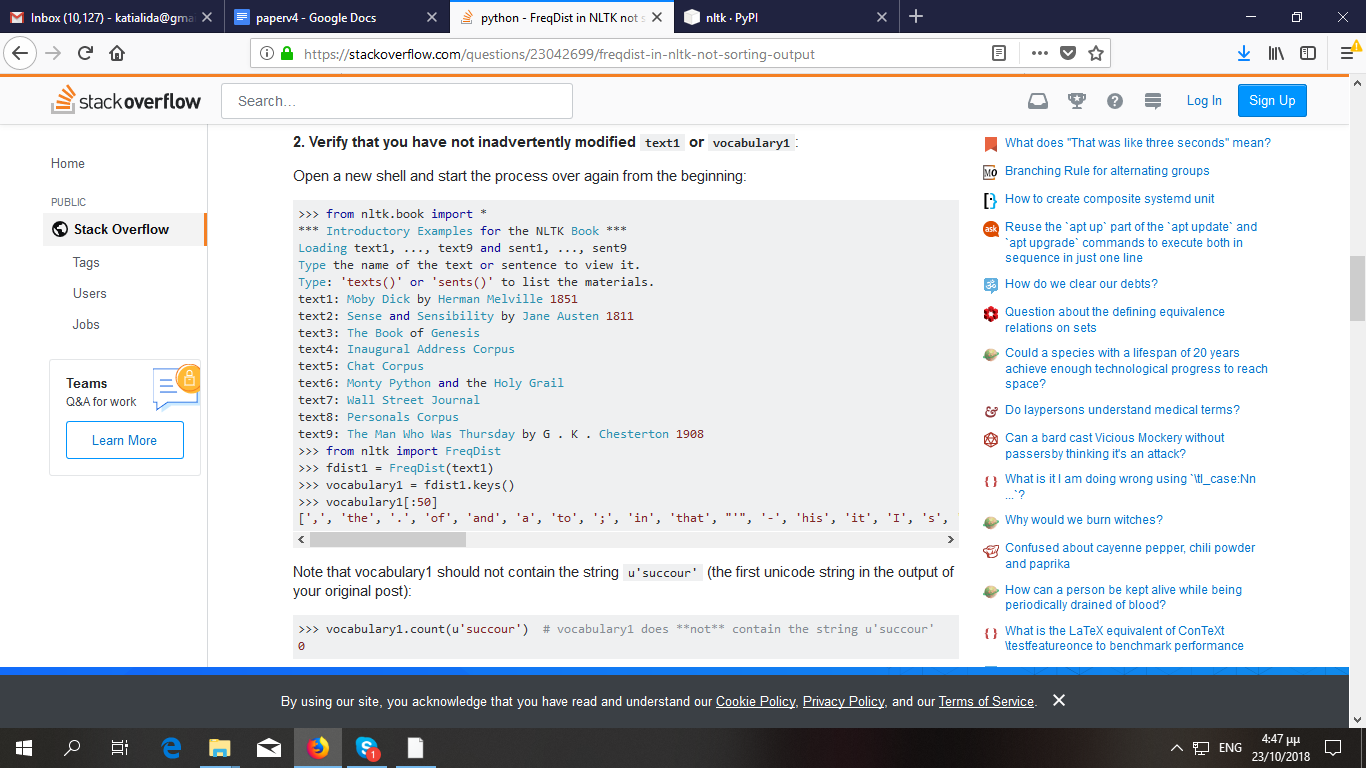
-------------------------------------------------------------------------------------------------------

**KATANOMEΣ ΕΠΙΣΗΜΕΙΩΣEΩN**

1. Φορτώστε το επισημειωμένο σώμα κειμένων Brown Corpus.

>>>brown\_a = nltk.corpus.brown.tagged\_words(categories= ‘adventure’)

1. Η συνάρτηση *FreqDist()* επιστρέφει την κατανομή συχνοτήτων των λέξεων ενός κειμένου.



>>> fd=nltk.FreqDist(brown\_a)

>>> c=0

>>> l=[]

>>> for item in fd:

if c==20: break

c+=1

l.append(item)

>>> l (βγαίνουν οι 20 πιο συχνές λέξεις)

>>> for item in l:

print item, fd[item] (εμφανίζεται και η συχνότητά τους)

Η τελευταία εντολή θα εμφανίσει τις 20 πιο συχνές δυάδες λέξεων/επισημειώσεων.

Για να δείτε πόσες φορές εμφανίζεται η δυάδα (‘for’, ‘IN’) πληκτρολογείτε:

>>>fd[(‘for’, ‘IN’)]

Όλες τις επισημειώσεις στο Brown Corpus και την ερμηνεία τους μπορείτε να τις βρείτε στην διεύθυνση http://www.scs.leeds.ac.uk/ccalas/tagsets/brown.html.

1. Για να βρείτε τα 20 πιο συχνά μέρη του λόγου στο κείμενο πληκτρολογείτε:

>>> fd1=nltk.FreqDist(tag for (word, tag) in brown\_a)

>>> c=0

>>> l=[]

>>> for item in fd1:

if c==20: break

c+=1

l.append(item)

>>> l

9. len(brown\_a) 🡪 θα μου δώσει τον αριθμό των λέξεων του κειμένου

len(fd) 🡪 θα μου δώσει τον αριθμό των διαφορετικών λέξεων του κειμένου

10. Δεδομένου ότι οι επισημειώσεις για όλα τα ρήματα ξεκινούν με τα γράμματα VB, τα 30 πιο συχνά ρήματα του κειμένου μπορούν να εμφανιστούν ως εξής:

>>>fd2=nltk.FreqDist((word,tag) for (word,tag) in brown\_a if tag[:2]=='VB')

>>> c=0

>>> l=[]

>>> for item in fd2:

if c==30: break

c+=1

l.append(item)

>>> l

**Σημείωση:**

Στην Python υπάρχει σχήμα ελέγχου ροής ως εξής:

>>> word= “cat”

>>> if len(word)<5:

print ‘word length is less than 5’

word length is less than 5

Οι τελεστές σύγκρισης είναι

> μεγαλύτερο >= μεγαλύτερο ή ίσο

< μικρότερο <= μικρότερο ή ίσο

== ίσο != διάφορο

10. Σε ομάδες των δύο βρείτε αγγλικές λέξεις που ανήκουν σε περισσότερα από ένα μέρη του λόγου. Στοιχηματίστε ο καθένας πάνω σε κάποιο μέρος του λόγου ότι είναι αυτό με το οποίο η λέξη εμφανίζεται πιο συχνά στο σώμα κειμένων. Βρείτε ποιος έχει δίκιο.

------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------

1. Επισημείωση με χρήση κανονικών εκφράσεων.

Το ΝLΤΚ περιλαμβάνει έναν tagger που χρησιμοποιεί κανονικές εκφράσεις, τον RegexpTagger().

Ορίζουμε μια λίστα από πρότυπα κανονικών εκφράσεων:

**>>> patterns = [**

**(r‘.\*ing$’, ‘VBG’), # gerunds**

**(r‘.\*ed$’, ‘VBD’), # simple past**

**(r‘.\*es$’, ‘VBZ’), # 3rd singular present**

**(r‘.\*ould$’, ‘MD’), # modals**

**(r‘.\*\’s$’, ‘NN$’), # possessive nouns**

**(r‘.\*s$’, ‘NNS’), # plural nouns**

**(r‘^-?[0-9]+(.[0-9]+)?$’, ‘CD’), # cardinal numbers**

**(r‘.\*’, ‘NN’) # nouns (default)**

**]**

Τα πρότυπα υφίστανται επεξεργασία διαδοχικά, και το πρώτο που θα «ταιριάξει», εφαρμόζεται. Επομένως, το τελευταίο πρότυπο είναι η προ-επιλογή (default). Αν δεν έχει βρεθεί κάποιο προηγούμενο «ταίριασμα», τότε η λέξη επισημειώνεται ως ουσιαστικό.

Ο επισημειωτής εφαρμόζεται στην τρίτη πρόταση του σώματος κειμένων ως εξής:

>>> regexp\_tagger = nltk.RegexpTagger(patterns)

>>> regexp\_tagger.tag(nltk.corpus.brown.sents(categories=‘adventure’)[3])

Η απόδοση του επισημειωτή μπορεί να μετρηθεί:

>>> regexp\_tagger.evaluate(nltk.corpus.brown.tagged\_sents(categories=‘adventure’))

**Εργασία (Προθεσμία 2 εβδομάδες)**:

Προσθέστε καινούρια πρότυπα κανονικών εκφράσεων και/ή τροποποιήστε τα ήδη υπάρχοντα με στόχο την βελτίωση της απόδοσης του επισημειωτή και την κάλυψη περισσότερων μερών του λόγου.

1. Επισημείωση με μοντέλο μονογράμμου (Unigram Tagging)

Η επισημείωση με μοντέλο μονογράμμου αναθέτει σε κάθε λέξη το μέρος του λόγου με το οποίο αυτή εμφανίζεται πιο συχνά. Αυτός ο αλγόριθμος εμπεριέχει μια διαδικασία εκπαίδευσης (training), κατά την οποία ο επισημειωτής «μαθαίνει» την πιο συχνή επισημείωση της κάθε λέξης.

>>> brown\_sen = nltk.corpus.brown.tagged\_sents(categories=‘adventure’)

Αρχικοποίηση και εκπαίδευση του επισημειωτή:

>>> unigram\_tagger = nltk.UnigramTagger(brown\_sen)

Εφαρμογή του επισημειωτή στην πρόταση 2007 του σώματος κειμένων:

>>> sent = nltk.corpus.brown.sents(categories=’adventure’)[2007]

>>> unigram\_tagger.tag(sent)

[(’Various’, None), (’of’, ’IN’), (’the’, ’AT’), (’apartments’, ’NNS’), (’are’, ’BER’),

(’of’, ’IN’), (’the’, ’AT’), (’terrace’, ’NN’), (’type’, ’NN’), (’,’, ’,’),

(’being’, ’BEG’), (’on’, ’IN’), (’the’, ’AT’), (’ground’, ’NN’), (’floor’, ’NN’),

(’so’, ’QL’), (’that’, ’CS’), (’entrance’, ’NN’), (’is’, ’BEZ’), (’direct’, ’JJ’), (’.’, ’.’)]

Υπολογισμός της απόδοσης του επισημειωτή:

>>> unigram\_tagger.evaluate(brown\_sen)

1. Επισημείωση με βάση προθέματα/επιθέματα (Affix Tagging)

Οι επισημειωτές με βάση το πρό/επί-θεμα εκπαιδεύονται πάνω στο πρόθεμα ή/και το επίθεμα μιας λέξης (έναν συγκεκριμένο αριθμό χαρακτήρων στην αρχή ή/και το τέλος μιας λέξης). Χρησιμοποιείται θετικός ακέραιος για το όρισμα affix length αν πρόκειται για πρόθεμα (δηλώνει το μήκος του προθέματος) και αρνητικός αν πρόκειται για επίθεμα (δηλώνει το μήκος του επιθέματος). Για παράδειγμα ο παρακάτω επισημειωτής θα λάβει υπόψη του επιθέματα μήκους τριών χαρακτήρων για λέξεις μήκους τουλάχιστον 5 χαρακτήρων.

>>> affix\_tagger = nltk.AffixTagger(brown\_sen, affix\_length=-3, min\_stem\_length=2)

>>> affix\_tagger.tag(sent)

[(’Various’, ’JJ’), (’of’, None), (’the’, None), (’apartments’, ’NNS’), (’are’, None),

(’of’, None), (’the’, None), (’terrace’, ’NN’), (’type’, None), (’,’, None),

(’being’, ’VBG’), (’on’, None), (’the’, None), (’ground’, ’NN’), (’floor’, ’NN’),

(’so’, None), (’that’, None), (’entrance’, ’NN’), (’is’, None), (’direct’, ’NN’),

(’.’, None)]

Πειραματιστείτε με διαφορετικά μήκη καταλήξεων (π.χ. 2 και 4) . Τι παρατηρείτε;

------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------

1. Επισημείωση με μοντέλο ν-γράμμου (Ν-gram Tagging)

Ο επισημειωτής που βασίζεται στο μοντέλο ν-γράμμων λαμβάνει υπόψη του τις επισημειώσεις των ν-1 προηγούμενων λέξεων (που προηγούνται της λέξης εστίασης, δηλ. της λέξης της οποίας αναζητείται το μέρος του λόγου) στο κείμενο, καθώς και την ίδια την λέξη εστίασης. Παράδειγμα ενός επισημειωτή βασισμένου στο μοντέλο διγράμμων φαίνεται παρακάτω:

>>> bigram\_tagger = nltk.BigramTagger(brown\_sen, backoff=unigram\_tagger)

>>> bigram\_tagger.tag(sent)

[(’Various’, ’JJ’), (’of’, ’IN’), (’the’, ’AT’), (’apartments’, ’NNS’), (’are’, ’BER’),

(’of’, ’IN’), (’the’, ’AT’), (’terrace’, ’NN’), (’type’, ’NN’), (’,’, ’,’),

(’being’, ’BEG’), (’on’, ’IN’), (’the’, ’AT’), (’ground’, ’NN’), (’floor’, ’NN’),

(’so’, ’CS’), (’that’, ’CS’), (’entrance’, ’NN’), (’is’, ’BEZ’), (’direct’, ’JJ’),

(’.’, ’.’)]