Πανεπιστήμιο Πατρών
Τμήμα Μηχ. Η/Υ & Πληροφορικής

ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ

ΕΡΓΑΣΙΑ 2016-2017

ΑΜ	E-mail	Ονοματεπώνυμο	Έτος
5753	dionys@ceid.upatras.gr	Διονυσόπουλος Φώτιος	4
5958	marinou@ceid.upatras.gr	Μαρίνου Ελευθερία- Αρετή	4

ΜΕΡΟΣ Α

Προσκομιστής Ιστοσελίδων.

Ο προσκομιστής ιστοσελίδων (crawler), υλοποιεί και παρακολουθεί μια σειρά απο ειδησεογραφικές ιστοσελίδες και κατεβάζει άρθρα στο σύστημα τα οποία μορφοποιούνται σε html. Συγκεντρώνουμε επομένως όλα τα Link από τις κλάσεις και τα tag που μας ενδιαφέρουν και κάνουμε χρήση του πακέτου module BeautifulSoup της python και χρησιμοποιούμε για δική μας ευρυθμία μια συνάρτηση που υλοποιήσαμε μόνοι μας η οποία επιτελεί την παραπάνω διαδικασία.

Προεπεξεργασία δεδομένων

Το συγκεκριμενο υποσύστημα εξάγει το καθαρό κείμενο από τις ιστοσελίδες που συγκεντρώσαμε. Ο καθαρισμός αφορά την απομόνωση του κειμένου περιεχομένου των html σελίδων σύμφωνα με τον τρόπο που διαμορφόνονται οι ειδήσεις στο κάθε ειδησιογραφικό site που παρακολουκεί ο προσκομιστής.
Κάνουμε χρήση της βιβλιοθήκης BeautifulSoup.

Μορφοσυντακτική Ανάλυση

Για το μορφοσυντακτικό σχολιασμό χρησιμοποιήσαμε PoSTagger . Στο τέλος της μορφοσυντακτικής ανάλυσης, κάθε κείμενο της συλλογής ιστοσελίδων κα υπόκειται μορφοσυντακτικό σχολιασμό (PoStags) για κάθε λέξη που περιέχει. Τα μορφοσυντακτικά σχολιασμένα κείμενα της συλλογής αποθηκεύονται σε βοηθητικό ενδιάμεσο κείμενο για μελλοντική χρήση.
Κάνουμε χρήση των συναρτήσεων word_tokenize, pos_tag απο το πακέτο nltk.s

Αναπαράσταση ιστοσελίδων στο Μοντέλο Διανυσματικού Χώρου.

Η αναπαράσταση του περιεχομένου κάθε κειμένου ως διάνυσμα υλοποιήθηκε μέσα από τα μορφοσυντακτικά σχολιασμένα κείμενα, αφού αφαιρέσαμε τους τερματικούς όρους (stopwords) από κάθε κείμενο. Οι τερματικοί όροι είναι λέξεις που δεν έχουν σημασιολογικό περιεχόμενο και εμφανίζονται σε όλα τα κείμενα, με αποτέλεσμα να μην αποτελούν χρήσιμους όρους δεικτοδότησης.
Στο παρακάτω link: http://www.infogistics.com/tagset.html βρήκαμε δύο πίνακες, με τα PoStags για open class categories και τα PoStags για closed class categories. Τα openclasscategories είναι γραμματικές κατηγορίες των λέξεων που έχουν σημασιολογικό περιεχόμενο όπου χρειαζόμαστε. Αντίθετα, τα closedclasscategories είναι γραμματικές κατηγορίες για λέξεις άνευ σημασιολογικού περιεχομένου, δηλ., stopwords. Συνεπώς, αφού εξαλείψουμε τους τερματικούς όρους από κάθε μορφοσυντακτικά σχολιασμένο κείμενο της συλλογής και έπειτα για κάθε μοναδικό λήμμα του κειμένου, μετρήσαμε τη συχνότητα εμφάνισης στο κείμενο.
Κάνουμε χρήση του TextBlob για τον υπολογισμό του TF-IDF.

Δημιουργία του ευρετηρίου

Στην φάση αυτή υλοποιήσαμε κατασκευή του ανεστραμμένου ευρετθρίου. Για την ολοκλήρωση της συλλογης ιστοσελίδων που έχουμε συγκεντρώσει εντοπίσαμε τα μοναδικά λιμματα, καθώς και τα κείμενα στα οποία εμφανίζεται το κάθε λιμμα. Δημιουργήσαμε τιςαντίστοιχες εγγραφές στο ανεστραμμένο ευρετήριο και υπολογήσαμε τα αντίστοιχα βάρη. Το βάρος του κάθε λιμματος για τα κείμενα αντιπροσωπεύει το βαθμό σπουδαιότητας του λιμματος για το συγκεκριμένο κείμενο και υπολογίστηκαν χρησιμοποιώντας τη μετρική TF-IDF .

Αποθήκευση και επαναφόρτωση ευρετηρίου - Αξιολόγηση ευρετηρίου.

Υλοποιήσαμε έναν απλό μηχανισμό υποβολής ερωτημάτων στο ευρετήριο. Ο μηχανισμός δέχεται input τα οποία ταυτοποιεί (με χρήση string matching) στα λιμματα του ευρετηρίου και κα επιστρέφει στο χρήστη τα id – url των ιστοσελίδων τα οποία περιέχουν το λιμμα του ερωτήματος.
Η λίστα των ιστοσελίδων που επιστρέφεται είναι ταξινομημένη σε φθίνουσα σειρά με βάσθό το TF-IDF βάρος που έχει το λήμμα του ερωτιματος για το κάκε κείμενο.
Έγινε χρήση των λεξικών της python καθώς επίσης και κάποιων πακέτων επεξεργασίας του xml αρχείου.

Επίσης, έγιναν μετρήσεις για 20 ερωτήματα της μιας λέξης, 20 ερωτήματα των δύο λέξεων, 30 ερωτήματα των τριών λέξεων και 30 ερωτήματα των τεσσάρων λέξεων.
Τα αποτελέσματα για τον μέσο χρόνο απόκρισης ήταν τα εξής:

Ερωτήματα της μιας λέξης: 0.0666532747626
Ερωτήματα των δύο λέξεων: 0.193450715084
Ερωτήματα των τριών λέξεων: 0.241616330449
Ερωτήματα των τεσσάρων λέξεων: 0.30124924232

Όσο αυξάνεται ο αριθμός των λέξεων αυξάνεται και ο μέσος χρόνος απόκρισης. Κάτι λογικό, αν σκεφτούμε ότι γίνεται αναζήτηση περισσότερων λέξεων.

ΜΕΡΟΣ Β

Κατηγοριοποίηση:
Στη μηχανική μάθηση, η ταξινόμηση είναι το πρόβλημα του προσδιορισμού σε ποιο σύνολο κατηγοριών ανήκει μια νέα παρατήρηση, με βάση ένα σετ εκπαίδευσης των δεδομένων που περιέχει τις παρατηρήσεις των οποίων η κατηγορία μέλους είναι γνωστή.
Ένας αλγόριθμος που υλοποιεί την ταξινόμηση, ειδικά σε μια συγκεκριμένη εφαρμογή, είναι γνωστός ως ταξινομητής. Ο όρος «ταξινομητής» μερικές φορές επίσης αναφέρεται στην μαθηματική συνάρτηση, που υλοποιείται από έναν αλγόριθμο ταξινόμησης, που χαρτογραφεί την εισαγωγή δεδομένων σε μια κατηγορία.
Για την κατηγοριοποιηση χρησημοποιηθηκε η εξης διαδικασια :
1.Φορτωση του αρχειου 20 Newsgroups και των επιθυμητων κατηγοριων
2.Εξαγωγη του διανυσματος χαρακτηριστικων
3. «Εκπαίδευση» του ταξινομιτή
4.Προβλεψη της κατηγορίας μέσω συγκρισης

Αρχικά, για την επιτευξη της Κατηγοριοποίησης, χρησημοποιήθηκε το module scikit-learn και το 20 Newsgroups data set. Το συγκεκριμένο αρχειο δεδομενων, είναι μια συλλογη από 20.000 έγγραφα το οποία ανήκουν σε 20 διαφορετικές κατηγορίες.
Το πρωτο βημα είναι να ορίσουμε ποιες από τις 20 κατηγοριες μας ενδιαφερουν. Στην δικη μας υλοποιηση, χρσημοποιησαμε και τις 20. Στη συνεχεια, «φορτωνουμε» τα αρχεια τα οποια αντιστοιχουν στις επιλγμενες κατηγοριες.
Στο δευτερο βημα γινεται η μετατροπη των αρχειων σε διανυσματα χαρακτηριστικων.
Για να επιτευχθει αυτό, χρησημοποιησαμε την εξης αναπαρασταση :
Δινεται ένα ID σε κάθε λεξη που εμφανιζεται σε καποιο αρχειο και δημιουργειται ένα λεξικο. Αρα, δημιουργειται ένα λεξικο οπου για κάθε αρχειο μετράτε το πληθος της κάθε λεξης και αποθηκευεται στην μορφη X[i, j], οπου i είναι το κάθε αρχειο και j το index της λεξης στο λεξικο.
Επειτα, υπολογιζοντας το tf-idf(Term Frequency times Inverse Document Frequency), βρισκουμε το «βαρος» μιας λεξης, δηλαδη το ποσο σημαντικη είναι και ποσο συχνα εμφανιζεται(συχνοτητα). Με αυτόν τον τροπο «εξαγουμε» χαρακτηριστικα για το κάθε κειμενο.
Το επομενο βημα είναι, χρησημοποιώντας τα χαρακτηριστικα, να κατηγοριοποιησουμε το κάθε κειμενο από το πακετο 20 News Group. Αυτό γίνεται «εκπαιδευοντας» τον ταξηνομητη “cosine” και Jaccard.
Τέλος, παιρνουμε το κείμενο από την δικη μας συλλογη και εξαγουμε τον διανυσμα χαρακτηριστικων και υπολογιζουμε το tfidf. Επειτα, συγκρινουμε τα αποτελεσματα με τον παραπανω ταξινομητη και εξαγουμε την κατηγορια στην οποια ανηκει το κειμενο.

Αποτελέσματα Jaccard:

Classifying: article4.txt
Category: talk.religion.misc Jaccard similarity: 0.0428713858425
Category: talk.politics.mideast Jaccard similarity: 0.0420792079208
Category: talk.politics.guns Jaccard similarity: 0.0412844036697
Category: talk.politics.mideast Jaccard similarity: 0.0399673735726
Category: talk.politics.mideast Jaccard similarity: 0.0379562043796
****

Classifying: article1.txt
Category: sci.space Jaccard similarity: 0.0377906976744

Category: talk.politics.misc Jaccard similarity: 0.0353143841516
Category: talk.politics.misc Jaccard similarity: 0.0341013824885
Category: comp.sys.mac.hardware Jaccard similarity: 0.0340425531915
Category: talk.politics.mideast Jaccard similarity: 0.0334412081985
****

Classifying: article2.txt
Category: soc.religion.christian Jaccard similarity: 0.0583941605839
Category: talk.politics.misc Jaccard similarity: 0.0515055467512
Category: talk.politics.guns Jaccard similarity: 0.0514096185738
Category: sci.space Jaccard similarity: 0.0513728963685
Category: talk.religion.misc Jaccard similarity: 0.0507462686567
****

Classifying: article3.txt
Category: talk.politics.mideast Jaccard similarity: 0.043661971831
Category: talk.politics.mideast Jaccard similarity: 0.0434332988625
Category: talk.politics.mideast Jaccard similarity: 0.0422993492408
Category: talk.politics.mideast Jaccard similarity: 0.0393013100437
Category: talk.politics.mideast Jaccard similarity: 0.0392772977219
****

Classifying: article5.txt
Category: talk.politics.mideast Jaccard similarity: 0.0547195622435
Category: soc.religion.christian Jaccard similarity: 0.0542279411765
Category: talk.politics.guns Jaccard similarity: 0.0510887772194
Category: talk.politics.mideast Jaccard similarity: 0.0486533449175
Category: talk.politics.mideast Jaccard similarity: 0.04670558799
****

Classifying: article6.txt
Category: talk.politics.guns Jaccard similarity: 0.0290178571429
Category: soc.religion.christian Jaccard similarity: 0.0281081081081
Category: rec.sport.hockey Jaccard similarity: 0.0261519302615
Category: talk.religion.misc Jaccard similarity: 0.0253699788584
Category: talk.politics.mideast Jaccard similarity: 0.0252918287938
****

Classifying: article7.txt
Category: soc.religion.christian Jaccard similarity: 0.0422680412371
Category: rec.motorcycles Jaccard similarity: 0.0401106500692
Category: rec.autos Jaccard similarity: 0.0399449035813
Category: rec.autos Jaccard similarity: 0.039751552795
Category: rec.autos Jaccard similarity: 0.0397219463754
****

Classifying: article8.txt
Category: sci.med Jaccard similarity: 0.0643340857788
Category: rec.autos Jaccard similarity: 0.0531594784353
Category: talk.politics.mideast Jaccard similarity: 0.0522284122563
Category: talk.politics.guns Jaccard similarity: 0.0520146520147
Category: sci.med Jaccard similarity: 0.0507685142059
****

Αποτελέσματα Cosine:

Classifying: article4.txt
Category: rec.autos Cosine similarity: 0.104620446454
Category: rec.autos Cosine similarity: 0.103523341327
Category: talk.politics.guns Cosine similarity: 0.0873300300039
Category: talk.politics.mideast Cosine similarity: 0.0823367487542
****

Classifying: article1.txt
Category: talk.politics.mideast Cosine similarity: 0.0892223110446
Category: talk.politics.misc Cosine similarity: 0.0862622919428
Category: talk.politics.misc Cosine similarity: 0.0859321765024
Category: rec.motorcycles Cosine similarity: 0.0829821351185
****

Classifying: article2.txt
Category: rec.autos Cosine similarity: 0.0971391255758
Category: talk.politics.misc Cosine similarity: 0.0948365760168
Category: talk.politics.mideast Cosine similarity: 0.0904674671259
Category: misc.forsale Cosine similarity: 0.089146863413
****

Classifying: article3.txt
Category: soc.religion.christian Cosine similarity: 0.158204009848
Category: soc.religion.christian Cosine similarity: 0.129564508605
Category: soc.religion.christian Cosine similarity: 0.123714224045
Category: talk.religion.misc Cosine similarity: 0.105412988865
****

Classifying: article5.txt
Category: sci.med Cosine similarity: 0.1014662194
Category: sci.med Cosine similarity: 0.096719665942
Category: soc.religion.christian Cosine similarity: 0.0952048134642
Category: sci.med Cosine similarity: 0.094857217166
****

Classifying: article6.txt
Category: comp.windows.x Cosine similarity: 0.0934564961459
Category: comp.os.ms-windows.misc Cosine similarity: 0.0837359326021
Category: soc.religion.christian Cosine similarity: 0.082533644079
Category: sci.space Cosine similarity: 0.081600298591
****

Classifying: article7.txt
Category: sci.electronics Cosine similarity: 0.1125471981
Category: comp.os.ms-windows.misc Cosine similarity: 0.111729255419
Category: sci.crypt Cosine similarity: 0.102774488136
Category: sci.crypt Cosine similarity: 0.0962173842789
****

Classifying: article8.txt
Category: sci.med Cosine similarity: 0.216427879635
Category: sci.med Cosine similarity: 0.173681167459
Category: sci.med Cosine similarity: 0.161179899575
Category: sci.med Cosine similarity: 0.157707439508
****

Έπειτα από διερεύνηση των αποτελεσμάτων, καταλήξαμε στο συμπέρασμα ότι η μετρική Jaccard είναι αρκετά πιο γρήγορα από την Cosine, αλλά η cosine, παράγει πιο αξιόπιστα αποτελέσματα.

Επίσης, η μετρική Cosine, υλοποίηθηκε κυρίως με την χρήση του πακέτου SciKit, ενώ η Jaccard, υλοποιήθηκε από την αρχή χωρίς, δηλαδή, την χρήση κάποιου επιπλέον module.