

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ · ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ




ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ

PROJECT 2012-2013 · ΟΜΑΔΑ Α22 · 4465 · 4646 · 4651

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΥΠΟΛΟΓΙΣΤΗ & ΛΟΓΙΣΜΙΚΟΥ

| | |
|--------------|----------------------------------|
| Επεξεργαστής | Intel Core 2 Quad @ 2.40GHz |
| Μητρική | GIGABYTE G41M |
| Μνήμη RAM | 4GB |
| Λειτουργικό | Windows 7 Professional SP1 (x64) |
| Πρόγραμμα | IDLE Python 2.7.5 (για x86) |



Για την εκπόνηση της παρούσας εργασίας χρησιμοποιήθηκε η γλώσσα **Python** και κυρίως οι βιβλιοθήκες **NLTK**, **SciPy** και **NumPy** (για τους μαθηματικούς υπολογισμούς).

ΣΤΑΤΙΣΤΙΚΑ ΕΚΤΕΛΕΣΗΣ

| | | | |
|--|-------------|-------------|-------------|
| Μέγεθος χώρου S | 3000 | 4000 | 5000 |
| Αριθμός αρχείων στη συλλογή A | 3169 | 3169 | 3169 |
| Αριθμός αρχείων στη συλλογή E | 4762 | 4762 | 4762 |
| Χρόνος αρχικοποίησης | 0,125 sec | 0,229 sec | 0,062 sec |
| Χρόνος tokenization, stemming & TF-IDF για E | 37,799 sec | 56,912 sec | 38,235 sec |
| Χρόνος tokenization, stemming & TF-IDF για A | 22,089 sec | 34,133 sec | 22,463 sec |
| Χρόνος δημιουργίας χώρου S & διανυσμάτων | 36,722 sec | 50,197 sec | 58,095 sec |
| Χρόνος κατηγοριοποίησης με cosine similarity | 3,884 sec | 4,414 sec | 4,275 sec |
| Χρόνος κατηγοριοποίησης με Tanimoto similarity | 682,578 sec | 680,161 sec | 682,470 sec |
| Συνολικός χρόνος εκτέλεσης | 783,231 sec | 826,103 sec | 805,664 sec |
| Ακρίβεια με cosine similarity | 86,05% | 87,28% | 88,16% |
| Ακρίβεια με Tanimoto similarity | 86,62% | 86,62% | 86,62% |

Να σημειωθεί ότι έγιναν δοκιμαστικές εκτελέσεις για μέγεθος χώρου S ίσο με 5500 και 6000 με float16 ως τύπο δεδομένων για τα *lil_matrix*, και προέκυπτε σφάλμα μνήμης κατά την εκτέλεση. Αντίστοιχα και με τύπο δεδομένων float32 και float, και αυτός ήταν ο λόγος της επιλογής του δεκαδικού μισής ακρίβειας. Παρά τη λιγότερη πληροφορία, τα αποτελέσματα συγκριτικά με τον float32 για μέγεθος χώρου S 4000 ήταν παραπλήσια, τόσο ως προς την ακρίβεια όσο και ως προς το χρόνο εκτέλεσης.

ΕΠΕΞΗΓΗΣΗ ΔΟΜΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στο σημείο αυτό επεξηγούνται τα στάδια του προγράμματος που υλοποιεί τα ζητούμενα της εκφώνησης. Περισσότερες και παρουσιασμένες με πιο συγκεκριμένο τρόπο πληροφορίες υπάρχουν στον κώδικα του προγράμματος, με τη μορφή σχολίων.

ΣΤΑΔΙΟ #1 - ΑΡΧΙΚΟΠΟΙΗΣΗ

Στο πρώτο στάδιο καθορίζονται οι φάκελοι του υπολογιστή απ' όπου το πρόγραμμα θα διαβάζει τα έγγραφα που απαιτούνται, καθώς επίσης γίνεται και καταγραφή της κατηγορίας στην οποία ανήκουν. Αρχικοποιούνται ακόμα μεταβλητές που αφορούν το εργαλείο για το stemming (βλ. Στάδιο #2) και το μέγεθος του χώρου χαρακτηριστικών.

ΣΤΑΔΙΟ #2 - TOKENIZATION, STEMMING & TF-IDF ΓΙΑ ΤΗ ΣΥΛΛΟΓΗ E

Στο δεύτερο στάδιο γίνεται ο διαχωρισμός των λεκτικών μονάδων κάθε κειμένου της συλλογής E (tokenization) και η θεματοποίησή τους με την απομάκρυνση της κατάληξης (stemming). Αυτό το στάδιο αφορά το στάδιο της προεπεξεργασίας της συλλογής E καθώς και του υπολογισμού του TF-IDF για όλα τα θέματά της. Να σημειώσουμε ότι η συλλογή E εξετάζεται ολόκληρη χωρίς να υπάρχει μερική επεξεργασία του `rec.train` ή του `sci.train` φακέλου της.

ΣΤΑΔΙΟ #3 - TOKENIZATION, STEMMING & TF-IDF ΓΙΑ ΤΗ ΣΥΛΛΟΓΗ A

Στο τρίτο στάδιο γίνεται ο διαχωρισμός των λεκτικών μονάδων κάθε κειμένου της συλλογής A, η θεματοποίηση και ο υπολογισμός του TF-IDF για τα θέματά της αντίστοιχα.

ΣΤΑΔΙΟ #4 - ΤΑΞΙΝΟΜΗΣΗ, ΚΑΤΑΣΚΕΥΗ ΧΩΡΟΥ S & ΔΙΑΝΥΣΜΑΤΩΝ

Στο τέταρτο στάδιο γίνεται πρώτα μια ταξινόμηση (από το μεγαλύτερο στο μικρότερο) στο λεξικό που περιέχει τα αποτελέσματα του TF-IDF για τη συλλογή E, και επιλέγονται έπειτα οι πρώτοι 4000 (ή 6000, κτλ.) όροι. Αυτοί σχηματίζουν το ζητούμενο χώρο χαρακτηριστικών S. Στη συνέχεια, δημιουργούμε ένα διάνυσμα για κάθε έγγραφο κάθε συλλογής, αρχίζοντας από αυτά της E και μετά της A.

ΣΤΑΔΙΟ #5 - ΥΠΟΛΟΓΙΣΜΟΣ & ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΜΕ ΜΕΘΟΔΟ ΣΥΝΗΜΙΤΟΝΟΥ

Στο πέμπτο στάδιο γίνεται ο υπολογισμός της ομοιότητας μεταξύ εγγράφων χρησιμοποιώντας τη μετρική του συνημιτόνου (cosine similarity). Έτσι, αξιοποιώντας τη δομή των `sparse` μητρώων μας και σύμφωνα με τον αντίστοιχο τύπο της μετρικής, υπολογίζουμε τις τιμές της μετρικής, και έπειτα εντοπίζουμε τις μέγιστες από αυτές. Με τον εντοπισμό αυτό κατηγοριοποιούμε το κάθε έγγραφο της συλλογής A σύμφωνα με το έγγραφο της συλλογής E που μοιάζει περισσότερο. Ακόμα, εκτυπώνουμε το αποτέλεσμα σε ένα έγγραφο σε περίπτωση ελέγχου μετά το πέρας του προγράμματος, ενώ στο τέλος κρατάμε και στατιστικά για τον υπολογισμό της ακρίβειας, μετρώντας πόσα έγγραφα κατηγοριοποιήθηκαν στην κατηγορία για την οποία προορίζονταν.

ΣΤΑΔΙΟ #6 - ΥΠΟΛΟΓΙΣΜΟΣ & ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΜΕ ΜΕΘΟΔΟ TANIMOTO

Στο έκτο στάδιο γίνεται ο υπολογισμός της ομοιότητας μεταξύ εγγράφων χρησιμοποιώντας τη μετρική του Tanimoto (Tanimoto similarity). Για τον υπολογισμό της μετρικής αυτής δε μπορούμε να εκμεταλλευτούμε περισσότερο τα sparse μητρώα που έχουμε, γι' αυτό και είναι το πιο χρονοβόρο κομμάτι του προγράμματός μας. Έτσι, έχουμε δύο βρόχους, τον ένα εμφωλευμένο στον άλλο, όπου για κάθε έγγραφο της συλλογής A υπολογίζουμε τις μετρικές για όλα τα έγγραφα της E, και εντοπίζοντας τη μέγιστη, κατηγοριοποιούμε το έγγραφο στην αντίστοιχη κατηγορία. Και εδώ εκτυπώνουμε το αποτέλεσμα σε ένα έγγραφο σε περίπτωση ελέγχου μετά το πέρας του προγράμματος, ενώ στο τέλος κρατάμε πάλι στατιστικά για τον υπολογισμό της ακρίβειας.

ΣΤΑΔΙΟ #7 - ΕΛΕΓΧΟΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ & ΣΤΑΤΙΣΤΙΚΑ

Στο τελευταίο στάδιο υπολογίζουμε κάποια στατιστικά που θα μας δείξουν κατά πόσο η κατηγοριοποίηση των εγγράφων ήταν επιτυχής. Πιο συγκεκριμένα, υπολογίζουμε της ακρίβειας για τις δύο μεθόδους και εκτυπώνουμε τα αντίστοιχα αποτελέσματα μαζί με το συνολικό χρόνο.