ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

Ανάκτηση Πληροφοριών

Εργασία Υλοποίησης

Γεωργιάδης Άγγελος E12029

Περιγραφή μηχανής αναζήτησης “search_engine”

Η μηχανή αναζήτησης έχει σκοπό την ανάκτηση και αναζήτηση λέξεων ή φράσεων μέσα σε αρχεία της μορφής .txt από ερωτήματα που εισάγει ο χρήστης και την επιστροφή των αποτελεσμάτων.

Η μηχανή αναζήτησης σχεδιάστηκε μέσω της Python 3.5.

Όπως αναφέρεται και στην περιγραφή της εργασίας , η μηχανή αναζήτησης θα πρέπει να δέχεται ως είσοδο εναν φάκελο (που ενδεχομένως περιέχει και υποφακέλους) με αρχεία .txt .Εδώ η μηχανή αναζήτησης δεν δέχεται ως είσοδο τον φάκελο,αλλά τοποθετείται μεσα σε αυτόν και εκτελείται εκεί.

Αρχικά έχουμε τον πηγαίο κώδικα “search_engine.py” το οποίο απαιτεί την Python 3.5 για να εκτελεστεί(δεν δοκίμασα νεότερες εκδόσεις οπότε δεν γνωρίζω εάν θα λειτουργεί σε Python 2),και εκτελείται είτε μεσω command prompt είτε ανοίγωντάς το.

Επίσης , εάν ο χρήστης έχει διαφορετική έκδοση της Python ή δεν έχει καμία έκδοση της Python ή αντιμετωπίζει άλλα προβλήματα στην εκτέλεσή του , υπάρχει το αρχείο “search_engine.exe” το οποίο είναι η εκτελέσιμη μορφή του αρχείου “search_engine.py” .Το αρχείο “search_engine.exe” , για να εκτελεστεί σωστά, πρέπει να βρίσκεται στο φάκελο που θέλουμε να κάνουμε την αναζήτηση (όμοια με το “search_engine.py”).

Για την μετατροπή αυτή χρησιμοποιήθηκε το PyInstaller το οποίο μετατρέπει python scripts σε εκτελέσιμη μορφή .exe .

Επεξήγηση αρχείων και φακέλων

Το αρχείο 12029_Γεωργιάδης-Άγγελος.zip περιέχει τρεις φακέλους :

- “exe” folder
- “src” folder
- “text files” folder

“exe” folder : Περιέχει τα αρχεία του PyInstaller που δημιούργηθηκαν μετά την μετατροπή του “search_engine.py” σε “search_engine.exe”.Για την εκτέλεση του χρειαζόμαστε μόνο το αρχείο “search_engine.exe” που βρίσκεται στον φάκελο “dist”.

“src” folder : Περιέχει τον πηγαίο κώδικα “search_engine.py”

“text files” folder : Περιέχει μία συλλογή απο txt files τα οποία χρησιμοποιήθηκαν για τον έλεγχο της σωστης λειτουργίας της μηχανής.Μέσα στον φάκελο υπάρχουν και τρεις ενδεικτικοί υποφάκελοι οι οποίοι περιέχουν και αυτοί είτε αρχεία txt είτε υποφακέλους με αρχεία.Τα αρχεία αυτά τα βρήκα στο site http://textfiles.com/directory.html στην κατηγορία “Stories”.

Ορισμένα από τα αρχεία δεν χρησιμοποιήθηκαν γιατι δεν ήταν σε μορφή .txt και κάποια αρχεία είχαν χαρακτήρες που δεν αναγνωρίζονταν απο τον parser.

Υπάρχει επίσης το αρχείο “README.txt” στο οποίο υπάρχουν οδηγίες για την εκτέλεση του κώδικα με διαφορετικούς τρόπους.

Modules μηχανής

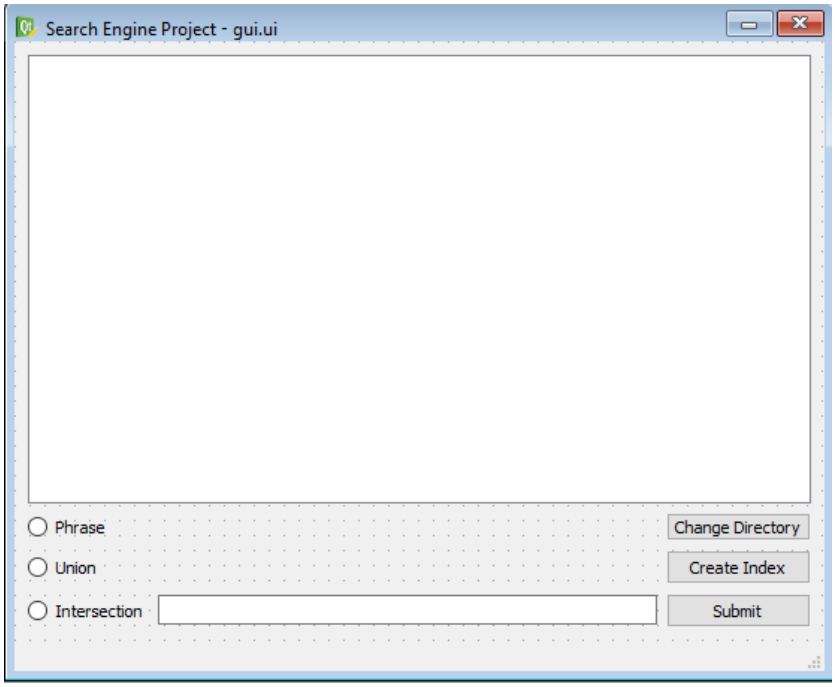
Βασικά Modules :

- Parser εγγράφων κειμένων που να υλοποιεί διαίρεση σε σύμβολα και παράγει όρους (tokens) που θα τοποθετηθούν στο λεξικό του ευρετηρίου
- Απλό αντεστραμμένο ευρετήριο (λεξικό και λίστες καταχώρησης)
- Απλό αλγόριθμο ανάκτησης εγγράφων που να υποστηρίζει κάποια ερωτήματα (π.χ. τουλάχιστον ανάκτηση Boole)
- Query parser που παίρνει ως είσοδο ένα ερώτημα σε κάποια μορφή (π.χ. έκφραση Boole ή ερώτημα ελεύθερου κειμένου) και το κατευθύνει στο κατάλληλο ευρετήριο
- Εμφάνιση των αποτελεσμάτων της αναζήτησης στην οθόνη

Προαιρετικά Modules :

- Πιο προηγμένοι τύποι ευρετηρίων: ευρετήρια θέσης, ευρετήρια ζευγών λέξεων, ευρετήρια φράσεων
- Υλοποίηση γραφικής διεπαφής χρήστη*

* Η γραφική διεπαφή υπάρχει στον φάκελο “gui” όμως δεν μπορούσα να την χρησιμοποιήσω διότι δεν αναγνωρίζονταν κάποια dll αρχεία απο την python.(Και δεν πρόλαβα να βρω κάποια λύση)
Ο σχεδιασμός του GUI έγινε μέσω του QtDesigner και η μετατροπή του αρχείου “gui.ui” σε “gui.py” μέσω του PyQt4.
Η επιθυμητή μορφή που ήθελα να πάρει η διεπαφή φαίνεται στην παρακάτω εικόνα :



Τεκμηρίωση Κώδικα

Το πρόγραμμα ξεκινάει εμφανίζοντας στον χρήστη το directory στο οποίο βρίσκονται τα txt αρχεία και το εκτελέσιμο αρχείο ,και αρχίζει να επεξεργάζεται τα αρχεία με τη σειρά . Δημιουργεί το αντεστραμένο ευρετήριο και μπαίνει σε ένα while loop το οποίο σταματάει όταν ο χρήστης δεν δώσει κάποιο ερώτημα προς αναζήτηση(δηλαδή αν πατήσει απλά Enter").

Στην επανάληψη αυτή αρχικά ζητείται από τον χρήστη το ερώτημα(query) ,το οποίο ,ανάλογα την μορφή του,εντάσσεται σε μια κατηγορία αναζήτησης.

Εαν το ερώτημα περιέχει μία μόνο λέξη γίνεται αναζήτηση μέσω της "one_word_query" συνάρτησης , η οποία αναζητεί την ύπαρξη της λέξης στο ευρετήριο.Αν η λέξη δεν υπάρχει σε κανένα αρχείο τότε εμφανίζεται το μήνυμα "The word '"+query+"' was not found in any file.\n" . Αλλιώς αν υπάρχει η λέξη σε κάποια απο τα αρχεία, τοτε εμφανίζονται τα ονόματα των αρχείων που την περιέχουν.

Αν το ερώτημα περιέχει παραπάνω απο μία λέξη,ο χρήστης καλείται να απαντήσει εάν το ερώτημα που εισήγαγε είναι φράση.Αν ειναι φράση(δηλαδή αν απαντήσει "y"), τότε γίνεται αναζήτηση μεσω της συνάρτησης "phrase_query" , η οποία αναζητεί ολόκληρη τη φράση δηλαδή τις λέξεις που δόθηκαν,με τη σειρά που δόθηκαν.

Τέλος,αν το ερώτημα δεν είναι ούτε λέξη ούτε φράση,τότε παρουσιάζονται δύο σύνολα αρχείων .Το πρώτο είναι η ένωση(union) των αρχείων,όπου το κάθε αρχείο περιέχει μία τουλάχιστον λέξη του ερωτήματος.Το δεύτερο σύνολο είναι η τομή(intersection) των αρχείων,όπου το κάθε αρχείο περιέχει όλες τις λέξεις.Οι αναζητήσεις γίνονται με τις συναρτήσεις "free_text_query_union" και "free_text_query_intersection" αντίστοιχα.

Πηγές

Μεγάλο μέρος του κώδικα το πήρα από : *https://github.com/logicx24/Text-Search-Engine*

Γενικές Πληροφορίες για python : *https://docs.python.org*

Γενικές απορίες για python : *http://stackoverflow.com/*

Γενικές Πληροφορίες για python : *http://www.tutorialspoint.com/python3/*

Γενικές Πληροφορίες για python : *https://wiki.python.org*

Πηγή των text files που χρησιμοποιήθηκαν : *http://textfiles.com/stories/*

Μετατροπή .py σε .exe : *http://stackoverflow.com/questions/33168229/how-to-create-standalone-executable-file-from-python-3-5-scripts*

Σχεδιασμός γραφικής διεπαφής : *https://www.youtube.com/watch?v=GLqrzLIIW2E*