



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Σύστημα Γεωγραφικού Εντοπισμού Ιδρύματος Ή
Εταιρείας



Της φοιτήτριας
Θάλεια Μότσιου

Επιβλέπων
Αντώνης Σιδηρόπουλος

Αρ. Μητρώου: 113754

Θεσσαλονίκη 2021

Τίτλος Δ.Ε.: Σύστημα Γεωγραφικού Εντοπισμού Ιδρύματος ή Εταιρίας

Κωδικός Δ.Ε.: 19006

Ονοματεπώνυμο φοιτήτριας: Ευθαλία Μότσιου

Ονοματεπώνυμο εισηγητή: Αντώνης Σιδηρόπουλος

Ημερομηνία ανάληψης Δ.Ε.: 07-04-2019

Ημερομηνία περάτωσης Δ.Ε.: 14-01-21

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Ευθαλία Μότσιου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

1. Πρόλογος

Το παρόν κείμενο αποτελεί το γραπτό κομμάτι της πτυχιακής εργασίας με τίτλο Σύστημα Γεωγραφικού Εντοπισμού Ιδρύματος Ή Εταιρείας.

Η συγκεκριμένη πτυχιακή εργασία στοχεύει στο σχεδιασμό και την ανάπτυξη μιας εφαρμογής ειδικού ενδιαφέροντος, η οποία θα προσφέρει στους χρήστες πληροφορίες σχετικά με την γεωγραφική θέση εκπαιδευτικών ιδρυμάτων και εταιρειών ανά τον κόσμο . Θα μελετηθούν τα βήματα σχεδιασμού της εφαρμογής, όπως για παράδειγμα οι πρωταρχικές ενέργειες για την επιλογή των κατάλληλων εργαλείων αλλά και η ανάλυση του συστήματος.

2. Περίληψη

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργία ενός συστήματος εντοπισμού της γεωγραφικής θέσης εκπαιδευτικών ιδρυμάτων και εταιρειών απ' όλο τον κόσμο.

Το σύστημα που θα αναλυθεί και αναπτυχθεί θα επιτρέπει στους χρήστες να εντοπίζουν την θέση (χώρα/περιοχή/πόλη) ενός συνόλου ιδρυμάτων που θα διαβάζονται από ένα αρχείο κειμένου. Εφόσον βρεθεί η θέση του ιδρύματος θα αποθηκεύεται σε ένα άλλο αρχείο κειμένου το οποίο ο χρήστης θα μπορεί να διαβάσει και να αναλύσει όπως επιθυμεί. Η εφαρμογή θα εκτελείται μέσω της γραμμής εντολών.

Η πτυχιακή εργασία αρχικά πραγματοποιεί μια ανάλυση της διαδικασίας του Geocoding. Στη συνέχεια, παρουσιάζεται μια ιστορική αναδρομή σχετικά με την εξέλιξη του Geocoding στο πέρασμα του χρόνου. Έπειτα, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία του συστήματος. Επιπρόσθετα, θα αναλυθούν διεξοδικά όλα τα APIs που παρέχουν υπηρεσίες Geocoding και χρησιμοποιήθηκαν για τον εντοπισμό των θέσεων. Τέλος, η εργασία ολοκληρώνεται με την παράθεση τεχνικών λεπτομερειών υλοποίησης καθώς και του οδηγού χρήσης.

Institution or Company Geographical Location System

Thalia Motsiou

3. Abstract

The purpose of this dissertation is to create a system for finding the geographical location of educational institutions and companies from around the world.

The system to be analyzed and developed will allow users to locate the country, region, and city of a set of institutions that will be read from a text file. Once the location of the institution is found it will be saved in another text file, which the user will be able to read and analyze as he wishes. This application will be executed from the command line interface.

The dissertation initially performs an analysis of the Geocoding process. Following that is a historical overview of the evolution of Geocoding over time. Next, the technologies used to create the system are presented. In addition, all the APIs that provide Geocoding services and were used to locate the sites will be analyzed in detail. Finally, the dissertation is completed with the dissertation of technical details of implementation as well as the user guide.

4. Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή Αντώνη Σιδηρόπουλο για την εξαιρετική συνεργασία και την πολύτιμη συμβολή της στην ολοκλήρωση αυτής.

Θέλω επίσης να ευχαριστήσω όλους του δικούς μου ανθρώπους που μου στάθηκαν αυτό το διάστημα, με στήριξαν με υπομονή και κατανόηση κατά την διάρκεια της φοίτησής μου και με βοήθησαν να πετύχω τους στόχους μου.

5. Περιεχόμενα

Πίνακας περιεχομένων

1. Πρόλογος.....	1
2. Περίληψη.....	2
3. Abstract	3
4. Ευχαριστίες	4
5. Περιεχόμενα	5
6. Κατάλογος Σχημάτων	9
7. Κατάλογος Πινάκων.....	10
Κεφάλαιο 1ο: Το Geocoding	11
1.1 Εισαγωγή.....	11
1.2 Ιστορία.....	11
1.2.1 Δεκαετία ‘60.....	11
1.2.2 Δεκαετία ‘70.....	11
1.2.3 Δεκαετία ‘90.....	12
1.2.4 Δεκαετία ‘80.....	12
1.2.5 Δεκαετία ‘00.....	12
1.2.6 Δεκαετία ‘10.....	13
Κεφάλαιο 2ο: Geocoding APIs που χρησιμοποιήθηκαν	14
2.1 Εισαγωγή.....	14
2.2 MapQuest	14
2.2.1 Authentication	14
2.2.2 Περιορισμοί.....	14
2.3 Here	15
2.3.1 Authentication	15
2.3.2 Περιορισμοί.....	15
2.4 PositionStack	15
2.4.1 Authentication	15
2.4.2 Περιορισμοί.....	16
2.5 PickPoint	16
2.5.1 Authentication	16
2.5.2 Περιορισμοί.....	16

2.6	Nominatim.....	16
2.6.1	Authentication	16
2.6.2	Περιορισμοί.....	17
2.7	MapBox.....	17
2.7.1	Authentication	17
2.7.2	Περιορισμοί.....	17
Κεφάλαιο 3ο: Η γλώσσα προγραμματισμού Python.....		18
3.1	Εισαγωγή.....	18
3.2	Εισαγωγή στην Python.....	18
3.3	Ιστορία της Python.....	18
3.4	Φιλοσοφία της Python.....	19
3.5	Εγκατάσταση Python.....	20
3.5.1	Windows.....	20
3.5.2	Linux	20
3.5.3	Mac OS X.....	20
3.6	Λόγος επιλογής Python	20
Κεφάλαιο 4ο: Υλοποίηση πτυχιακής εργασίας		21
4.1	Εισαγωγή.....	21
4.2	PyCharm.....	21
4.3	Βιβλιοθήκη geopy	21
4.3.1	Εγκατάσταση.....	21
4.3.2	Τρόπος χρήσης	22
4.4	Βιβλιοθήκη langdetect.....	22
4.4.1	Τρόπος χρήσης	22
4.5	Βιβλιοθήκη autocorrect.....	23
4.5.1	Τρόπος χρήσης	23
4.5.2	Χρόνος εκτέλεσης	23
4.6	Βιβλιοθήκη spellcheck	24
4.6.1	Τρόπος χρήσης	25
4.6.2	Χρόνος εκτέλεσης	25
4.7	Βιβλιοθήκη requests.....	26
4.7.1	Τρόπος χρήσης	26
4.8	Πιθανά Exceptions	26
4.9	Επίλογος.....	27
Κεφάλαιο 5ο: Σύγκριση αποτελεσμάτων αναζήτησης.....		28

5.1	Εισαγωγή.....	28
5.2	Περίπτωση χρήσης #1.....	28
5.3	Περίπτωση χρήσης #2.....	29
5.4	Περίπτωση χρήσης #3.....	30
5.5	Περίπτωση χρήσης #4.....	30
5.6	Περίπτωση χρήσης #5.....	32
5.7	Επίλογος.....	34
Κεφάλαιο 6ο: Σειρά χρήσης APIs.....		35
6.1	Εισαγωγή.....	35
6.2	Δείγμα δεδομένων.....	35
6.3	Ανάλυση αποτελεσμάτων.....	35
6.3.1	Αποτελέσματα MapQuest.....	35
6.3.2	Αποτελέσματα Here.....	35
6.3.3	Αποτελέσματα PositionStack.....	36
6.3.4	Αποτελέσματα PickPoint.....	36
6.3.5	Αποτελέσματα Nominatim.....	36
6.3.6	Αποτελέσματα MapBox.....	36
6.4	Συμπεράσματα.....	36
Κεφάλαιο 7ο: Οδηγός εγκατάστασης και χρήσης λογισμικού.....		38
7.1	Εισαγωγή.....	38
7.2	Οδηγός εγκατάστασης.....	38
7.2.1	Πιθανό σφάλμα κατά την εγκατάσταση των προϋποθέσεων.....	39
7.3	Οδηγός χρήσης.....	40
7.3.1	Δημιουργία και αποθήκευση API keys.....	40
7.3.2	Αρχείο δεδομένων.....	40
7.3.3	Εκτέλεση εφαρμογής.....	40
7.3.4	Ανάγνωση αποτελεσμάτων στο Excel.....	41
7.3.5	Ανάγνωση αποτελεσμάτων στο LibreOffice.....	42
Κεφάλαιο 8ο: Προτάσεις για μελλοντική βελτίωση.....		43
8.1.1	Γραφικό περιβάλλον.....	43
8.1.2	Απεικόνιση δεδομένων.....	43
Κεφάλαιο 9ο: Συμπεράσματα.....		44
8.	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	45
ΠΑΡΑΡΤΗΜΑ Α: ΔΕΙΓΜΑ ΔΕΔΟΜΕΝΩΝ.....		46
ΠΑΡΑΡΤΗΜΑ Β: ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ.....		48

6. Κατάλογος Σχημάτων

Σχήμα 2.1: Λογότυπο MapQuest

Σχήμα 2.2: Λογότυπο Here

Σχήμα 2.3: Λογότυπο Positionstack

Σχήμα 2.4: Λογότυπο PickPoint

Σχήμα 2.5: Λογότυπο Nominatim

Σχήμα 2.6: Λογότυπο MapBox

Σχήμα 3.1: Λογότυπο Python

Σχήμα 3.2: The Zen of Python

Σχήμα 4.1: Λογότυπο PyCharm

Σχήμα 4.2: Εκτέλεση εντολής `pip install geopy`

Σχήμα 4.3: Παράδειγμα χρήσης `geopy`

Σχήμα 4.4: Παράδειγμα χρήση βιβλιοθήκης `landdetect`

Σχήμα 4.5: Παράδειγμα χρήση βιβλιοθήκης `autocorrect`

Σχήμα 4.6: Χρόνος εκτέλεσης `autocorrect`

Σχήμα 4.7: Χρόνος γρήγορης εκτέλεσης `autocorrect`

Σχήμα 4.8: Παράδειγμα χρήσης `spellcheck`

Σχήμα 4.9: Χρόνος εκτέλεσης `spellcheck`

Σχήμα 4.10: Παράδειγμα χρήσης `requests`

Σχήμα 7.1: Σφάλμα εκτέλεσης

Σχήμα 7.2: Απαραίτητες βιβλιοθήκες

Σχήμα 7.3: Εκτέλεση παραδείγματος χωρίς σφάλμα

Σχήμα 7.4: Πιθανό σφάλμα εγκατάστασης Windows

Σχήμα 7.5: Εγκατάσταση C++ build tools

Σχήμα 7.6: Αρχείο `config.txt`

Σχήμα 7.7: Εκτέλεση εφαρμογής

Σχήμα 7.8: Άνοιγμα αποτελεσμάτων στο Excel

Σχήμα 7.9: Αποτελέσματα αναζήτησης

Σχήμα 7.10: Άνοιγμα αποτελεσμάτων στο LibreOffice

Σχήμα 9.1: Απεικόνιση αποτελεσμάτων σε χάρτη

7. Κατάλογος Πινάκων

Πίνακας 5.1: Περίπτωση χρήσης #1

Πίνακας 5.2: Περίπτωση χρήσης #2

Πίνακας 5.3: Περίπτωση χρήσης #3

Πίνακας 5.4: Περίπτωση χρήσης #4

Πίνακας 5.5: Περίπτωση χρήσης #4 στα αγγλικά

Πίνακας 5.6: Περίπτωση χρήσης #5 χωρίς ορθογραφικά

Πίνακας 5.7: Περίπτωση χρήσης #5 με ορθογραφικό λάθος (1)

Πίνακας 5.8: Περίπτωση χρήσης #5 με ορθογραφικό λάθος (2)

Πίνακας 6.1: Συγκεντρωτικά αποτελέσματα ανάλυσης APIs

Πίνακας A.1: Δείγμα δεδομένων 1-50

Πίνακας A.2: Δείγμα δεδομένων 51-100

Πίνακας B.1: Αποτελέσματα αναζήτησης

Κεφάλαιο 1ο: Το Geocoding

1.1 Εισαγωγή

Geocoding ονομάζεται η διαδικασία μετατροπής της περιγραφής μιας τοποθεσίας -όπως για παράδειγμα ενός ζεύγους συντεταγμένων, μια διεύθυνση ή το όνομα κάποιου σημείου- σε μια τοποθεσία στη γη. Μπορούμε να κάνουμε geocoding εισάγοντας μια περιγραφή τοποθεσίας κάθε φορά ή παρέχοντας πολλές τοποθεσίες ταυτόχρονα, αυτό εξαρτάται κυρίως από την υπηρεσία που χρησιμοποιείται, καθώς κάποια APIs επιτρέπουν την μαζική αναζήτηση ενώ άλλα όχι. Οι προσκόπτουσες τοποθεσίες εξάγονται ως γεωγραφικά χαρακτηριστικά, για παράδειγμα συντεταγμένες, διεύθυνση και ταχυδρομικοί κώδικες, τα οποία μπορούν να χρησιμοποιηθούν για χαρτογράφηση ή χωρική ανάλυση.

1.2 Ιστορία

Το geocoding - ένα υποσύνολο χωρικής ανάλυσης του Γεωγραφικού Συστήματος Πληροφοριών (ΓΣΠ)- αποτελεί αντικείμενο ενδιαφέροντος από τις αρχές της δεκαετίας του 1960.

1.2.1 Δεκαετία '60

Το 1960 το πρώτο λειτουργικό ΓΣΠ -που ονομάστηκε Canada Geographic Information System (CGIS) - εφευρέθηκε από τον Δρ. Roger Tomlinson, ο οποίος έκτοτε αναγνωρίζεται ως πατέρας του ΓΣΠ. Η μέθοδος του μετέτρεπε σχήματα χάρτη και εικόνες σε αριθμητική μορφή και μπορούσαν έτσι να συνδεθούν ψηφιακά μεταξύ τους για να ορίσουν γεωγραφικές περιοχές.

Το CGIS χρησιμοποιήθηκε για την αποθήκευση και την ανάλυση δεδομένων που συλλέχθηκαν για το Canada Land Inventory, το οποίο χαρτογράφησε πληροφορίες σχετικά με την γεωργία, την άγρια ζωή και τη δασοκομία σε κλίμακα 1:50000, με σκοπό να ρυθμιστούν οι δυνατότητες του εδάφους στον αγροτικό Καναδά. Το CGIS διήρκεσε έως την δεκαετία του 1990 και δεν ήταν ποτέ διαθέσιμο για εμπορική χρήση.

Το 1967, μια ομάδα αποφοίτων και φοιτητών του Yale, με επικεφαλής τον Donald Cooke (Yale, '67) - εφηύραν το Dual Independent Map Encoding (DIME) - για την απογραφή της πόλης New Haven, στο Connecticut των ΗΠΑ. Το μοντέλο που δημιούργησαν κωδικοποιούσε ένα εύρος από διευθύνσεις σε αρχεία οδικού δικτύου και αυτό επέτρεψε την χρήση του αλγορίθμου γεωγραφικής κωδικοποίησης “percent along” που εξακολουθεί να χρησιμοποιείται μέχρι και σήμερα, από πλατφόρμες χαρτών όπως οι Χάρτες Google και το MapQuest. Το New Haven ήταν η πρώτη πόλη στη Γη με βάση δεδομένων δικτύου οδών στα οποία μπορούσε να πραγματοποιηθεί γεωκωδικοποίηση.

1.2.2 Δεκαετία '70

Το πρώτο project ΓΣΠ του Γραφείου Απογραφής των Η.Π.Α ήταν το Urban Atlas το 1972. Δημιουργήθηκε για να δοκιμάσει την κατάταξη σε πίνακες των στατιστικών δεδομένων του 1970 χρησιμοποιώντας αρχεία GFB/DIME, σε συνεργασία με το Lawrence Berkeley Laboratory και το Harvard University's Computer Graphics and Spatial Analysis Laboratory. Η χρήση της ψηφιακής γεωχωρικής βάσης δεδομένων από το Γραφείο Απογραφής για γεωχωρική ανάλυση πάνω από 35.000 εγγραφών απογραφής ήταν επιτυχής.

1.2.3 Δεκαετία '90

Στα τέλη της δεκαετίας του 1990 το geocoding γίνεται προσανατολισμένο προς τον χρήστη, ειδικά μέσω ΓΣΠ λογισμικού ανοιχτού κώδικα. Οι εφαρμογές χαρτογράφησης και τα γεωχωρικά δεδομένα έχουν γίνει πιο προσιτά μέσω του Διαδικτύου.

Το Γραφείο Απογραφής των Η.Π.Α. μπόρεσε να συγκεντρώσει μια μεγάλη γεωχωρική βάση δεδομένων, χρησιμοποιώντας geocoding δρομων. Αυτή η βάση δεδομένων - μαζί με τα δεδομένα της απογραφής του 1980 για κάθε σπίτι στις Η.Π.Α - οδήγησε στη δημιουργία του TIGER (Topologically Integrated Geographic Encoding and Referencing).

Το TIGER -περιέχοντας εύρη διευθύνσεων αντί για μεμονωμένες διευθύνσεις- έχει έκτοτε εφαρμοστεί σχεδόν σε όλα τα λογισμικά geocoding που χρησιμοποιούνται σήμερα.

1.2.4 Δεκαετία '80

Στα τέλη της δεκαετίας του 1970 αναπτύχθηκαν δύο δημόσιες πλατφόρμες geocoding, οι GRASS GIS και MOSS. Στις αρχές της δεκαετίας του 1980 σημειώθηκε η άνοδος πολλών περισσότερων εμπορικών προμηθευτών λογισμικού geocoding όπως οι Intergraph, ESRI, CARIS, ERDAS και MapInfo Corporation. Αυτές οι πλατφόρμες συγχώνευσαν την προσέγγιση της δεκαετίας του 1960 για το διαχωρισμό των χωρικών πληροφοριών με την οργάνωση αυτών των χωρικών πληροφοριών σε δομές βάσεων δεδομένων.

Το 1986 το Mapping Display and Analysis System (MIDAS) έγινε το πρώτο desktop λογισμικό για geocoding, το οποίο σχεδιάστηκε για το λειτουργικό σύστημα DOS. Η εξαγορά του MIDAS από την MapInfo αποτέλεσε την αρχή για την αναβάθμιση του geocoding από τον ερευνητικό τομέα στον επιχειρηματικό κόσμο.

1.2.5 Δεκαετία '00

Στις αρχές της δεκαετίας του 2000 σημειώθηκε η άνοδος του Coding Accuracy Support System (CASS), ενός τρόπου προτυποποίησης διευθύνσεων. Η πιστοποίηση CASS προσφέρεται σε όλους τους προμηθευτές λογισμικού και τους διαφημιστές μέσω αλληλογραφίας που θέλουν οι Ταχυδρομικές Υπηρεσίες των Η.Π.Α να αξιολογήσουν την ποιότητα του λογισμικού τυποποίησης διευθύνσεων που χρησιμοποιούν.

Το 2003 οι πλατφόρμες geocoding μπορούσαν να συγχωνεύσουν ταχυδρομικούς κώδικες με δεδομένα δρόμων που ενημερώνονται κάθε μήνα. Αυτή η διαδικασία έγινε γνωστή ως "ενοποίηση".

Το 2006 οι πλατφόρμες geocoding εισήγαγαν το Αντίστροφο Geocoding, το οποίο επέτρεψε την γεωκωδικοποίηση της αριθμητικής θέσης ενός σημείου - με γεωγραφικό μήκος και πλάτος- σε μια ευανάγνωστη διεύθυνση.

Το 2008 και 2009 υπήρξε μια άνοδος σε διαδραστικές user-oriented πλατφόρμες geocoding όπως το Google Maps, MapQuest, Bing Maps κ.α. Αυτές οι πλατφόρμες έγιναν ακόμα πιο εύκολα προσβάσιμες στο ευρύ κοινό, καθώς παράλληλα αναπτύχθηκε και η βιομηχανία κινητής τηλεφωνίας και εισήλθε στις ζωές μας το smartphone.

1.2.6 Δεκαετία '10

Τη δεκαετία του 2010 το geocoding και το αντίστροφο geocoding υποστηρίζονται πλήρως από όλες τις εταιρείες παροχής υπηρεσιών geocoding. Τα cloud-based geocoding APIs επέτρεψαν μεγαλύτερο ρυθμό αντιστοίχισης, μεγαλύτερη ακρίβεια και μεγαλύτερη ταχύτητα.

Κεφάλαιο 2ο: Geocoding APIs που χρησιμοποιήθηκαν

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μια ανάλυση των APIs που χρησιμοποιήθηκαν για την εύρεση των γεωγραφικών θέσεων.

2.2 MapQuest



Σχήμα 2.1: Λογότυπο MapQuest

Η MapQuest είναι μια δωρεάν υπηρεσία χαρτογράφησης, η οποία κυκλοφόρησε το 1996 ως η πρώτη εμπορική διαδικτυακή υπηρεσία χαρτογράφησης

Πέρα από το Geocoding API που χρησιμοποιήθηκε στη παρούσα πτυχιακή εργασία, η MapQuest προσφέρει πολλά ακόμα APIs, όπως το Directions API, το Traffic API, το Place Search καθώς και mobile SDKs για συσκευές Android και iOS.

Το Geocoding API παρέχει την δυνατότητα Forward Geocoding, ή αλλιώς Geocoding διευθύνσεων, της διαδικασίας δηλαδή εύρεσης της γεωγραφικής θέσης με συντεταγμένες της δοθείσης διεύθυνσης. Παρέχει επίσης το Αντίστροφο Geocoding, την διαδικασία κατά την οποία βρίσκουμε την διεύθυνση ενός σημείου με βάση τις συντεταγμένες του. Τέλος, προσφέρει την δυνατότητα Batch Geocoding η οποία μας επιτρέπει να υποβάλουμε πολλαπλά αιτήματα Forward ή Αντίστροφου Geocoding την φορά. Το μέγεθος του batch ορίζεται σε 100 διευθύνσεις κάθε φορά.

2.2.1 Authentication

Για να μπορέσουμε να στείλουμε αιτήματα geocoding απαιτείται μια διαδικασία επαλήθευσης η οποία γίνεται με χρήση ενός API Key.

Το API Key μπορούμε να το εκδώσουμε από την επίσημη [ιστοσελίδα](#) της MapQuest αφού πρώτα δημιουργήσουμε έναν λογαριασμό χρήστη.

2.2.2 Περιορισμοί

Το MapQuest παρέχεται δωρεάν χωρίς καμία χρέωση, αυτό όμως έχει ως συνέπεια να υπάρχει ένα μηνιαίο όριο στον αριθμό των αιτημάτων που μπορούμε να στείλουμε. Αυτός ο αριθμός είναι 15000 αιτήματα ανά μήνα.

2.3 Here



Σχήμα 2.2: Λογότυπο Here

Η Here είναι μια Ολλανδική εταιρεία η οποία μεταξύ άλλων παρέχει υπηρεσίες χαρτογράφησης και τοποθεσίας σε μεμονωμένα άτομα ή επιχειρήσεις. Η Here έχει χάρτες για πάνω από 200 χώρες, προσφέρει φωνητική καθοδήγηση σε 94 χώρες και παρέχει δεδομένα της κίνησης στους δρόμους, σε πραγματικό χρόνο σε 33 χώρες.

Όπως και η MapQuest, η Here παρέχει την δυνατότητα Forward Geocoding, Αντίστροφου Geocoding και Batch Geocoding.

Για να υποβάλλουμε ένα Batch Geocoding αίτημα χρειάζεται να στείλουμε ένα αρχείο με τα δεδομένα μας. Το αρχείο αυτό δεν μπορεί να ξεπερνά τις 1,000,000 εγγραφές και το μέγιστο μέγεθος μπορεί να είναι έως 2GB.

2.3.1 Authentication

Για να μπορέσουμε να χρησιμοποιήσουμε το Here API απαιτείται μια διαδικασία επαλήθευσης, η οποία γίνεται με χρήση ενός API Key.

Το API Key μπορούμε να το εκδώσουμε μέσα από την επίσημη [ιστοσελίδα](#) της Here, αφού δημιουργήσουμε έναν λογαριασμό χρήστη.

2.3.2 Περιορισμοί

Η δωρεάν έκδοση του Here API παρέχει έως 250,000 αιτήματα το μήνα

2.4 PositionStack



Σχήμα 2.3: Λογότυπο Positionstack

Το Geocoding API της PositionStack δημιουργήθηκε για να προσφέρει μια δωρεάν και αξιόπιστη εναλλακτική ως λογισμικό geocoding. Η βάση δεδομένων τους περιλαμβάνει πάνω από 2 δισεκατομμύρια τοποθεσίες και διευθύνσεις παγκοσμίως.

Η PositionStack παρέχει την δυνατότητα Forward Geocoding, Αντίστροφου Geocoding καθώς και Batch Geocoding. Ο μέσος χρόνος απόκρισης στα αιτήματα που στέλνουμε κυμαίνεται από 10 ms σε 100 ms ανάλογα με το μέγεθος του request.

2.4.1 Authentication

Για να μπορέσουμε να χρησιμοποιήσουμε το PositionStack API απαιτείται μια διαδικασία επαλήθευσης, η οποία γίνεται με χρήση ενός API Key.

Το API Key μπορούμε να το εκδώσουμε μέσα από την επίσημη [ιστοσελίδα](#) της PositionStack, αφού δημιουργήσουμε έναν λογαριασμό χρήστη.

2.4.2 Περιορισμοί

Η δωρεάν έκδοση του PositionStack API έχει έναν περιορισμό στον αριθμό αιτημάτων που μπορούμε να κάνουμε μέσα σε έναν μήνα. Ο αριθμός αυτός είναι 25,000 αιτήματα. Ένας επιπλέον περιορισμός της δωρεάν έκδοσης είναι ότι δεν μας δίνεται η δυνατότητα να υποβάλλουμε batch requests.

2.5 PickPoint



Σχήμα 2.4: Λογότυπο PickPoint

Η PickPoint είναι μια εταιρεία με έδρα την Ρωσία. Η συλλογή των δεδομένων της γίνεται με τη βοήθεια πάνω από 500,000 ανθρώπων από όλο τον κόσμο οι οποίοι καθημερινά δημιουργούν καινούργια αντικείμενα ή ανανεώνουν τα ήδη υπάρχοντα, όπως δρόμους, κτήρια, περιοχές κ.α.

Η PickPoint παρέχει δυνατότητες Forward και Αντίστροφου Geocoding. Το Batch Geocoding δυστυχώς δεν είναι διαθέσιμο εδώ.

2.5.1 Authentication

Για να μπορέσουμε να χρησιμοποιήσουμε το PickPoint API απαιτείται μια διαδικασία επαλήθευσης, η οποία γίνεται με χρήση ενός API Key.

Το API Key μπορούμε να το εκδώσουμε μέσα από την επίσημη [ιστοσελίδα](#) της PickPoint, αφού δημιουργήσουμε έναν λογαριασμό χρήστη.

2.5.2 Περιορισμοί

Η δωρεάν έκδοση του PickPoint API έχει έναν περιορισμό στον αριθμό αιτημάτων που μπορούμε να κάνουμε μέσα σε μια ημέρα. Ο αριθμός αυτός είναι 2,500 αιτήματα.

2.6 Nominatim

Nominatim

Σχήμα 2.5: Λογότυπο Nominatim

Το Nominatim είναι ένα geocoding λογισμικό ανοιχτού κώδικα το οποίο χρησιμοποιεί τα δεδομένα του OpenStreetMap για να βρει τοποθεσίες πάνω στη Γη, είτε αναζητώντας με όνομα είτε με διεύθυνση. Παρέχει επίσης την δυνατότητα Αντίστροφου Geocoding.

2.6.1 Authentication

Το Nominatim δεν απαιτεί κάποια επαλήθευση, μπορούμε απευθείας να αρχίσουμε να πραγματοποιούμε αιτήματα.

2.6.2 Περιορισμοί

Το Nominatim δεν έχει κάποιο περιορισμό στον αριθμό των αιτημάτων που μπορούμε να πραγματοποιήσουμε.

2.7 MapBox



Σχήμα 2.6: Λογότυπο MapBox

Το MapBox είναι ένας πάροχος διαδικτυακών χαρτών για ιστοσελίδες και εφαρμογές όπως το Facebook και το Foursquare. Αντλεί τα δεδομένα του από πηγές ανοιχτού κώδικα όπως είναι το OpenStreetMap και η NASA.

Το MapBox Geocoding API που χρησιμοποιείται εδώ παρέχει υπηρεσίες Forward Geocoding και Αντίστροφου Geocoding.

2.7.1 Authentication

Για να μπορέσουμε να χρησιμοποιήσουμε το MapBox API απαιτείται μια διαδικασία επαλήθευσης, η οποία γίνεται με χρήση ενός Access Token.

Το Access Token μπορούμε να το εκδώσουμε μέσα από την επίσημη [ιστοσελίδα](#) της MapBox, αφού δημιουργήσουμε έναν λογαριασμό χρήστη.

2.7.2 Περιορισμοί

Η δωρεάν έκδοση του MapBox API έχει έναν περιορισμό στον αριθμό αιτημάτων που μπορούμε να κάνουμε στο χρονικό διάστημα ενός μήνα. Ο αριθμός αυτός είναι 100,000 αιτήματα.

Κεφάλαιο 3ο: Η γλώσσα προγραμματισμού Python

3.1 Εισαγωγή

Η υλοποίηση του κώδικα της παρούσας πτυχιακής εργασίας έχει γίνει με την χρήση της γλώσσας προγραμματισμού Python. Σε αυτό το κεφάλαιο θα γίνει μια περιγραφή της Python.

3.2 Εισαγωγή στην Python



Σχήμα 3.1: Λογότυπο Python

Η Python είναι μια πολύ ισχυρή γλώσσα προγραμματισμού πολύ υψηλού επιπέδου. Είναι αρκετά εύκολη στην εκμάθηση της, κάτι που την καθιστά κατάλληλη και για αρχάριους προγραμματιστές και για πιο έμπειρους. Είναι μια γλώσσα γενικής χρήσης, αποδοτική και επεκτάσιμη που σημαίνει πως μπορεί να χρησιμοποιηθεί το ίδιο εύκολα τόσο για εκπαιδευτικούς σκοπούς όσο και για ανάπτυξη ολοκληρωμένων εμπορικών εφαρμογών. Αποτελεί μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού μαζί με την Java και την C++.

Η Python εκτός από αντικειμενοστραφή προσέγγιση, υποστηρίζει διαδικαστικό και συναρτησιακό προγραμματισμό και η σύνταξη της είναι απλή με δυναμικούς τύπους. Ένα ακόμα θετικό της Python είναι πως διαθέτει πάρα πολλές έτοιμες βιβλιοθήκες που μπορούν να χρησιμοποιηθούν εύκολα και γρήγορα.

Αναπτύσσεται ως λογισμικού ανοιχτού κώδικα, με τον συντονισμό και την διαχείριση να πραγματοποιούνται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation (PSF). Σύμφωνα με το PSF, η αποστολή του είναι η προώθηση, προστασία και η εξέλιξη της γλώσσας προγραμματισμού Python. Καθώς και να διευκολύνει την ανάπτυξη μιας διαφορετικής και διεθνούς κοινότητας προγραμματιστών Python.

Η Python όμως εκτός από τα πολλά θετικά χαρακτηριστικά της έχει και κάποια μειονεκτήματα. Ένα εκ των οποίων είναι ότι ο χρόνος εκτέλεσης κάποιων προγραμμάτων μπορεί να μην είναι το ίδιο γρήγορος όσο είναι σε άλλες γλώσσες όπως η C και η C++. Αυτό οφείλεται στο γεγονός πως η Python είναι μια διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού και όχι μεταγλωττιζόμενη (compiled) όπως η C και η C++, που σημαίνει πως ένα πρόγραμμα γραμμένο σε Python δεν εκτελείται άμεσα από τον επεξεργαστή του υπολογιστή, αντ' αυτού διαβάζεται και εκτελείται από ένα άλλο πρόγραμμα (διερμηνευτή – interpreter). Θα μπορούσαμε όμως να πούμε πως αυτό το μειονέκτημα στο χρόνο εκτέλεσης της Python αντισταθμίζεται από το ότι ορισμένες φορές είναι πιο σημαντική η εξοικονόμηση χρόνου στην υλοποίηση μιας εφαρμογής, κάτι που μπορούμε εύκολα να πετύχουμε με την χρήση της Python.

3.3 Ιστορία της Python

Η Python σχεδιάστηκε στα τέλη της δεκαετίας του 1980 από τον Ολλανδό Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI), ως διάδοχος της γλώσσας προγραμματισμού ABC.

Ο Van Rossum ανέλαβε την αποκλειστική ευθύνη για το έργο ως βασικός προγραμματιστής έως τις 12 Ιουλίου 2018, όπου ανακοίνωσε τις «μόνιμες διακοπές» του. Η κοινότητα της Python του έχει αποδώσει τον τίτλο Benevolent Dictator For Life (Καλοσυνάτος Δικτάτορας για μια Ζωή), που αντικατοπτρίζει την μακροχρόνια αφοσίωση του στο έργο ως επικεφαλής λήψης αποφάσεων. Αυτή τη στιγμή ο Van Rossum μοιράζεται την ηγεσία ως μέλος ενός πενταμελούς διοικητικού συμβουλίου, μαζί με τους Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing και απέσυρε την υποψηφιότητα του ως μέλος του διοικητικού συμβουλίου για το 2020.

Η πρώτη έκδοση της Python κυκλοφόρησε το 1991 και η ονομασία της είναι εμπνευσμένη από την δημοφιλή σειρά του BBC Monty Python's Flying Circus και όχι από το φίδι πύθωνας όπως νομίζουν πολλοί.

Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου 2000 με πολλά νέα χαρακτηριστικά συμπεριλαμβανομένου ενός garbage-collector, ο οποίος προσέφερε αυτόματη διαχείριση της μνήμης και έτσι πλέον οι προγραμματιστές δεν χρειάζονται να ανησυχούν για αντικείμενα που δεν είναι πλέον χρήσιμα να καταλαμβάνουν μνήμη.

Η Python 3.0 κυκλοφόρησε στις 3 Δεκεμβρίου 2008. Ήταν μια σημαντική αναθεώρηση της γλώσσας που δεν ήταν πλήρως backwards-compatible, με όμως από τα κύρια χαρακτηριστικά της τελευταίας έκδοσης υποστηρίχθηκαν στις εκδόσεις Python 2.6.x και 2.7.x

3.4 Φιλοσοφία της Python

Η Python έχει ως βάση στη φιλοσοφία της κυρίως την απλότητα και την αναγνωσιμότητα του κώδικα. Στο "The Zen of Python" από τον Tim Peters (PEP 20, The Zen of Python), μια συλλογή με τις 19 βασικές αρχές για την συγγραφή προγραμμάτων, οι οποίες επηρέασαν τον σχεδιασμό της Python.

Για να δούμε αυτές τις αρχές μπορούμε να τρέξουμε τις εντολές `python` και στη συνέχεια `import this`.

Αυτό θα μας τυπώσει στην οθόνη τις αρχές αυτές όπως φαίνεται στο Σχήμα 3.2

```
C:\Users\thmot\Desktop\pt\LocationFinder>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Σχήμα 3.2: The Zen of Python

3.5 Εγκατάσταση Python

Η Python μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά λειτουργικά συστήματα και η διαδικασία της εγκατάστασης είναι πολύ απλή.

3.5.1 Windows

Για να εγκαταστήσουμε την Python στα Windows επισκεπτόμαστε την επίσημη ιστοσελίδα της [Python](#) και επιλέγουμε τον σύνδεσμο Downloads. Από εκεί κατεβάζουμε την τελευταία έκδοση 3.x για Windows. Μπορούμε να διαλέξουμε μεταξύ 32-bit και 64-bit εγκατάστασης. Μόλις κατέβει το αρχείο εγκατάστασης το εκτελούμε και ακολουθούμε τις οδηγίες όπως εμφανίζονται στην οθόνη.

3.5.2 Linux

Συνήθως η Python είναι προεγκατεστημένη σε συστήματα Linux. Αν δεν έχουμε όμως την τελευταία έκδοση μπορούμε να κατεβάσουμε τον πυγαίο κώδικα της τελευταίας έκδοσης από την επίσημη ιστοσελίδα της Python και να τον μεταγλωττίσουμε ή μπορούμε να χρησιμοποιήσουμε τον package manager του λειτουργικού μας συστήματος.

3.5.3 Mac OS X

Συνήθως η Python είναι προεγκατεστημένη σε συστήματα Mac OS. Αν δεν έχουμε όμως την τελευταία έκδοση μπορούμε να κατεβάσουμε την τελευταία έκδοση από την επίσημη ιστοσελίδα της Python επιλέγοντας την έκδοση για λειτουργικό σύστημα Mac OS.

3.6 Λόγος επιλογής Python

Υπάρχουν αρκετοί λόγοι για τους οποίους επιλέχθηκε η Python για την υλοποίηση του κώδικα της παρούσας πτυχιακής εργασίας. Αρχικά είναι αρκετά εύκολη γλώσσα όσο αφορά την εκμάθηση της και ταυτόχρονα έχει πολλές δυνατότητες. Επιπλέον, η Python χρησιμοποιείται πιο συχνά για την ανάπτυξη μικρότερων εφαρμογών όπως η εφαρμογή που υλοποιήθηκε εδώ. Τέλος, και πιο σημαντικό ίσως, όπως αναφέρθηκε προηγουμένως η Python διαθέτει πολλές έτοιμες βιβλιοθήκες που εξυπηρετούν τον σκοπό της παρούσας εργασίας, το geocoding δηλαδή, πολλές απ' αυτές τις βιβλιοθήκες χρησιμοποιήθηκαν για την υλοποίηση του κώδικα.

Κεφάλαιο 4ο: Υλοποίηση πτυχιακής εργασίας

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφερθούμε στα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας πτυχιακής εργασίας.

4.2 PyCharm



Σχήμα 4.1: Λογότυπο PyCharm

Το PyCharm είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών ειδικά σχεδιασμένο για την γλώσσα προγραμματισμού Python. Αναπτύχθηκε από την τσεχική εταιρεία JetBrains. Παρέχει ανάλυση κώδικα, ένα γραφικό πρόγραμμα εντοπισμού σφαλμάτων (debugger), ενοποίηση με συστήματα ελέγχου εκδόσεων (Version Control Systems), όπως το Git.

Η δωρεάν έκδοση του (Community Edition) κυκλοφόρησε κάτω από την άδεια ανοιχτού κώδικα Apache License. Στην υλοποίηση του κώδικα της παρούσας πτυχιακής χρησιμοποιήθηκε η επαγγελματική έκδοση (Professional Edition) η οποία παρέχει πολλά επιπλέον χαρακτηριστικά, όπως υποστήριξη για HTML, JavaScript και SQL.

Η JetBrains παρέχει δωρεάν όλα τα περιβάλλοντα ανάπτυξης εφαρμογών σε μαθητές, φοιτητές και καθηγητές λυκείων και πανεπιστημίων.

Για να χρησιμοποιήσει κάποιος το PyCharm αρκεί απλά να το κατεβάσει από την επίσημη ιστοσελίδα της [JetBrains](https://www.jetbrains.com/pycharm/) και να ακολουθήσει τις οδηγίες όπως αυτές εμφανίζονται στην οθόνη.

4.3 Βιβλιοθήκη geopy

Το geopy είναι μια βιβλιοθήκη που παρέχει τις υλοποιήσεις πολλών διαφορετικών API's σε ένα μόνο πακέτο (package).

Πέρα από την πλειοψηφία των υπηρεσιών που αναφέρθηκαν στο Κεφάλαιο 2, περιλαμβάνει επιπλέον τις υλοποιήσεις για πολλές ακόμα υπηρεσίες geocoding όπως το Bing, το Google Maps v3 API, το ArcGIS και πολλά ακόμα. Έχει εκδοθεί με την άδεια ανοιχτού κώδικα MIT License.

4.3.1 Εγκατάσταση

Η εγκατάσταση του μπορεί να γίνει εκτελώντας την εντολή `pip install geopy` στη γραμμή εντολών.

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\thmot>pip install geopy
Collecting geopy
  Downloading geopy-2.1.0-py3-none-any.whl (112 kB)
    | 112 kB 1.1 MB/s
Collecting geographiclib<2,>=1.49
  Using cached geographiclib-1.50-py3-none-any.whl (38 kB)
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.50 geopy-2.1.0
```

Σχήμα 4.2: Εκτέλεση εντολής pip install geopy

Εναλλακτικά μπορούμε να το κατεβάσουμε το συμπιεσμένο αρχείο από την ιστοσελίδα pypi.org

4.3.2 Τρόπος χρήσης

Για να το χρησιμοποιήσουμε αρκεί να κάνουμε εισαγωγή (import) την βιβλιοθήκη στην εφαρμογή μας. Ένα απλό παράδειγμα χρήσης παρουσιάζεται στο Σχήμα 4.2.

```
from geopy.geocoders import Nominatim

geolocator = Nominatim(user_agent="Geolocation Project")
location = geolocator.geocode("Harvard University")
print((location.latitude, location.longitude))
```

Σχήμα 4.3: Παράδειγμα χρήσης geopy

Στο παραπάνω παράδειγμα αναζητούμε την θέση του Harvard University μέσω του Nominatim API και στο τέλος τυπώνουμε τις συντεταγμένες της θέσης που βρέθηκε.

Για την υλοποίηση της συγκεκριμένης εφαρμογής από την βιβλιοθήκη geopy κρατήσαμε μόνο τα APIs που χρησιμοποιήσαμε και διαγράψαμε τις υλοποιήσεις των υπολοίπων ώστε η εφαρμογή μας να μην περιέχει περιττά στοιχεία.

4.4 Βιβλιοθήκη langdetect

Η βιβλιοθήκη langdetect αποτελεί τη μεταφορά της βιβλιοθήκης language-detection της Google από την Java σε Python. Προσφέρει πάνω από 99% ακρίβεια στην αναγνώριση της γλώσσας για πάνω από 53 γλώσσες.

Ο λόγος που χρησιμοποιείται σε αυτή την υλοποίηση είναι για να μπορέσουμε να αναγνωρίσουμε την γλώσσα που είναι γραμμένο το όνομα του πανεπιστήμιου ή της εταιρίας που αναζητούμε την θέση του, ώστε να μπορέσουμε να κάνουμε ορθογραφικό έλεγχο πριν την αναζήτηση για να μεγιστοποιήσουμε τις πιθανότητες εύρεσης.

4.4.1 Τρόπος χρήσης

Για να χρησιμοποιήσουμε την βιβλιοθήκη langdetect αρκεί να την εισάγουμε στον κώδικά μας. Στο Σχήμα 4.3 παρουσιάζεται ένα παράδειγμα χρήσης για τα αγγλικά και τα ελληνικά.

```

from langdetect import detect

language = detect("Microsoft Corporation")
print(language)

language = detect("Διεθνές Πανεπιστήμιο της Ελλάδος")
print(language)

```

Σχήμα 4.4: Παράδειγμα χρήση βιβλιοθήκης landdetect

Το παραπάνω κομμάτι κώδικα όταν εκτελεστεί θα τυπώσει en και el αντίστοιχα.

4.5 Βιβλιοθήκη autocorrect

Η βιβλιοθήκη autocorrect αποτελεί βιβλιοθήκη ορθογραφικού ελέγχου και διόρθωσης για την Python.

Αυτή τη στιγμή υποστηρίζει έλεγχο για τις γλώσσες Αγγλικά, Πολωνικά, Τούρκικα, Ρώσικα, Ουκρανικά, Τσέχικα, Πορτογαλικά και Ισπανικά.

Έχει εκδοθεί με την άδεια ανοιχτού κώδικα GNU Lesser General Public License v3 (LGPLv3).

4.5.1 Τρόπος χρήσης

Για να χρησιμοποιήσουμε την βιβλιοθήκη autocorrect αρκεί να την εισάγουμε στον κώδικά μας. Στο Σχήμα 4.4 παρουσιάζεται ένα παράδειγμα χρήσης για τα αγγλικά.

```

from autocorrect import Speller
spell = Speller()

corrected = spell("Amezon Inc.")
print(corrected)

corrected = spell("Yaele Uneversety")
print(corrected)

```

Σχήμα 4.5: Παράδειγμα χρήση βιβλιοθήκης autocorrect

Το παραπάνω κομμάτι κώδικα όταν εκτελεστεί θα τυπώσει “Amazon Inc.” και “Yale University” αντίστοιχα.

4.5.2 Χρόνος εκτέλεσης

Ο χρόνος εκτέλεσης της διόρθωσης εξαρτάται κυρίως από τις λέξεις που προσπαθούμε να διορθώσουμε και δεν είναι πάντα σταθερός ακόμα και για τις ίδιες λέξεις.

Αν μετρήσουμε το χρόνο εκτέλεσης των εντολών στο Σχήμα 4.4 θα δούμε ότι ο χρόνος εκτέλεσης κυμαίνεται από ~200ms έως και ~400ms. Αυτό φαίνεται στο Σχήμα 4.5

```

C:\Users\thmot>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import timeit
>>> SETUP_CODE = '''
... from autocorrect import Speller
... spell = Speller()'''
>>> TEST_CODE = '''
... spell("Amezon Inc.")
... spell("Yaele Uneversety)'''
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.3080562000000029
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.2770663000000013
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.4294234999999986

```

Σχήμα 4.6: Χρόνος εκτέλεσης autocorrect

Μπορούμε να μειώσουμε τον χρόνο εκτέλεσης χρησιμοποιώντας την επιλογή fast. Όπως φαίνεται στο Σχήμα 4.6 ο χρόνος εκτέλεσης πλέον έχει μειωθεί σχεδόν στο 1,5ms.

```

C:\Users\thmot>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import timeit
>>> SETUP_CODE = '''
... from autocorrect import Speller
... spell = Speller(fast=True)'''
>>> TEST_CODE = '''
... spell("Amezon Inc.")
... spell("Yaele Uneversety)'''
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.001559599999993111
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.001476900000001142
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.0018976000000066051

```

Σχήμα 4.7: Χρόνος γρήγορης εκτέλεσης autocorrect

Εάν χρησιμοποιήσουμε την επιλογή fast όμως έχουμε ένα σημαντικό μειονέκτημα. Λέξεις με περισσότερα από ένα λάθη δεν μπορούν να διορθωθούν. Στο παράδειγμα εκτέλεσης από το Σχήμα 4.4 η λάθος λέξη “Uneversety” δεν θα διορθωθεί.

Για τις ανάγκες τις παρούσας πτυχιακής είναι προτιμότερο να χρησιμοποιήσουμε την επιλογή fast καθώς χωρίς αυτήν ο χρόνος εκτέλεσης του προγράμματος θα αυξηθεί σημαντικά.

4.6 Βιβλιοθήκη spellcheck

Η βιβλιοθήκη autocorrect που αναφέρθηκε στην προηγούμενη ενότητα θα χρησιμοποιηθεί για τον ορθογραφικό έλεγχο στην αγγλική γλώσσα. Για τα ελληνικά θα χρησιμοποιήσουμε την βιβλιοθήκη spellcheck, η οποία αποτελεί έναν μηχανισμό ελέγχου και διόρθωσης κειμένου χρησιμοποιώντας τη μονάδα αντιστοίχισης συμβολοσειρών Fuzzywuzzy.

Πραγματοποιεί τον έλεγχο βασισμένη σε ένα λεξικό και κατασκευάστηκε από την εταιρεία Pragnakalp.

4.6.1 Τρόπος χρήσης

Για να χρησιμοποιήσουμε την βιβλιοθήκη spellcheck πρέπει πρώτα να κατασκευάσουμε ένα λεξικό σε ένα αρχείο κειμένου με ελληνικές λέξεις. Δεν χρειάζεται να χρησιμοποιήσουμε όλες τις λέξεις τις ελληνικής γλώσσας, εφόσον το σύστημα μας αναζητά εκπαιδευτικά ιδρύματα, λέξεις σχετικές με αυτά θα είναι αρκετές, για παράδειγμα «πανεπιστήμιο» και «ίδρυμα».

Αφού κατασκευάσουμε το λεξικό μας αρκεί να εισάγουμε την βιβλιοθήκη στον κώδικά μας και είμαστε έτοιμοι να την χρησιμοποιήσουμε. Στο Σχήμα 4.7 φαίνεται ένα παράδειγμα χρήσης.

```
from spellcheck import SpellCheck

spell = SpellCheck("greek.txt", encod='UTF-8')
gr = spell.check("Διεθνές Πανεπιστήμιο της Ελλάδας")
suggestions = spell.suggestions()
corrected = ' '.join(suggestions)
print(corrected)
```

Σχήμα 4.8: Παράδειγμα χρήσης spellcheck

Το παραπάνω κομμάτι κώδικα όταν εκτελεστεί θα επιστρέψει «Διεθνές Πανεπιστήμιο της Ελλάδος».

Ο τρόπος λειτουργίας του συγκεκριμένου αλγορίθμου ελέγχου και διόρθωσης είναι αρκετά απλός. Ελέγχει κάθε μια από τις λέξεις στην φράση που έχουμε εισάγει με τις λέξεις στο λεξικό μας. Εάν βρεθεί λέξη στο λεξικό που ταιριάζει σε ποσοστό μεγαλύτερο του 70% με την λέξη που έχουμε εισάγει, την αντικαθιστούμε.

4.6.2 Χρόνος εκτέλεσης

Ο χρόνος εκτέλεσης της διόρθωσης εξαρτάται κυρίως από το μέγεθος του λεξικού μας, εφόσον στην παρούσα πτυχιακή θα χρησιμοποιήσουμε ένα σχετικά μικρό λεξικό αναμένουμε και ο χρόνος εκτέλεσης να είναι μικρός. Στο Σχήμα 4.8 βλέπουμε ότι ο χρόνος εκτέλεσης της διόρθωσης από το παράδειγμα στο Σχήμα 4.7 κυμαίνεται από ~1.5ms μέχρι ~3ms.

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import timeit
>>> SETUP_CODE = '''
... from spellcheck import SpellCheck
... spell = SpellCheck("greek.txt", encod='UTF-8')'''
>>>
>>> TEST_CODE = '''
... gr = spell.check("Διεθνές Πανεπιστήμιο της Ελλάδας")
... suggestions = spell.suggestions()
... corrected = ' '.join(suggestions)'''
>>>
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.003029399999999996
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.001978199999999993
>>> print(timeit.timeit(setup=SETUP_CODE, stmt=TEST_CODE, number=1))
0.0017764000000006774
```

Σχήμα 4.9: Χρόνος εκτέλεσης spellcheck

4.7 Βιβλιοθήκη requests

Όπως αναφέρθηκε στην ενότητα 4.3, η πλειοψηφία των APIs που χρησιμοποιήθηκαν για την εύρεση της γεωγραφικής θέσης των ιδρυμάτων/εταιριών περιλαμβάνεται στην βιβλιοθήκη geopy. Το positionstack όμως δεν υπάρχει στη συγκεκριμένη βιβλιοθήκη. Για να το καλέσουμε χρησιμοποιήθηκε η βιβλιοθήκη requests, η οποία αποτελεί μια απλή βιβλιοθήκη που μας επιτρέπει να στείλουμε HTTP/1.1 αιτήματα πολύ εύκολα.

Με την συγκεκριμένη βιβλιοθήκη μπορούμε να στείλουμε αιτήματα GET, POST, PUT, DELETE, HEAD και OPTIONS, να προσθέσουμε παραμέτρους, να καθορίσουμε custom headers και πολλά ακόμα.

4.7.1 Τρόπος χρήσης

Για να χρησιμοποιήσουμε τη βιβλιοθήκη requests αρκεί να την εισάγουμε στον κώδικά μας.

Στο σχήμα 4.8 μπορούμε να δούμε ένα παράδειγμα χρήσης, όπου δημιουργούμε τις απαραίτητες παραμέτρους για την

```
import requests

payload = {'access_key': 'd90089b961713b9c01c2319f0975243e',
          'query': 'Microsoft Corporation',
          'limit': 1}

response = requests.get(
    url='http://api.positionstack.com//v1/forward', params=payload)
if response.ok:
    print(response.content)
```

Σχήμα 4.10: Παράδειγμα χρήσης requests

Στο παραπάνω παράδειγμα πρώτα ορίζουμε τις παραμέτρους που απαιτούνται για να κληθεί το positionstack API. Οι παράμετροι αυτοί

- **access_key**: ο μοναδικός κωδικός που έχουμε εκδώσει ώστε να γίνει η επαλήθευση χρήστη
- **query**: το οποίο είναι το όνομα της εταιρίας/ιδρύματος που αναζητούμε την διεύθυνση του
- **limit**: με αυτή την παράμετρο περιορίζουμε τον αριθμό των αποτελεσμάτων στην απάντηση

Στη συνέχεια πραγματοποιούμε ένα GET request στο URL του positionstack API χρησιμοποιώντας τις παραμέτρους που δημιουργήσαμε νωρίτερα.

Στην απάντηση επιστρέφονται οι γεωγραφικές πληροφορίες που αναζητούμε.

4.8 Πιθανά Exceptions

Κατά της κλήση των APIs υπάρχει πιθανότητα να προκύψει κάποιο Exception το οποίο εάν δεν το αντιμετωπίσουμε κατάλληλα θα διακοπεί απρόοπτα η εκτέλεση της εφαρμογής μας, κάτι το οποίο πρέπει να αποφύγουμε.

Εφόσον ο σκοπός μας είναι να αναζητήσουμε ένα μεγάλο όγκο δεδομένων με κάθε εκτέλεση της εφαρμογής εάν προκύψει κάποιο Exception το οποίο δεν το διαχειριζόμαστε και διακοπεί απρόοπτα η

εκτέλεση θα πρέπει να ξεκινήσουμε από την αρχή. Για αυτό το λόγο έχουμε τοποθετήσει όλες τις κλήσεις που κάνουμε σε ένα `try-except` block. Τα πιθανά Exceptions που μπορεί να συναντήσουμε είναι:

- **GeocoderTimedOut**

Η geocoding υπηρεσία στην οποία υποβάλαμε το αίτημα μας δεν μας επέστρεψε κάποια απάντηση μέσα στο χρονικό διάστημα που καθορίζεται από την μεταβλητή `timeout`.

- **requests.HTTPError**

Προηγουμένως αναφέραμε πως καλούμε την υπηρεσία `PositionStack` μέσω της βιβλιοθήκης `requests` πραγματοποιώντας ένα HTTP αίτημα. Αυτό το αίτημα μπορεί να αποτύχει για πολλούς λόγους με τον πιο συνηθισμένο να είναι το `error 404 Not Found`.

- **KeyError**

Οι απαντήσεις που μας επιστρέφουν τα APIs περιέχουν κάποια `keys` όπως “city”, “country” κλπ. Αυτά τα `keys` όμως δεν είναι εγγυημένο ότι θα περιέχονται σε κάθε απάντηση. Για αυτό τον λόγο χρειάζεται να διαχειριστούμε και αυτή την περίπτωση.

- **GeocoderServiceError**

Αυτό σημαίνει πως κάποιο Exception προέκυψε κατά την κλήση της geocoding υπηρεσίας αλλά δεν γνωρίζουμε ακριβώς ποιο ήταν το πρόβλημα.

Εάν κατά την εκτέλεση της εφαρμογής μας συναντήσουμε κάποιο από τα παραπάνω κάποιο από τα παραπάνω exceptions θα προσπεράσουμε την συγκεκριμένη εγγραφή και θα συνεχίσουμε με την αναζήτηση των υπολοίπων εγγραφών.

4.9 Επίλογος

Συνδυάζοντας όλες τις βιβλιοθήκες που αναλύθηκαν στην προηγούμενες ενότητες υλοποιήθηκε η συγκεκριμένη πτυχιακή εργασία.

Κεφάλαιο 5ο: Σύγκριση αποτελεσμάτων αναζήτησης

5.1 Εισαγωγή

Στο παρόν κεφάλαιο θα δούμε μερικές περιπτώσεις χρήσης, θα αναζητήσουμε συγκεκριμένα ιδρύματα σε όλα τα API's που έχουν χρησιμοποιηθεί και θα συγκρίνουμε τα αποτελέσματα που μας επιστράφηκαν από το κάθε API με το αποτέλεσμα που βλέπουμε στο Google Maps.

5.2 Περίπτωση χρήσης #1

Ας ξεκινήσουμε με την αναζήτηση “*Harvard University*”. Στον Πίνακα 5.1 φαίνονται το αποτέλεσμα που πήραμε από το κάθε API.

	Διεύθυνση	Latitude	Longitude
MapQuest	Soldiers Field Road, Boston, MA, USA	42.367820	-71.126666
Here	Harvard Ave, University City, MO	38.65905	-90.3096
PositionStack	Boston, MA, USA	42.368714	-71.125877
PickPoint	Soldiers Field Road, Boston, MA, USA	42.367908	-71.126782
Nominatim	Garden Court, Cambridge, MA, USA	42.367908	-71.126782
MapBox	1350 Massachusetts Ave, Cambridge, MA, USA	42.373761	-71.118108

Πίνακας 5.1: Περίπτωση χρήσης #1

Σύμφωνα με την επίσημη σελίδα του πανεπιστημίου του Harvard η διεύθυνση του είναι: *1351 Massachusetts Ave Harvard University Cambridge, MA*. Άρα το ακριβέστερο αποτέλεσμα σύμφωνα με τον Πίνακα 5.1 είναι αυτό που μας επιστράφηκε από το MapBox.

Τι μας επέστρεψαν όμως τα υπόλοιπα APIs; Για να διαπιστώσουμε πόσο σωστά ή λανθασμένα είναι τα αποτελέσματά τους μπορούμε να εισάγουμε τις συντεταγμένες στο Google Maps και να δούμε το αποτέλεσμα.

Σύμφωνα με το Google Maps λοιπόν, στη θέση που μας επέστρεψε το MapQuest βρίσκονται οι αθλητικές εγκαταστάσεις του πανεπιστημίου του Harvard. Οπότε δεν μπορούμε να θεωρήσουμε αυτή την απάντηση λανθασμένη.

Το αποτέλεσμα που μας επέστρεψε το Here από την άλλη είναι ένας δρόμος- ο Harvard Avenue, της πόλης University City στο Μιζούρι των Η.Π.Α-. Το συγκεκριμένο αποτέλεσμα λοιπόν, είναι λανθασμένο.

Εισάγοντας στο Google Maps τα αποτελέσματα των PositionStack, PickPoint και Nominatim επιστρέφονται οι αθλητικές εγκαταστάσεις του πανεπιστημίου, όπως είδαμε και για το MapQuest.

5.3 Περίπτωση χρήσης #2

Σε αυτή την περίπτωση χρήσης θα αναζητήσουμε μια εταιρία, πιο συγκεκριμένα την “Microsoft Corporation”. Στον Πίνακα 5.2 φαίνονται τα αποτελέσματα που πήραμε από κάθε API.

	Διεύθυνση	Latitude	Longitude
MapQuest	Greater Hyderabad Municipal Corporation West Zone, India	17.430769	78.342140
Here	1 Microsoft Way, Redmond, WA, USA	47.63828	-122.13271
PositionStack	Davidson County, Nashville, TN, USA	36.151294	-86.820318
PickPoint	8 CITY BLVD, Nashville-Davidson, TN, USA	36.151294	-86.820318
Nominatim	8 CITY BLVD, Nashville-Davidson, TN, USA	36.151294	-86.820318
MapBox	1 Microsoft Way, Redmond, WA, USA	47.642633	-122.130352

Πίνακας 5.2: Περίπτωση χρήσης #2

Τα κεντρικά γραφεία της Microsoft βρίσκονται στο *Redmond, WA, USA* άρα σύμφωνα με τον Πίνακα 5.2 τα ακριβέστερα αποτελέσματα μας τα επέστρεψαν το Here και το MapBox.

Ας δούμε τα αποτελέσματα που μας επέστρεψαν τα υπόλοιπα APIs. Αναζητώντας τις συντεταγμένες στο Google Maps βλέπουμε πως το MapQuest μας επέστρεψε τα γραφεία της Microsoft στην Ινδία. Μπορεί να μην μας επέστρεψε τα κεντρικά γραφεία της εταιρίας αλλά εφόσον αυτή η διεύθυνση αντιστοιχεί σε παράρτημα της δεν μπορούμε να την θεωρήσουμε λανθασμένη.

Τα PickPoint και Nominatim μας επέστρεψαν το ίδιο αποτέλεσμα, το οποίο είναι η διεύθυνση των γραφείων της Microsoft στο Nashville, Tennessee των Η.Π.Α. Όπως και με το MapQuest δεν μπορούμε να θεωρήσουμε την απάντηση λανθασμένη εφόσον αντιστοιχεί όντως σε γραφεία της εταιρίας.

5.4 Περίπτωση χρήσης #3

Για αυτό το παράδειγμα χρήσης θα χρησιμοποιήσουμε σαν όρισμα αναζήτησης μια λιγότερο ευρέως γνωστή εταιρία, την “Plansee SE” ώστε να δούμε τι αποτελέσματα θα έχουμε. Στον Πίνακα 5.3 φαίνονται τα αποτελέσματα της αναζήτησης μας.

	Διεύθυνση	Latitude	Longitude
MapQuest	Planseestraße, Reutte, Österreich	47.474329	10.812884
Here	Breitenwang, Tirol, Österreich	47.4856	10.83599
PositionStack	Sweden	65.859144	19.017205
PickPoint	Planseestraße, Breitenwang, Österreich	47.474329	10.812884
Nominatim	Planseestraße, Breitenwang, Österreich	47.474329	10.812884
MapBox	Semarang, Central Java, Indonesia	-6.96667	110.41667

Πίνακας 5.3: Περίπτωση χρήσης #3

Τα κεντρικά γραφεία της Plansee SE, σύμφωνα με την επίσημη ιστοσελίδα της, βρίσκονται στο Breitenwang της Αυστρίας.

Όπως μπορούμε να δούμε στον Πίνακα 5.3 σωστό αποτέλεσμα μας έχουν επιστρέψει το MapQuest, το Here, το PickPoint και το Nominatim.

Αναζητώντας τις συντεταγμένες όπως αυτές επιστράφηκαν από το PositionStack και το MapBox δεν φαίνεται να υπάρχει κάποια συσχέτιση με την εταιρία Plansee SE. Στην επίσημη ιστοσελίδα της εταιρίας δεν υπάρχει κάποια αναφορά για γραφεία σε αυτές τις δύο χώρες οπότε τα δύο αυτά αποτελέσματα είναι λανθασμένα.

5.5 Περίπτωση χρήσης #4

Ας δούμε τώρα μια περίπτωση χρήσης για ένα ελληνικό πανεπιστήμιο, το «*Διεθνές Πανεπιστήμιο της Ελλάδος*». Πλέον στο Διεθνές Πανεπιστήμιο υπάγεται και το πρώην Αλεξάνδρειο Τεχνολογικό Ίδρυμα Θεσσαλονίκης, θα έχει ενδιαφέρον λοιπόν να δούμε τι αποτελέσματα θα έχουμε σε αυτή την αναζήτηση. Στον Πίνακα 5.4 φαίνονται τα αποτελέσματα αυτά.

	Διεύθυνση	Latitude	Longitude
MapQuest			
Here			

PositionStack	Central Macedonia, Greece	40.657106	22.807226
PickPoint			
Nominatim	Κανελλοπούλου 2, Δήμος Κατερίνης	40.263767	22.510252
MapBox	Αμερικής 3, Αθήνα	37.977855	23.73429

Πίνακας 5.4: Περίπτωση χρήσης #4

Όπως βλέπουμε στον Πίνακα 5.4 τα MapQuest, Here και PickPoint δεν επέστρεψαν κάποιο αποτέλεσμα στην αναζήτηση μας.

Αν αναζητήσουμε της συντεταγμένες όπως μας επιστράφηκαν από το PositionStack θα δούμε ότι αντιστοιχούν στο τμήμα μας στη Σίνδο. Το συγκεκριμένο αποτέλεσμα λοιπόν είναι σωστό.

Η απάντηση που πήραμε από το Nominatim αντιστοιχεί στο παράρτημα του πανεπιστημίου στην Κατερίνη. Συνεπώς και αυτή η απάντηση θεωρείται σωστή,

Τέλος, η απάντηση που πήραμε από το MapBox αντιστοιχεί στο Μουσείο Τράπεζας της Ελλάδος η οποία είναι και λανθασμένη.

Θα έχει ενδιαφέρον σε αυτή την περίπτωση χρήσης να δούμε αν θα πάρουμε τα ίδια αποτελέσματα αναζητώντας το πανεπιστήμιο στα αγγλικά. Στον Πίνακα 5.5 φαίνονται τα αποτελέσματα αναζήτησης για “*International Hellenic University*”

	Διεύθυνση	Latitude	Longitude
MapQuest	Θεσσαλονίκη - Νέα Μουδανιά, Θέρμη	40.537248	23.008367
Here	5201 University Blvd, Laredo, TX, USA	27.57555	-99.43507
PositionStack	Central Macedonia, Greece	40.537331	23.008567
PickPoint	Θεσσαλονίκη - Νέα Μουδανιά, Θέρμη	40.537357	23.008582
Nominatim	Θεσσαλονίκη - Νέα Μουδανιά, Θέρμη	40.537331	23.008566
MapBox	300 9th Ave, International, WA, USA	47.603732	-122.322524

Πίνακας 5.5: Περίπτωση χρήσης #4 στα αγγλικά

Μπορούμε πολύ εύκολα να διαπιστώσουμε από τον Πίνακα 5.5 ότι με την αναζήτηση του ονόματος στα αγγλικά έχουμε περισσότερα αποτελέσματα. Δεν είναι όλα σωστά βέβαια, το Here και το MapBox μας επέστρεψαν διευθύνσεις στις Η.Π.Α.

Τα υπόλοιπα APIs σε αυτή την αναζήτηση μας επέστρεψαν την έδρα της Θέρμης.

5.6 Περίπτωση χρήσης #5

Η πλειοψηφία των geocoding APIs έχει ως ένα βαθμό ανοχή στα ορθογραφικά λάθη. Στην παρούσα υλοποίηση έχουμε υλοποιήσει ορθογραφικό έλεγχο αλλά δεν μπορούμε να είμαστε 100% βέβαιοι ότι όλα τα ορθογραφικά λάθη θα διορθωθούν. Σε αυτή λοιπόν την περίπτωση χρήσης θα εξετάσουμε τα αποτελέσματα που θα μας επιστραφούν από τα APIs αναζητώντας αρχικά το όνομα ενός ιδρύματος χωρίς ορθογραφικά και στη συνέχεια θα αναζητήσουμε το ίδιο όνομα με ορθογραφικό λάθος. Τα αποτελέσματα με το όνομα χωρίς ορθογραφικά λάθη φαίνονται στον Πίνακα 5.6 και τα αποτελέσματα τις αναζήτησης με το ορθογραφικό λάθος φαίνονται στον Πίνακα 5.7 και 5.8 αντίστοιχα.

Το όρισμα της αναζήτησης μας είναι το “*Meharry Medical College*”.

	Διεύθυνση	Latitude	Longitude
MapQuest	1005 Dr D B Todd Jr Boulevard, Nashville, TN, USA	36.166037	-86.807570
Here	Medical Ave, College Station, TX, USA	30.58286	-96.27585
PositionStack	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.165315	-86.807454
PickPoint	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.166037	-86.807570
Nominatim	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.166022	-86.807561
MapBox	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.166909	-86.806423

Πίνακας 5.6: Περίπτωση χρήσης #5 χωρίς ορθογραφικά

Το Meharry Medical College βρίσκεται στο 1005 Dr D B Todd Jr Boulevard, Nashville, Tennessee. Άρα όπως βλέπουμε στον Πίνακα 5.6 το μόνο API που δεν μας επέστρεψε σωστό αποτέλεσμα είναι το Here, το οποίο μας επέστρεψε την τοποθεσία του Baylor Scott & White Medical Center στο Texas των Η.Π.Α.

Ας δούμε τώρα τι αποτέλεσμα θα έχουμε αν αναζητήσουμε “*Mehary Medical College*”, με ορθογραφικό λάθος στην λέξη “Meharry” δηλαδή. Τα αποτελέσματα αυτής της αναζήτησης φαίνονται στον Πίνακα 5.7

	Διεύθυνση	Latitude	Longitude
MapQuest			
Here	Medical Ave, College Station, TX, USA	30.58286	-96.27585
PositionStack			
PickPoint			
Nominatim			
MapBox	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.166909	-86.806423

Πίνακας 5.7: Περίπτωση χρήσης #5 με ορθογραφικό λάθος (1)

Από τον παραπάνω Πίνακα 5.7 βλέπουμε πως τα MapQuest, PositionStack, PickPoint και Nominatim δεν επέστρεψαν κάποιο αποτέλεσμα στην αναζήτηση μας αυτή τη φορά.

Το Here επέστρεψε το ίδιο λανθασμένο αποτέλεσμα με τον Πίνακα 5.6

Το MapBox είναι το μόνο API που και αυτή τη φορά, παρόλο που είχαμε ορθογραφικό λάθος στο όρισμα της αναζήτησης μας επέστρεψε σωστό αποτέλεσμα

Σε αυτό το σημείο θα έχει ενδιαφέρον να δούμε εάν θα υπάρξει διαφορά εάν το ορθογραφικό λάθος βρισκόταν σε διαφορετική λέξη. Αυτή τη φορά θα αναζητήσουμε “Meharry Medcal College”, το ορθογραφικό λάθος αυτή τη φορά βρίσκεται στη λέξη Medical. Στον Πίνακα 5.8 βλέπουμε τα αποτελέσματα αυτής της αναζήτησης.

	Διεύθυνση	Latitude	Longitude
MapQuest			
Here	Medical Ave, College Station, TX, USA	30.58286	-96.27585
PositionStack	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.165315	-86.807454
PickPoint			
Nominatim			
MapBox	1005 Dr D. B. Todd Jr. Boulevard, Nashville, TN, USA	36.166909	-86.806423

Πίνακας 5.8: Περίπτωση χρήσης #5 με ορθογραφικό λάθος (2)

Στον Πίνακα 5.8 βλέπουμε πως το PositionStack αυτή τη φορά επέστρεψε αποτέλεσμα και μάλιστα το σωστό.

Τα MapQuest, PickPoint και Nominatim από την άλλη εξακολουθούν να μην επιστρέφουν κάποιο αποτέλεσμα. Όπως και το Here εξακολουθεί να επιστρέφει λάθος αποτέλεσμα.

Το MapBox και σε αυτή την περίπτωση επέστρεψε το σωστό αποτέλεσμα.

Από τα παραπάνω μπορούμε λοιπόν να συμπεράνουμε πως έχει σημασία ποια λέξη της αναζήτησης μας περιέχει ορθογραφικό λάθος, καθώς αυτό επηρεάζει και τα αποτελέσματα που θα πάρουμε.

5.7 Επίλογος

Μέσα από τις περιπτώσεις χρήσεις που εξετάσαμε είδαμε πώς στην πλειοψηφία των περιπτώσεων θα πάρουμε απάντηση στην αναζήτηση μας και τις περισσότερες φορές θα είναι σωστή.

Κεφάλαιο 6ο: Σειρά χρήσης APIs

6.1 Εισαγωγή

Όπως είδαμε στις περιπτώσεις χρήσης του Κεφαλαίου 5 δεν μας επιστρέφουν πάντα όλα τα APIs το ίδιο αποτέλεσμα, οπότε σε αυτό το κεφάλαιο θα εξετάσουμε πιο API είναι πιο πιθανό να μας επιστρέψει το σωστό αποτέλεσμα ώστε να χρησιμοποιήσουμε αυτό πρώτο για την αναζήτηση μας. Το αμέσως επόμενο με τις μεγαλύτερες πιθανότητες επιτυχίας θα χρησιμοποιηθεί δεύτερο κ.ο.κ

6.2 Δείγμα δεδομένων

Για να μπορέσουμε να αξιολογήσουμε τα αποτελεσμάτων από κάθε API θα χρειαστεί να χρησιμοποιήσουμε ένα τυχαίο δείγμα δεδομένων. Το δείγμα μας θα είναι ένα αρχείο με 100 τυχαία ονόματα ιδρυμάτων/εταιριών. Τα ονόματα αυτά φαίνονται στο ΠΑΡΑΡΤΗΜΑ Α.

Πρέπει να σημειωθεί πως το δείγμα μας είναι πολύ μικρό για να μπορέσουμε να εξάγουμε ακριβή αποτελέσματα και πως ο λόγος που κάνουμε αυτή την στατιστική ανάλυση είναι μόνο για να αξιολογήσουμε τα APIs ώστε να αποφασίσουμε με πια σειρά θα τα καλούμε στην υλοποίηση μας.

6.3 Ανάλυση αποτελεσμάτων

Μετά την αναζήτηση του δείγματος δεδομένων προέκυψαν τα αποτελέσματα που φαίνονται στον Πίνακα Β.1.

Συνολικά για την εφαρμογή μας βλέπουμε πως έχουμε 0% πιθανότητα να μην βρούμε έστω ένα αποτέλεσμα, το αποτέλεσμα αυτό όμως δεν σημαίνει πως θα είναι σωστό, η πιθανότητα να μας επιστραφεί μόνο λάθος αποτέλεσμα είναι 9%

Στις επόμενες υποενότητες θα αναλύσουμε αυτά τα αποτελέσματα για κάθε API.

6.3.1 Αποτελέσματα MapQuest

Το MapQuest θα εντοπίσει το σωστό αποτέλεσμα στο 55% των περιπτώσεων όπως προκύπτει απ' τον Πίνακα Β.1.

Έχουμε 25% πιθανότητα να μην μας επιστραφεί κανένα αποτέλεσμα μέσω του MapBox, και το υπόλοιπο 20% είναι η πιθανότητα να μας επιστραφεί λάθος αποτέλεσμα.

Από τα παραπάνω προκύπτει πως το MapQuest έχει μέτρια ακρίβεια στα αποτελέσματα που μας επιστρέφει.

6.3.2 Αποτελέσματα Here

Το Here εντόπισε το σωστό αποτέλεσμα μόνο στο 16% των περιπτώσεων.

Το υπόλοιπο ποσοστό χωρίζεται σε 25% πιθανότητα να μην μας επιστρέψει κανένα αποτέλεσμα και 59% πιθανότητα να μας επιστρέψει λάθος αποτέλεσμα.

Απευθείας από τα παραπάνω προκύπτει πως δεν μπορούμε να χρησιμοποιήσουμε το Here ως το πρώτο API στην αναζήτηση μας, αντιθέτως θα είναι το τελευταίο και θα λειτουργεί ως fallback στην περίπτωση που τα υπόλοιπα APIs δεν μας επιστρέψουν κάποιο αποτέλεσμα.

6.3.3 Αποτελέσματα PositionStack

Από τον Πίνακα B.1 προκύπτει πως το PositionStack θα μας επιστρέψει σωστό αποτέλεσμα στο 50% των περιπτώσεων.

Το υπόλοιπο 50% χωρίζεται σε 12% πιθανότητα να μην επιστραφεί κανένα αποτέλεσμα και 38% πιθανότητα λάθος αποτελέσματος.

Από τα παραπάνω προκύπτει πως το PositionStack έχει και αυτό μέτρια αξιοπιστία στα αποτελέσματα που μας επιστρέφει.

6.3.4 Αποτελέσματα PickPoint

Το PickPoint θα μας επιστρέψει σωστό αποτέλεσμα στο 55% των περιπτώσεων.

Το υπόλοιπο 45% χωρίζεται σε 29% πιθανότητα να μην έχουμε κανένα αποτέλεσμα στην αναζήτηση μας και 16% πιθανότητα να μας επιστραφεί λάθος αποτέλεσμα.

Από τα παραπάνω προκύπτει πως και το PickPoint έχει και αυτό μέτρια αξιοπιστία, είναι περίπου στο ίδιο επίπεδο με το MapBox

6.3.5 Αποτελέσματα Nominatim

Από τον Πίνακα B.1 προκύπτει πως το Nominatim θα μας επιστρέψει σωστό αποτέλεσμα στο 66% των περιπτώσεων.

Υπάρχει πιθανότητα 20% να μην μας επιστραφεί κανένα αποτέλεσμα και 14% να έχουμε λάθος αποτέλεσμα στην αναζήτηση μας.

Από τα παραπάνω καταλαβαίνουμε πως το Nominatim είναι το πιο αξιόπιστο από τα APIs που έχουμε αξιολογήσει έως τώρα.

6.3.6 Αποτελέσματα MapBox

Τέλος ας δούμε τα αποτελέσματα που μας επιστράφηκαν από το MapBox. Σύμφωνα και πάλι με τον Πίνακα B.1 το ποσοστό να έχουμε σωστό αποτέλεσμα είναι 68%.

Η πιθανότητα να μην μας επιστραφεί κανένα αποτέλεσμα είναι 0% και η πιθανότητα λάθος αποτελέσματος είναι 32%.

Μπορούμε να διακρίνουμε από τα παραπάνω πως και το MapBox είναι πιο αξιόπιστο από τα υπόλοιπα APIs που αναλύσαμε.

6.4 Συμπεράσματα

Στον Πίνακα 6.1 φαίνονται συγκεντρωτικά τα αποτελέσματα από την ανάλυση που έγινε προηγουμένως και μέσω αυτού του πίνακα θα μπορέσουμε να αποφασίσουμε την κατάλληλη σειρά για να χρησιμοποιήσουμε τα APIs.

API	Ποσοστό σωστών αποτελεσμάτων	Ποσοστό λάθος αποτελεσμάτων	Ποσοστό μη εύρεσης αποτελεσμάτων
MapQuest	55%	20%	25%
Here	16%	59%	25%
PositionStack	50%	38%	12%
PickPoint	55%	16%	29%
Nominatim	66%	14%	20%
MapBox	68%	32%	0%

Πίνακας 6.1: Συγκεντρωτικά αποτελέσματα ανάλυσης APIs

Από τον Πίνακα 6.1 βλέπουμε πως τις περισσότερες πιθανότητες να βρούμε το σωστό αποτέλεσμα τις έχουμε χρησιμοποιώντας πρώτα το MapBox. Το Nominatim έρχεται δεύτερο στο ποσοστό επιτυχίας με μικρή διαφορά.

Δεν θα επιλέξουμε όμως το MapBox ως το πρώτο API στην αναζήτηση μας γιατί έχει σχεδόν την διπλάσια πιθανότητα λάθους απ' ότι το Nominatim.

Οπότε πρώτα θα κάνουμε αναζήτηση στο Nominatim και αν δεν βρούμε κάποιο αποτέλεσμα θα αναζητήσουμε στο MapBox.

Εάν δεν βρούμε αποτέλεσμα ούτε στο MapBox έχουμε να επιλέξουμε ανάμεσα στα PickPoint και MapQuest που είχαν τα αμέσως επόμενα πιο ακριβή αποτελέσματα, με 55% πιθανότητα εύρεσης του σωστού αποτελέσματος και στα δύο.

Εφόσον όμως το PickPoint έχει μικρότερη πιθανότητα λάθους θα επιλεγθεί αυτό ως τρίτο στη σειρά και ένα δεν βρούμε αποτέλεσμα ούτε εκεί θα κάνουμε αναζήτηση στο MapQuest.

Εάν ούτε στο MapQuest βρούμε κάποιο αποτέλεσμα θα προχωρήσουμε με την αναζήτηση μας στο PositionStack, που είναι το επόμενο πιο αξιόπιστο στη σειρά σύμφωνα με τον Πίνακα 6.1.

Τέλος, εάν ούτε το PositionStack μας επιστρέψει κάποιο αποτέλεσμα θα κάνουμε μια τελευταία αναζήτηση στο Here.

Χρησιμοποιώντας τα APIs με αυτή τη σειρά αυξάνουμε τις πιθανότητες μας να βρούμε το σωστό αποτέλεσμα στην αναζήτηση μας.

Κεφάλαιο 7ο: Οδηγός εγκατάστασης και χρήσης λογισμικού

7.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψουμε τη διαδικασία που πρέπει να ακολουθηθεί για την εγκατάσταση της εφαρμογής. Θα ακολουθήσει μετά ένας σύντομος οδηγός χρήσης.

7.2 Οδηγός εγκατάστασης

Για να μπορέσουμε να τρέξουμε την εφαρμογή στον υπολογιστή μας πρέπει πρώτα να έχουμε εγκαταστήσει την Python. Οι οδηγίες για την εγκατάσταση της Python έχουν αναφερθεί στην ενότητα 3.4

Η συγκεκριμένη υλοποίηση έχει δημιουργηθεί με την έκδοση Python 3.9 οπότε ιδανικά για να μην υπάρξει κάποιο πρόβλημα μη συμβατότητας είναι καλύτερο να επιλέξουμε την ίδια έκδοση για εγκατάσταση.

Δεν αρκεί όμως απλά να έχουμε εγκατεστημένη την Python ώστε να μπορέσει να τρέξει η εφαρμογή μας. Εφόσον έχουμε χρησιμοποιήσει third-party βιβλιοθήκες, όπως η langdetect, για την υλοποίηση μας πρέπει να τις εγκαταστήσουμε και αυτές διαφορετικά όταν πάμε να τρέξουμε την εφαρμογή θα έχουμε σφάλμα όπως αυτό που φαίνεται στο Σχήμα 7.1

```
C:\Users\thmot\Desktop\pt\LocationFinder>python findLocation.py
Traceback (most recent call last):
  File "C:\Users\thmot\Desktop\pt\LocationFinder\findLocation.py", line 3, in <module>
    from langdetect import detect
ModuleNotFoundError: No module named 'langdetect'
```

Σχήμα 7.1: Σφάλμα εκτέλεσης

Πρέπει λοιπόν πρώτα να εγκαταστήσουμε τις απαραίτητες βιβλιοθήκες. Η εγκατάσταση βιβλιοθηκών στην Python γίνεται με την εντολή `pip install`.

Ο χρήστης όμως δεν γνωρίζει όλες τις βιβλιοθήκες που χρησιμοποιήθηκαν, για αυτό το λόγο έχουμε δημιουργήσει το αρχείο `requirements.txt` το οποίο περιέχει τις απαραίτητες βιβλιοθήκες με τις εκδόσεις που χρησιμοποιήθηκαν στην παρούσα υλοποίηση. Το αρχείο αυτό φαίνεται στο Σχήμα 7.2

```
six~=1.15.0
pytz~=2020.1
geographiclib~=1.50
setuptools~=50.3.0

fuzzywuzzy~=0.18.0
autocorrect~=2.2.2
spellcheck~=1.0.2
langdetect~=1.0.8

requests~=2.25.1
python-Levenshtein~=0.12.0
```

Σχήμα 7.2: Απαραίτητες βιβλιοθήκες

Έτσι μπορούμε να εγκαταστήσουμε όλες τις απαραίτητες βιβλιοθήκες σε δύο μόνο βήματα.

1. Από την γραμμή εντολών μεταβαίνουμε στο φάκελο όπου υπάρχει η εφαρμογή μας.
2. Τρέχουμε την εντολή `pip install -r requirements.txt`

Αφού το κάνουμε αυτό θα έχουν εγκατασταθεί όλες οι απαραίτητες βιβλιοθήκες και εάν ξανατρέξουμε το παράδειγμα του Σχήματος 7.2 η εφαρμογή μας θα εκτελεστεί χωρίς πρόβλημα όπως φαίνεται στο Σχήμα 7.3.

```
C:\Users\thmot\Desktop\pt\LocationFinder>python findLocation.py
Harvard University
MapQuest: Πόλη: Boston Περιοχή: Massachusetts Χώρα: United States of America Συντεταγμένες: 42.367820 -71.126667
Here: Πόλη: University City Περιοχή: Missouri Χώρα: United States Συντεταγμένες: 38.659050 -90.309600
RotationStack: Πόλη: Boston Περιοχή: Massachusetts Χώρα: United States Συντεταγμένες: 38.659050 -90.309600
PickPoint: Πόλη: Boston Περιοχή: Massachusetts Χώρα: USA Συντεταγμένες: 42.367909 -71.126782
Nominatim: Πόλη: Cambridge Περιοχή: Massachusetts Χώρα: United States Συντεταγμένες: 42.367909 -71.126782
MapBox: Πόλη: Cambridge Περιοχή: Massachusetts Χώρα: United States Συντεταγμένες: 42.373761 -71.118109
```

Σχήμα 7.3: Εκτέλεση παραδείγματος χωρίς σφάλμα

7.2.1 Πιθανό σφάλμα κατά την εγκατάσταση των προϋποθέσεων

Παρατηρήθηκε ότι μπορεί να προκύψει ένα πιθανό σφάλμα κατά την εγκατάσταση των αναγκαίων βιβλιοθηκών στο λειτουργικό σύστημα Windows. Μέρος του σφάλματος φαίνεται στο Σχήμα 7.4

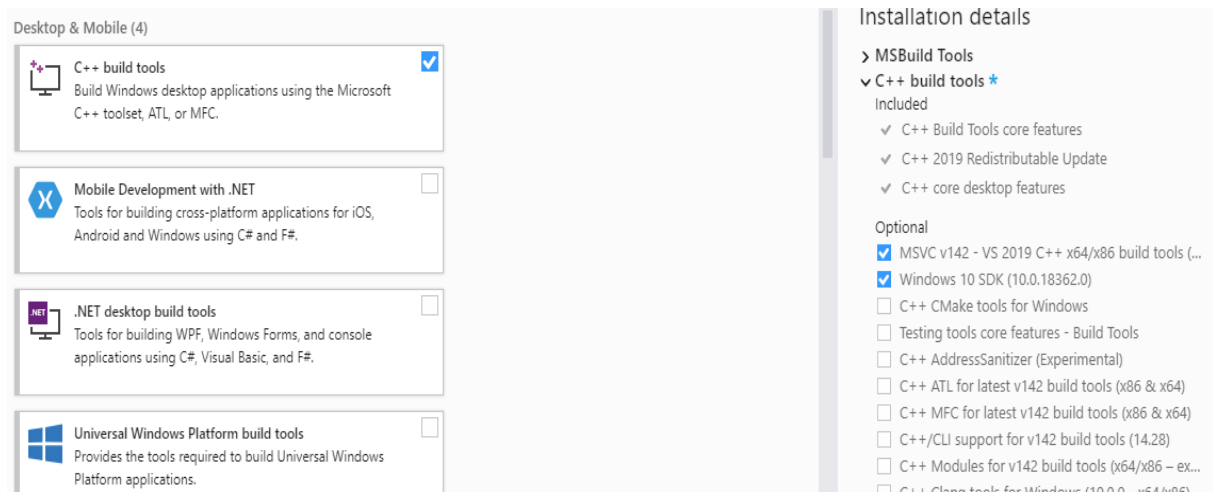
```
building 'Levenshtein_Levenshtein' extension
error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/
```

Σχήμα 7.4: Πιθανό σφάλμα εγκατάστασης Windows

Εάν δείτε αυτό το σφάλμα σημαίνει πως δεν υπάρχει εγκατεστημένη στον υπολογιστή σας η κατάλληλη έκδοση της Microsoft Visual C++ η οποία είναι απαραίτητη για την εγκατάσταση της βιβλιοθήκης python-Levenshtein η οποία χρησιμοποιείται από την βιβλιοθήκη spellcheck.

Η λύση για το σφάλμα είναι να μεταβούμε στην ιστοσελίδα του [VisualStudio](https://visualstudio.microsoft.com/visual-cpp-build-tools/) και να επιλέξουμε τον σύνδεσμο “Download Build Tools”

Μόλις κατέβει το αρχείο το εγκαθιστούμε και στη συνέχεια θα δούμε την σελίδα που φαίνεται στο Σχήμα 7.5



Σχήμα 7.5: Εγκατάσταση C++ build tools

Σε αυτή την σελίδα επιλέγουμε να εγκαταστήσουμε μόνο τα C++ build tools και από την στήλη Installation Details επιλέγουμε επιπλέον τις δυο πρώτες Optional επιλογές όπως φαίνεται στο Σχήμα 7.5.

Αφού ολοκληρωθεί η εγκατάσταση των C++ build tools και ξανατρέξουμε την εντολή `pip install -r requirements.txt` το σφάλμα θα έχει εξαφανιστεί και η εγκατάσταση όλων των απαραίτητων βιβλιοθηκών θα έχει ολοκληρωθεί με επιτυχία.

7.3 Οδηγός χρήσης

Στις ενότητες που ακολουθούν θα αναλύσουμε τον οδηγό χρήσης της εφαρμογής και θα δούμε ένα παράδειγμα ολοκληρωμένης λειτουργίας.

7.3.1 Δημιουργία και αποθήκευση API keys

Όπως αναφέραμε στο Κεφάλαιο 2 τα περισσότερα APIs απαιτούν την χρήση ενός API key για να γίνει το authentication.

Πρώτα λοιπόν πρέπει να ακολουθήσουμε τις οδηγίες για την έκδοση των API keys όπως αυτές περιγράφονται στο Κεφάλαιο 2.

Αφού εκδώσουμε τα API keys ανοίγουμε το αρχείο `config.txt` το οποίο βρίσκεται στον φάκελο της εφαρμογής μας. Εκεί, δίπλα από το όνομα του κάθε API προσθέτουμε το API key που εκδώσαμε προηγουμένως χωρίς κάποιο κενό όπως φαίνεται στο Σχήμα 7.6.

```
mapbox:pk.eyJ1IjoizW1vdHNpb3UuIiwiaWJ0IjoieY2tmdmZ5MGJjMTV5YzJzc3ZwMzhtdTk3MyJ9.ZTE_0ipjhq7TZPjxYU1Aag  
openmapquest:S0XIgELjKIf797c84o4dkIJ7pZmkj8y5  
pickpoint:MTJdTd3o1W8YUA7BDy-DD  
here:owD4DyWzNQY9a9MUckYa9PUMOSJz6_iILAxZsVh-m0  
positionstack:d90089b961713b9c01c2319f0975243e
```

Σχήμα 7.6: Αρχείο `config.txt`

Εάν για οποιοδήποτε λόγο χρειαστεί στο μέλλον να αλλάξουμε το API key για κάποιο από τα APIs αρκεί απλά να επεξεργαστούμε το αρχείο `config.txt`

7.3.2 Αρχείο δεδομένων

Η εφαρμογή μας λαμβάνει τα δεδομένα από ένα αρχείο κειμένου, οπότε χρειάζεται να δημιουργήσουμε ένα αρχείο κειμένου με ένα όρισμα ανά γραμμή. Το αρχείο κειμένου μπορεί να ονομαστεί όπως επιθυμούμε και να περιέχει όσα ονόματα επιθυμούμε. Επίσης μπορούμε να το αποθηκεύσουμε οπουδήποτε στον υπολογιστή μας εφόσον η διαδρομή του αρχείου θα περνά ως παράμετρος στον κώδικα μας όπως θα δούμε παρακάτω.

7.3.3 Εκτέλεση εφαρμογής

Αφού έχουμε ολοκληρώσει την δημιουργία του αρχείου `config.txt` και του αρχείου δεδομένων μπορούμε να εκτελέσουμε την εφαρμογή μας.

Πρώτα ανοίγουμε την γραμμή εντολών και μεταβαίνουμε στον φάκελο της εφαρμογής. Στη συνέχεια όπως φαίνεται στο Σχήμα 7.7 εκτελούμε την εντολή:

```
python findLocation.py -filePath $αρχείοΔεδομένων
```

Το αρχείο δεδομένων που χρησιμοποιήσαμε εδώ περιέχει τα ιδρύματα των Πινάκων Α.1 και Α.2

```
C:\Users\thmot\Desktop\pt\LocationFinder>python findLocation.py -filePath "C:\Users\thmot\Desktop\sample.txt"
Running Location Finder...
Results saved in: C:\Users\thmot\Desktop\pt\LocationFinder\results.csv
```

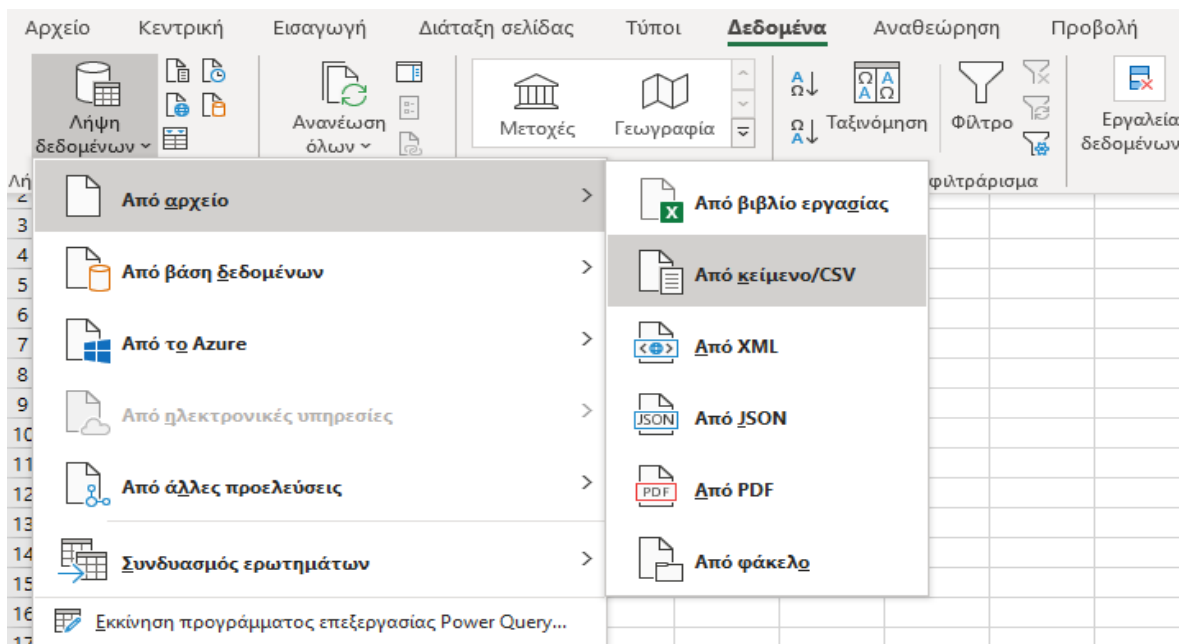
Σχήμα 7.7: Εκτέλεση εφαρμογής

Όταν ολοκληρωθεί η εκτέλεση της εφαρμογής τυπώνουμε την διαδρομή όπου έχει αποθηκευτεί το αρχείο αποτελεσμάτων, σε μορφή CSV.

7.3.4 Ανάγνωση αποτελεσμάτων στο Excel

Για να μπορέσουμε να δούμε τα αποτελέσματα χωρίς προβλήματα κωδικοποίησης στο Excel χρειάζεται να ακολουθήσουμε τα παρακάτω βήματα.

1. Ανοίγουμε το Excel και επιλέγουμε την δημιουργία νέου κενού βιβλίου εργασίας
2. Επιλέγουμε την καρτέλα Δεδομένα
3. Επιλέγουμε Λήψη Δεδομένων-Από Αρχείο-Από κείμενο/CSV (Σχήμα 7.8)
4. Στο παράθυρο που θα ανοίξει αλλάζουμε την «Προέλευση αρχείου» σε «650001: Unicode (UTF-8)»
5. Πατάμε το κουμπί «Φόρτωση»



Σχήμα 7.8: Ανοιγμα αποτελεσμάτων στο Excel

Αφού ολοκληρώσουμε τα παραπάνω βήματα το αρχείο αποτελεσμάτων θα ανοίξει και θα φαίνεται όπως στο Σχήμα 7.9

	A	B	C	D	E	F
	Όνομα Ιδρύματος	Πόλη	State	Χώρα	Latitude	Longitude
2	Universidad Alas Peruanas	Cajamarca	Cajamarca	Peru	-7.158711	-78.504119
3	King Mongkut's University of Technology Thonburi	Bang Mot Subdistrict	Bangkok	Thailand	13.6518	100.494422
4	University of Twente	Enschede	Overijssel	Netherlands	52.223386	6.885423
5	Mie University	Unknown	Florida	United States	25.714815	-80.276977
6	Shiga University of Medical Science	草津市	Shiga	Japan	34.973733	135.950386
7	Children's Hospital of Wisconsin	Wauwatosa	Wisconsin	United States	43.042224	-88.023516
8	Cincinnati Children's Hospital Medical Center	Cincinnati	Ohio	United States	39.140845	-84.502288
9	Lille University of Science and Technology	Cagayan de Oro City	Cagayan de Oro	Philippines	8.485531	124.656748
10	Brookings Institution	Washington	Washington, D.C.	United States	38.908814	-77.040181
11	Western Michigan University	Kalamazoo	Michigan	United States	42.283338	-85.609507
12	Cairo University	Giza	Been Al-Sarayat	Egypt	30.027145	31.205819
13	University of Sistan and Baluchestan	Zahedan	Sistan and Baluchestan Province	Iran	29.46457	60.856505
14	Aerojet Rocketdyne	Socorro	New Mexico	United States	34.05221	-106.898507
15	United States Postal Service	North Richland Hills	Texas	United States	32.855999	-97.213435
16	Irkutsk State Technical University	Иркутское	Irkutsk	Russia	52.527558	103.864676
17	Nishikyushu University	Fukuoka	Nishi Ward	Japan	33.599587	130.227943
18	University of North Sumatra	Belfast	Northern Ireland	United Kingdom	54.585299	-5.939172
19	Ball State University	Muncie	Indiana	United States	40.209784	-85.411203
20	Aims Community College	Greeley	Colorado	United States	40.40936	-104.763332
21	University of Kinshasa	Kinshasa	Golf	Democratic Republic of the Congo	-4.305121	15.3028

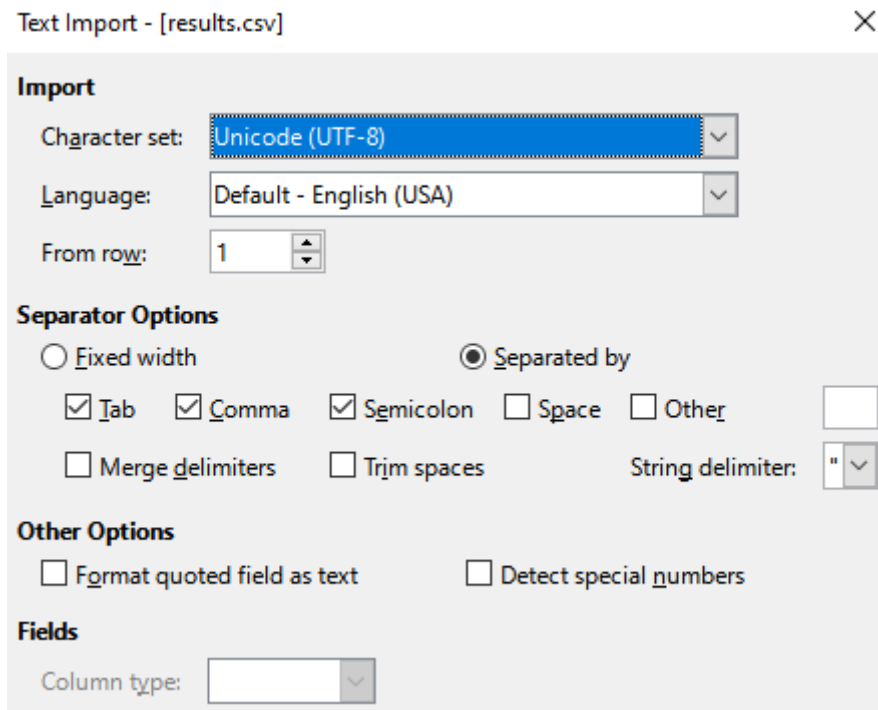
Σχήμα 7.9: Αποτελέσματα αναζήτησης

7.3.5 Ανάγνωση αποτελεσμάτων στο LibreOffice

Εάν χρησιμοποιούμε LibreOffice τα βήματα για την σωστή ανάγνωση του αρχείου χωρίς προβλήματα κωδικοποίησης είναι πιο απλά.

Απλά ανοίγουμε το αρχείο results.csv και στο παράθυρο του Σχήματος 7.10 επιλέγουμε “Unicode (UTF-8)” ως “Character set” και πατάμε OK.

Τα αποτελέσματα θα φαίνονται όπως και στο Excel (Σχήμα 7.9)



Σχήμα 7.10: Άνοιγμα αποτελεσμάτων στο LibreOffice

Κεφάλαιο 8ο: Προτάσεις για μελλοντική βελτίωση

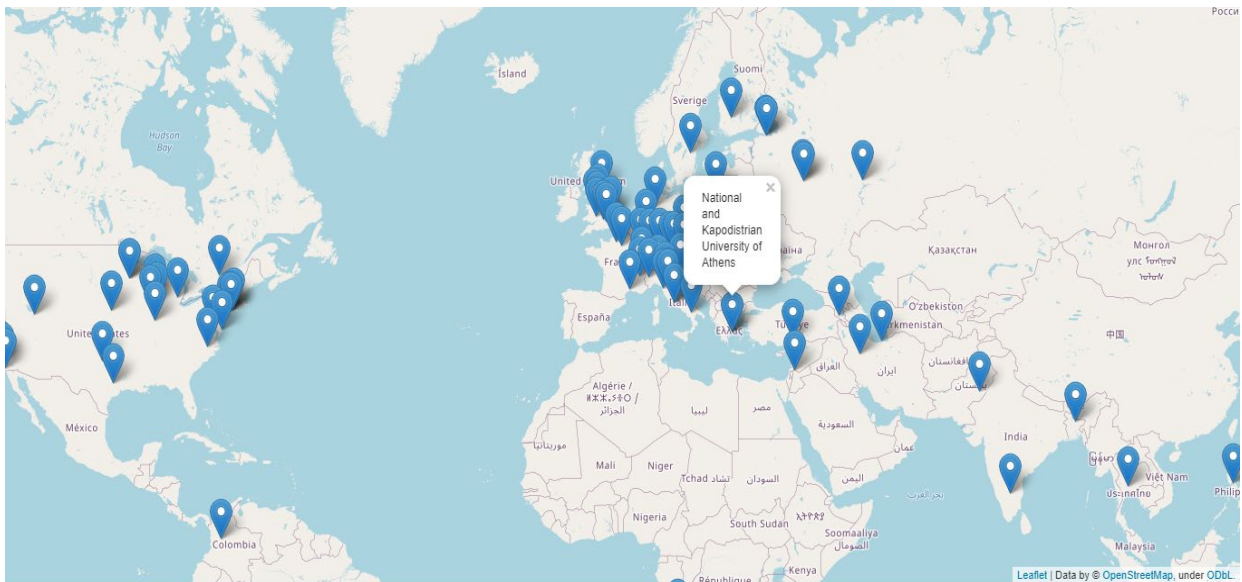
8.1.1 Γραφικό περιβάλλον

Η εφαρμογή που δημιουργήθηκε για την συγκεκριμένη πτυχιακή εργασία είναι σχεδιασμένη να τρέχει μέσω της γραμμής εντολών, εάν χρειαστεί στο μέλλον θα μπορούσε να δημιουργηθεί μια desktop εφαρμογή η οποία θα περιλαμβάνει και GUI. Στην Python αυτό μπορεί να γίνει είτε με την ενσωματωμένη βιβλιοθήκη Tkinter, είτε επιλέγοντας κάποιο third-party module, τα πιο δημοφιλή στην Python είναι τα PyQt, PySide, WxPython και Kivy.

8.1.2 Απεικόνιση δεδομένων

Στην Python είναι αρκετά απλό να απεικονίσουμε γεωγραφικά δεδομένα σε χάρτη. Αυτό μπορεί να επιτευχθεί με τη χρήση της βιβλιοθήκης folium. Στην παρούσα πτυχιακή δεν κρίθηκε απαραίτητο να απεικονίσουμε τα αποτελέσματα σε χάρτη αλλά αν χρειαστεί στο μέλλον μπορούμε να το πετύχουμε προσθέτοντας λίγες γραμμές κώδικα.

Ο χάρτης για ένα μέρος των δεδομένων που περιέχονται στο στην βάση Microsoft Academic Graph 2020 θα φαινόταν όπως στο Σχήμα 9.1



Σχήμα 9.1: Απεικόνιση αποτελεσμάτων σε χάρτη

Η Python επίσης προσφέρει αρκετές βιβλιοθήκες για δημιουργία heatmaps, όπως πχ η βιβλιοθήκη pandas. Οπότε στο μέλλον θα μπορούσαμε να βελτιώσουμε την παρούσα υλοποίηση με την απεικόνιση των αποτελεσμάτων σε heatmap.

Κεφάλαιο 9ο: Συμπεράσματα

Στη συγκεκριμένη πτυχιακή εργασία αναπτύχθηκε και υλοποιήθηκε ένα σύστημα το οποίο εντοπίζει την γεωγραφική θέση ενός συνόλου δεδομένων από πανεπιστημιακά ιδρύματα και εταιρίες αποθηκευμένα σε ένα αρχείο κειμένου, και αποθηκεύει τις πληροφορίες αυτές σε ένα αρχείο CSV για εύκολη ανάγνωση.

Επιπλέον χρησιμοποιεί και ένα σύστημα ορθογραφικού ελέγχου για τα αγγλικά και τα ελληνικά ώστε να μεγιστοποιήσουμε τις πιθανότητες εύρεσης αποτελέσματος.

8. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Νικόλαος Α. Αγγλιδάκης, *Εισαγωγή στον προγραμματισμό με την Python*. Ηράκλειο, 2015
- [2] Γεώργιος Μανής, *Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python*, Αθήνα, 2015
- [3] Nominatim Documentation, Available: <https://nominatim.org/release-docs/latest/>
- [4] Mapbox Documentattion, Available: <https://docs.mapbox.com/>
- [5] Positionstack API Documentation, Available: <https://positionstack.com/documentation>
- [6] Here Geocoding API, Available: <https://developer.here.com/>
- [7] United States Census Bureau, “25th Anniversary of TIGER”, Available: <https://census.maps.arcgis.com/apps/MapJournal/index.html?appid=2b9a7b6923a940db84172d6de138eb7e>.
- [8] Map Quest Developer Documentation, Available: <https://developer.mapquest.com/>
- [9] PickPoint Geocoding Service, Available: <https://pickpoint.io/>
- [10] Geocoding Resources, “Geographic Information Systems at Yale: Geocoding Resources”, Yale University Library [Online], Available: <https://guides.library.yale.edu/GIS/geocoding>
- [11] George Rebhan, “The Evolution of Geocoding: Moving Away From Conflation Confliction to Best Match” *Directions Magazine*, 12 Jul. 2007
- [12] Wikipedia, Available: https://en.wikipedia.org/wiki/Main_Page
- [13] Python, Available: <https://www.python.org/>
- [14] Geopy Documentation, Available: <https://geopy.readthedocs.io/en/latest/>

ΠΑΡΑΡΤΗΜΑ Α: ΔΕΙΓΜΑ ΔΕΔΟΜΕΝΩΝ

Στήλη1	Στήλη2
Universidad Alas Peruanas	Technical University of Lisbon
King Mongkut's University of Technology Thonburi	Concordia University
University of Twente	University of Franche-Comté
Mie University	Clarkson University
Shiga University of Medical Science	Shanghai University
Children's Hospital of Wisconsin	Cardiovascular Institute of the South
Cincinnati Children's Hospital Medical Center	Cedars-Sinai Medical Center
Lille University of Science and Technology	University of Technology, Sydney
Brookings Institution	University of Bolton
Western Michigan University	Universidad del Desarrollo
Cairo University	Flinders Medical Centre
University of Sistan and Baluchestan	University of Sussex
Aerojet Rocketdyne	University of Nigeria, Nsukka
United States Postal Service	Pennington Biomedical Research Center
Irkutsk State Technical University	Texas A&M University–Commerce
Nishikyushu University	PTC
University of North Sumatra	Texaco
Ball State University	Marine Biological Laboratory
Aims Community College	Biocrates Life Sciences AG
University of Kinshasa	Technical University of Košice
Joint Graduate School of Energy and Environment	American College of Cardiology
University of Modena and Reggio Emilia	Heart and Diabetes Center North Rhine-Westphalia
Clemson University	Computer Sciences Corporation
Indian Institute of Technology Delhi	Festo
Government Royapettah Hospital	SAGEM

Πίνακας Α.1: Δείγμα δεδομένων 1-50

Στήλη3	Στήλη4
Kobe Steel	University of Lagos
Mann+Hummel	Office National d'Études et de Recherches Aérospatiales
Universidad de Ciencias Medicas	Bilkent University
Ministry of Agriculture	Koç University
Hamedan University of Medical Sciences	Central University of Technology
Nara Institute of Science and Technology	Razi University
Lehman College	Jichi Medical University
Milwaukee Electric Tool Corporation	Beijing Normal University
Roquette Freres	Trinity College, Dublin
Sanyo	Al-Farabi University
University of Salford	University of Reading
Chongqing Normal University	National Agrarian University (Nicaragua)
Sharp Grossmont Hospital	German Research Centre for Artificial Intelligence
University of Macedonia	Ernst & Young
Halliburton	Citibank
Society of Hospital Medicine	SAS Institute
Universitas Ahmad Dahlan	Joseph Fourier University
University of Kragujevac	University of Tasmania
SRI International	University of Pardubice
Whitney Laboratory for Marine Bioscience	Dublin City University
Lille Catholic University	Kagoshima University
Yukon College	Institute of Cost and Management Accountants of Bangladesh
Helmut Schmidt University	PerkinElmer
McKnight Brain Institute	University of Sarajevo
Nippon Paper Group, Inc.	Federal University of Technology - Paraná

Πίνακας Α.2: Δείγμα δεδομένων 51-100

ΠΑΡΑΡΤΗΜΑ Β: ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ

Στον Πίνακα Β.1 για να κάνουμε πιο εύκολη την ανάγνωση εάν είχαμε απάντηση και ήταν σωστή θα χρησιμοποιούμε το σύμβολο ✓, εάν είχαμε απάντηση αλλά ήταν λανθασμένη θα χρησιμοποιούμε το σύμβολο ✘ και με εάν δεν πήραμε καμία απάντηση στην αναζήτηση μας θα χρησιμοποιήσουμε το σύμβολο -.

Όρισμα αναζήτησης	MapQuest	Here	PositionStack	PickPoint	Nominatim	MapBox
Universidad Alas Peruanas	✓	✘	✓	✓	✓	✓
King Mongkut's University of Technology Thonburi	✓	-	✓	✓	✓	✓
University of Twente	✓	✘	-	✓	✓	✓
Mie University	✘	✘	✘	✘	✘	✓
Shiga University of Medical Science	-	-	-	-	-	✓
Children's Hospital of Wisconsin	✘	-	✘	✘	✓	✓
Cincinnati Children's Hospital Medical Center	✓	✘	-	✓	✓	✘
Lille University of Science and Technology	-	-	✘	-	-	✘
Brookings Institution	✓	✘	✘	✓	✓	✓
Western Michigan University	✓	✘	✓	✓	✓	✓
Cairo University	✓	✘	✓	✓	✓	✓
University of Sistan and Baluchestan	✓	✘	-	✓	✓	✘
Aerojet Rocketdyne	✓	✓	✓	✓	✓	✓
United States Postal Service	✓	-	✓	✓	✓	✓
Irkutsk State Technical University	-	✘	✘	-	-	✓

Nishikyushu University	x	x	-	-	✓	x
University of North Sumatra	-	x	x	-	x	✓
Ball State University	✓	✓	✓	✓	✓	✓
Aims Community College	✓	✓	✓	✓	✓	✓
University of Kinshasa	-	x	✓	-	✓	x
Joint Graduate School of Energy and Environment	-	-	x	-	-	x
University of Modena and Reggio Emilia	-	x	-	-	-	✓
Clemson University	x	✓	x	✓	✓	✓
Indian Institute of Technology Delhi	✓	✓	✓	✓	✓	✓
Government Royapettah Hospital	-	-	✓	-	✓	x
Technical University of Lisbon	-	x	x	-	-	✓
Concordia University	✓	✓	✓	✓	✓	✓
University of Franche-Comté	-	x	x	-	-	✓
Clarkson University	✓	✓	✓	✓	✓	✓
Shanghai University	✓	x	x	✓	✓	✓
Cardiovascular Institute of the South	x	-	x	-	-	✓
Cedars-Sinai Medical Center	✓	-	✓	✓	✓	✓
University of Technology, Sydney	✓	✓	x	-	-	✓
University of Bolton	✓	✓	✓	✓	✓	x
Universidad del Desarrollo	✓	x	✓	✓	✓	✓
Flinders Medical Centre	✓	x	✓	✓	✓	✓

University of Sussex	✓	✗	✓	✓	✓	✓
University of Nigeria, Nsukka	✓	✓	✓	✓	✓	✗
Pennington Biomedical Research Center	-	✗	✗	-	-	✓
Texas A&M University– Commerce	✓	✓	✗	✓	✓	✓
PTC	✓	✗	✗	✓	✓	✗
Texaco	✓	✓	✓	✓	✓	✓
Marine Biological Laboratory	✓	-	✓	✓	✓	✓
Biocrates Life Sciences AG	-	-	-	-	-	✗
Technical University of Košice	✓	✗	✗	✗	✓	✗
American College of Cardiology	-	✗	✗	-	-	✓
Heart and Diabetes Center North Rhine- Westphalia	-	✗	-	-	-	✗
Computer Sciences Corporation	✓	-	✓	✓	✓	✓
Festo	✓	✗	✓	✗	✓	✓
SAGEM	✗	✗	✗	✓	✓	✗
Kobe Steel	-	✗	✓	-	✓	✓
Mann+Hummel	✓	-	✓	✓	✓	✓
Universidad de Ciencias Medicas	✓	✗	✓	✓	✓	✓
Ministry of Agriculture	✗	-	-	✓	✓	✓
Hamedan University of Medical Sciences	-	✗	✓	-	-	✗
Nara Institute of Science and Technology	✓	-	✓	✓	✓	✓

Lehman College	x	x	-	x	x	✓
Milwaukee Electric Tool Corporation	-	x	x	-	-	✓
Roquette Freres	✓	x	✓	✓	✓	✓
Sanyo	x	x	x	x	x	x
University of Salford	✓	✓	✓	✓	✓	✓
Chongqing Normal University	x	x	x	✓	✓	x
Sharp Grossmont Hospital	✓	x	-	✓	✓	✓
University of Macedonia	✓	x	x	✓	✓	✓
Halliburton	x	x	x	x	x	x
Society of Hospital Medicine	x	-	x	x	x	✓
Universitas Ahmad Dahlan	-	-	x	-	x	✓
University of Kragujevac	-	x	✓	x	x	x
SRI International	✓	x	✓	✓	✓	✓
Whitney Laboratory for Marine Bioscience	✓	-	✓	✓	✓	✓
Lille Catholic University	-	-	x	-	-	✓
Yukon College	✓	x	✓	✓	✓	✓
Helmut Schmidt University	x	x	✓	-	x	x
McKnight Brain Institute	✓	x	✓	✓	✓	✓
Nippon Paper Group, Inc.	-	-	x	-	-	x
University of Lagos	✓	x	✓	✓	✓	x
Office National d'Études et de Recherches Aérospatiales	-	-	x	-	-	x

Bilkent University	✓	✗	✓	✓	✓	✓
Koç University	✓	✗	✗	✓	✓	✓
Central University of Technology	✗	✗	✗	✗	✗	✓
Razi University	✓	✗	✗	✗	✗	✗
Jichi Medical University	✓	✗	✗	✓	✓	✓
Beijing Normal University	✓	✗	✓	✓	✓	✓
Trinity College, Dublin	✓	✓	✓	✓	✓	✓
Al-Farabi University	✗	✗	✓	✗	✗	✗
University of Reading	✗	✗	✓	✗	✓	✓
National Agrarian University (Nicaragua)	-	✗	✗	-	-	✗
German Research Centre for Artificial Intelligence	✓	-	✓	✓	✓	✗
Ernst & Young	✓	-	✗	✓	✓	✗
Citibank	✓	-	✓	✗	✗	✓
SAS Institute	✓	✗	✓	✓	✓	✓
Joseph Fourier University	✗	✗	✗	✗	✗	✓
University of Tasmania	✓	✓	✓	✓	✓	✓
University of Pardubice	-	✗	✗	-	✓	✓
Dublin City University	✓	✓	✓	✓	✓	✓
Kagoshima University	✓	✗	✓	✓	✓	✓
Institute of Cost and Management Accountants of Bangladesh	✓	✗	✓	✓	✓	✗

PerkinElmer	✓	-	✓	✓	✓	✓
University of Sarajevo	-	✗	✗	-	✓	✗
Federal University of Technology – Paraná	-	✗	-	-	-	✗

Πίνακας Β.1: Αποτελέσματα αναζήτησης