24 ΔΕΚ 2022

Αναφορά Project

Αναπαράσταση Γνώσης στον Παγκόσμιο Ιστό

Παπαδάκης Νικόλαος

Ψαλτάκης Γιώργος TH20027

Υλοποίηση SPARQL

Το υλοποιημένο SPARQL ανακτά όλες τις περιπτώσεις, τους τύπους τους, τις ιδιότητες και τις τιμές τους από το γράφημα RDF. Περιλαμβάνει επίσης μια προαιρετική ρήτρα για την ανάκτηση του ονόματος της περίπτωσης, εάν υπάρχει.

Ξεκινάω με τον ορισμό του προθέματος χώρου ονομάτων rdf ως <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. Αυτό το πρόθεμα χρησιμοποιείται για να αναφερθεί στο χώρο ονομάτων RDF, ο οποίος ορίζει αρκετές βασικές έννοιες που χρησιμοποιούνται στο RDF, συμπεριλαμβανομένης της ιδιότητας type.

Καθορίζω στη συνέχεια τις μεταβλητές που πρέπει να επιλεγούν: ?type, ?instance, ?property, ?value και ?name. Η μεταβλητή ?type αντιπροσωπεύει τον τύπο της περίπτωσης, η μεταβλητή ?instance αντιπροσωπεύει την ίδια την περίπτωση, η μεταβλητή ?property αντιπροσωπεύει μια ιδιότητα της περίπτωσης, η μεταβλητή ?value αντιπροσωπεύει την τιμή της ιδιότητας και η μεταβλητή ?name αντιπροσωπεύει το όνομα της περίπτωσης.

Στη συνέχεια, ορίζω το μοτίβο προς αντιστοίχιση, το οποίο αποτελείται από τρία τριπλά μοτίβα. Το πρώτο τριπλό μοτίβο ?instance rdf:type ?type ταιριάζει με κάθε τριπλό στο γράφημα όπου το υποκείμενο είναι μια περίπτωση και το αντικείμενο είναι ένας τύπος, και συνδέει το υποκείμενο με τη μεταβλητή ?instance και το αντικείμενο με τη μεταβλητή ?type.

Το δεύτερο τριπλό μοτίβο ?instance ?property ?value ταιριάζει με οποιαδήποτε τριπλέτα στο γράφημα όπου το υποκείμενο είναι μια περίπτωση και το αντικείμενο είναι μια τιμή, και συνδέει το υποκείμενο με τη μεταβλητή ?instance, το κατηγόρημα με τη μεταβλητή ?property και το αντικείμενο με τη μεταβλητή ?value. Το τρίτο τριπλό μοτίβο OPTIONAL { ?instance <http://xmlns.com/foaf/0.1/name> ?name } είναι μια προαιρετική ρήτρα που ταιριάζει με οποιαδήποτε τριπλέτα στο γράφημα όπου το υποκείμενο είναι μια περίπτωση και το κατηγόρημα είναι η ιδιότητα FOAF name, και δεσμεύει το υποκείμενο στη μεταβλητή ?instance και το αντικείμενο στη μεταβλητή ?name. Η λέξη-κλειδί OPTIONAL καθορίζει ότι το τριπλό μοτίβο είναι προαιρετικό. Στη συνέχεια, καθορίζω τη ρήτρα WHERE, η οποία περιέχει τα τριπλά πρότυπα που ορίζονται παραπάνω.

Συνοπτικά, το υλοποιημένο SPARQL ανακτά όλες τις περιπτώσεις, τους τύπους τους και όλες τις ιδιότητες και τις τιμές των περιπτώσεων, καθώς και τα ονόματα των περιπτώσεων, εφόσον υπάρχουν. Αυτό επιτυγχάνεται με την αντιστοίχιση τριπλετών στο γράφο όπου το υποκείμενο είναι μια περίπτωση, το κατηγόρημα είναι είτε η ιδιότητα τύπου RDF είτε οποιαδήποτε άλλη ιδιότητα και το αντικείμενο είναι είτε ένας τύπος είτε μια τιμή. Περιλαμβάνει επίσης μια προαιρετική ρήτρα για την αντιστοίχιση τριπλετών όπου το κατηγόρημα είναι η ιδιότητα FOAF name.

Αναλυτική Περιγραφή του κώδικα

Εισάγω απαραίτητες βιβλιοθήκες, συμπεριλαμβανομένων των rdflib, DictVectorizer και cosine_similarity από το scikit-learn και KMeans από το scikit-learn.
Στην συνέχεια δημιουργώ έναν γράφο RDF και αναλύστε ένα αρχείο RDF (που καθορίζεται από τη διαδρομή αρχείου "kenza_conference.rdf") στον γράφο χρησιμοποιώντας τη μέθοδο Graph.parse().
Ορίζω SPARQL για την ανάκτηση όλων των περιπτώσεων, τύπων, ιδιοτήτων και τιμών από το γράφημα RDF όπως αναλύω και πάνω Χρησιμοποιήστε τη μέθοδο Graph.query() για να εκτελέσω το SPARQL και να αποθηκεύσω τα αποτελέσματα σε μια μεταβλητή results.
Δημιουργώ μια κενή λίστα instances για να αποθηκεύσω τα λεξικά που αντιπροσωπεύουν κάθε instance.
Επαναλάμβάνω τα αποτελέσματα του ερωτήματος και για κάθε γραμμή αποτελέσματος:
Αποθηκευω την περίπτωση, την ιδιότητα και την τιμή σε μεταβλητές.
Ελέγχω αν υπάρχει ήδη ένα λεξικό για την περίπτωση στη λίστα instances αναζητώντας ένα λεξικό με κλειδί "uri" που έχει την ίδια τιμή με την περίπτωση. Εάν υπάρχει, το αποθηκευω σε μια μεταβλητή instance_dict. Αν δεν υπάρχει, δημιουργω ένα νέο λεξικό με κλειδί "uri" και την τιμή του instance και το αποθηκεύω στη μεταβλητή instance_dict.

Προσθέτω την ιδιότητα και την τιμή ως ζεύγος κλειδιού-τιμής στο λεξικό instance_dict.
Εάν το όνομα της περίπτωσης δεν είναι None, το προσθέτω στο λεξικό instance_dict ως ζεύγος κλειδιού-τιμής με το κλειδί "name".

Εάν το instance_dict είναι ένα νέο λεξικό (δηλαδή δεν βρέθηκε στη λίστα instances), το προσθέτω στη λίστα instances.
Δημιουργώ ένα αντικείμενο DictVectorizer και το χρησιμοποιώ για να προσαρμόσω και να μετασχηματίσω τη λίστα instances σε έναν αριθμητικό πίνακα χαρακτηριστικών X.
Χρησιμοποιώ τη συνάρτηση cosine_similarity από το scikit-learn για να υπολογίσετε την ομοιότητα συνημίτονου μεταξύ όλων των ζευγών περιπτώσεων στον πίνακα χαρακτηριστικών.
Δημιουργώ μια κενή λίστα sse για να αποθηκεύσω τις τιμές SSE για κάθε πιθανή τιμή του k (ο αριθμός των συστάδων).
Επαναλάμβάνω ένα εύρος τιμών για το k από 2 έως 30. Για κάθε τιμή του k, κάνω τα εξής:
Δημιουργώ ένα αντικείμενο KMeans με n_clusters που έχει οριστεί στην τρέχουσα τιμή του k.
Χρησιμοποιώ τη μέθοδο fit() του αντικειμένου KMeans για να ομαδοποιήσω τις περιπτώσεις με βάση την ομοιότητά τους.
Χρησιμοποιώ το χαρακτηριστικό inertia_ του αντικειμένου KMeans για να υπολογίσω το SSE για την τρέχουσα τιμή του k και να το προσαρτήσω στη λίστα sse.
Χρησιμοποιώ το matplotlib για να σχεδιάσω τις τιμές στη λίστα sse σε σχέση με το εύρος τιμών για το k.
Δημιουργώ ένα αντικείμενο KMeans με n_clusters που έχει οριστεί σε 17 (περίπου επιλογή απο το γράφημα elbow).