ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Υλοποίηση Ολοκληρωμένου Συστήματος με Λέξεις-Κλειδιά σε Σημασιολογικά Δεδομένα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της Παπαϊωάννου Αικατερίνης

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2012

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Υλοποίηση Ολοκληρωμένου Συστήματος με Λέξεις-Κλειδιά σε Σημασιολογικά Δεδομένα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της Παπαϊωάννου Αικατερίνης

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Σεπτεμβρίου 2012.

.....	.....	.....
Τιμολέων Σελλής,	Ιωάννης Βασιλείου,	Θοδωρής Δαλαμάγκας,
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Ερευνητής Β΄ ΙΠΣΥ/Ε.Κ. “Αθηνά”

Αθήνα, Σεπτέμβριος 2012

.....
Αικατερίνη Δ. Παπαϊωάννου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο Σημασιολογικός Ιστός αποτελεί μια συλλογική προσπάθεια οργάνωσης της γνώσης ώστε να διατίθεται σε μια ελαφρώς δομημένη μορφή (κείμενο, ήχος, βίντεο) κατάλληλη για ανθρώπους - αναγνώστες ενώ ταυτόχρονα να μπορεί αποδοτικά να γίνεται διαχείριση, προσπέλαση και συντήρησή της από μηχανές. Η αναπαράσταση της πληροφορίας σε μορφή διαχειρίσιμη από μια μηχανή γίνεται με τη βοήθεια συγκεκριμένου μοντέλου ενώ η μετέπειτα ανάκτησή της απαιτεί τη χρήση ερωτημάτων γραμμένα σε κατάλληλη γλώσσα. Δεδομένου ότι η αναζήτηση στο Σημασιολογικό Ιστό εγγυάται πιο ακριβή αποτελέσματα, σκοπός της παρούσας έρευνας είναι να παρέχει στο χρήστη τη δυνατότητα να απολαμβάνει το όφελος αυτό χωρίς να πρέπει να γνωρίζει τις απαιτούμενες γλώσσες ερωτημάτων αλλά χρησιμοποιώντας λέξεις κλειδιά, όπως του επιτρέπουν οι σημερινές μηχανές αναζήτησης. Στην κατεύθυνση αυτή μελετούνται προτεινόμενες στη βιβλιογραφία μέθοδοι εφαρμογής των παραπάνω ερωτημάτων εξειδικεύοντάς τες στο μοντέλο RDF. Εντοπίζονται κοινά προβλήματα που παρουσιάζονται και επιχειρείται η επίλυσή τους ή η βελτίωση του τρόπου αντιμετώπισής τους ενώ υλοποιούνται και ελέγχονται ως προς την απόδοση οι βελτιωμένες μορφές των μεθόδων. Ακόμα, μελετάται η καταλληλότητα των διαφόρων μορφών αναπαράστασης των αποτελεσμάτων της αναζήτησης ώστε να αναπαρίστανται σε μορφές εύκολα κατανοητές και διαχειρίσιμες από το χρήστη, μέσα από ένα πλήρες σύστημα όπου μπορούν αργότερα να διεξαχθούν συμπεράσματα ως προς διαφορετικές πτυχές του προβλήματος της αναζήτησης.

Λέξεις-κλειδιά

λέξη-κλειδί, αναζήτηση, σημασιολογικός, RDF, dbpedia, γράφος, οντολογία

Abstract

The Semantic Web is a collaborative movement of knowledge organization so that it is available in a slightly structured form (text, sound, video) suitable for human readers and so that it can easily be controlled, accessed and maintained by machines. Information is represented in form understandable by a machine using a specific framework which guarantees more accurate search results. However, the progress of the corresponding query processing has been delayed due to the complexity of the underlying query languages. A critical gap has been created between semantic search and end users, who have been accustomed to the traditional keyword search for years as it represents an intuitive way of specifying information needs. As part of this research, state of the art methods for querying semi-structured data (databases, graphs) using keywords are implemented, adapted to RDF(s) data, which we have chosen to emphasize. Common issues are spotted and are either resolved or improved. The modified methods are implemented and each one's effectiveness is evaluated under a common setting. Different forms of search results representations, which include either a substructure of the graph containing all query keywords or a set of possible SPARQL queries matching the keyword query, are compared through a system ready to adapt, in an upcoming research, the search results with respect to the different aspects of the keyword search problem (e.g. diversification, personalization).

Keywords

keyword, search, semantic, RDF, dbpedia, graph, ontology

Ευχαριστίες

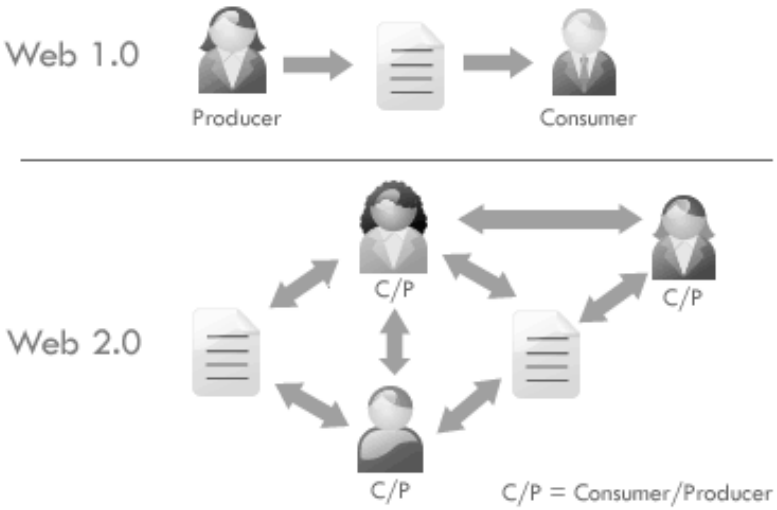
Για την εκπόνηση της παρούσας διπλωματικής εργασίας θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες προς τον επιβλέποντα καθηγητή κ. Τίμο Σελλή. Η βοήθειά του τη φετινή χρονιά ήταν ανεκτίμητη και η εμπιστοσύνη που μου έδειξε με βοήθησαν να προσπαθώ αδιάκοπα για ένα καλό αποτέλεσμα. Επιπλέον, σημαντική ήταν και η συμβολή του υποψήφιου διδάκτορα Γιώργου Γιαννόπουλου καθώς οι συμβουλές και οι διορθώσεις του αποτέλεσαν κίνητρο για συνεχή βελτίωση της δουλειάς.

Περιεχόμενα

Κεφαλαιο 1
Βαδίζοντας στο Σημασιολογικό Ιστό

1.1. Ο σημερινός Ιστός Αρχείων (Web of Documents)

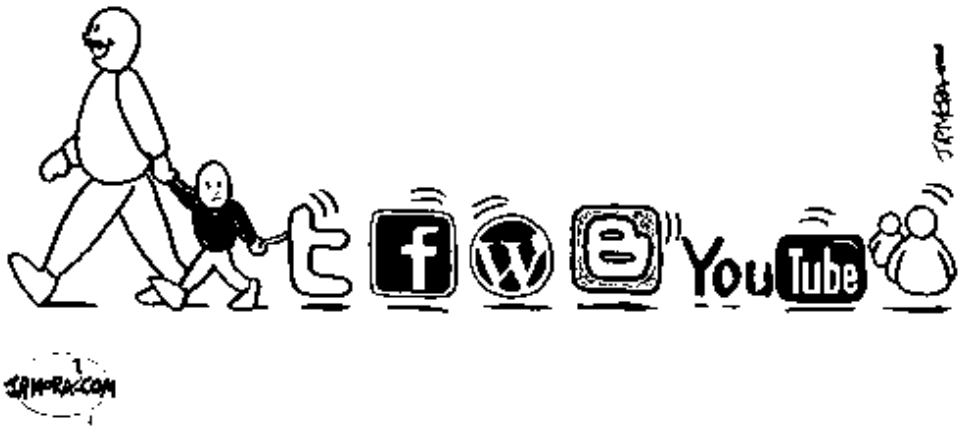
Στην απαρχή του, το διαδίκτυο μπορούσε να θεωρηθεί ένα σύνολο από Ιστοσελίδες (Web Pages) που προσέφεραν μια συλλογή από διαδικτυακά αρχεία (Web Documents) και στόχος του ήταν να προωθήσει το περιεχόμενό του στο ευρύ κοινό. Η επικοινωνία ήταν μίας κατεύθυνσης: οι χρήστες διάβαζαν οτιδήποτε ήταν διαθέσιμο και προσπαθούσαν να εντοπίσουν πληροφορίες που θα μπορούσαν να χρησιμοποιήσουν με μία πληθώρα τρόπων. Σήμερα, διανύουμε την εποχή του λεγόμενου Web 2.0 όπου το διαδίκτυο έχει γίνει πολύ πιο διαδραστικό διευκολύνοντας όλους του χρήστες του και δίνοντάς τους τη δυνατότητα να γίνουν αρωγοί στην παραγωγή της πληροφορίας παρά να μένουν απλοί καταναλωτές. (Εικόνα 1.1)



Εικόνα 1.1

Χαρακτηριστικό της εποχής αυτής του διαδικτύου αποτελεί η όλο και μεγαλύτερη ποσότητα από αυτό που θα χαρακτηρίζαμε ως περιεχόμενο παραχθέν από χρήστες (user-generated content) αλλά και πλήθος νέων εταιριών έχουν δημιουργηθεί βάσει της τάσης αυτής (Εικόνα 1.2). Συγκεκριμένα, αντί να αποτελούν απλούς αναγνώστες, οι χρήστες χρησιμοποιούν το διαδίκτυο για να παράγουν περιεχόμενο και να αλληλεπιδρούν χρησιμοποιώντας ιστοσελίδες κοινωνικής δικτύωσης (social networking sites). Ακόμα, όμως, και αν κάποιος δεν επιθυμεί να ακολουθήσει τα παραπάνω, υπάρχουν άλλοι τρόποι να απολαύσει τις παροχές του διαδικτύου όπως να συζητήσει (chat)[1] με τους φίλους του, να ψωνίσει (online shopping)[2], να πληρώσει τους λογαριασμούς του, να δει έναν αγώνα ποδοσφαίρου.

[1] http://en.wikipedia.org/wiki/Online_chat
[2] http://en.wikipedia.org/wiki/Online_shopping



Εικόνα 1.2

Πέρα, όμως, από διαφοροποίηση των δραστηριοτήτων που θα μπορούσε να ακολουθήσει ακόμα και ένας αρχάριος χρήστης, αλλαγές έχουν επέλθει και στη ζωή των προγραμματιστών διαδικτυακών εφαρμογών (Web Developers). Αντί να προσφέρουν αμιγώς στατικό περιεχόμενο στις εφαρμογές τους, είναι ικανοί να κατασκευάζουν ιστοσελίδες που μπορούν να εκτελέσουν πολύπλοκες συναλλαγές επιχειρήσεων, από την πληρωμή λογαριασμών μέχρι το κλείσιμο αεροπορικών εισιτηρίων και δωματίων σε ξενοδοχείο.

Τέλος, η τεράστια διάθεση περιεχομένου οδήγησε και στην ανάγκη ανταλλαγής και συνδυασμού αυτού για την αποδοτικότερη εκτέλεση εργασιών και την ικανοποίηση των χρηστών. Στην κατεύθυνση αυτή, όλο και περισσότερες ιστοσελίδες έχουν αρχίσει να δημοσιεύουν δομημένο περιεχόμενο ώστε διαφορετικές επιχειρησιακές οντότητες να μπορούν να κοινοποιούν το περιεχόμενό τους προκειμένου προσελκύσουν και να ολοκληρώσουν περισσότερες συναλλαγές. Για παράδειγμα, το Amazon και το eBay, δημοσιεύουν δομημένα δεδομένα μέσω standards Διαδικτυακών υπηρεσιών από τις βάσεις δεδομένων τους έτσι ώστε άλλες εφαρμογές να μπορούν να κατασκευαστούν χρησιμοποιώντας τα.

Λαμβάνοντας υπόψη τα παραπάνω, θα γίνει αναφορά σε τρεις βασικές χρήσεις του διαδικτύου, σε πιο υψηλό επίπεδο, και οι οποίες αφορούν τον τρόπο με τον οποίο καταναλώνουμε, ή αλλιώς αξιοποιούμε, τις πληροφορίες που είναι διαθέσιμες στον Ιστό. Σκοπός είναι να καταδειχθούν μερικά κρίσιμα ζητήματα που προκύπτουν και γίνει κατανοητή η ανάγκη αναδόμησής του.

Αναζήτηση Πληροφοριών (Searching)

Η αναζήτηση αποτελεί την πιο κοινή χρήση του Διαδικτύου και στόχος είναι ο εντοπισμός συγκεκριμένων πληροφοριών και πόρων (resources). Μια μηχανή αναζήτησης (search engine) αποτελεί το μέρος όπου ο χρήστης θα θέσει ερωτήματα που αφορούν γενικά θέματα (informational queries), για παράδειγμα *"αθλητισμός"*, ερωτήματα που προκύπτουν από την ανάγκη εντοπισμού μιας ιστοσελίδας (navigational queries), όπως *"youtube"*, αλλά και

ερωτήματα που υποδηλώνουν την πρόθεση του να εκτελέσει κάποια ενέργεια (transactional queries), όπως *"αγορά αυτοκινήτου"*.

Τα αποτελέσματα μίας αναζήτησης οργανώνονται σε σελίδες οι οποίες περιέχουν εικόνες, διευθύνσεις ιστοσελίδων (URL) και άλλα είδη αρχείων. Ο εντοπισμός των αποτελεσμάτων που ταιριάζουν με το ερώτημα του χρήστη ενδέχεται να είναι απλός, όπως στην περίπτωση εύρεσης διαφορετικών συνταγών για χοιρινό, ενώ υπάρχουν φορές που μπορεί να γίνει περίπλοκος. Ως παράδειγμα αναφέρεται η αναζήτηση πληροφοριών που αφορούν το Simple Object Access Protocol[3] με χρήση της λέξης-κλειδί "soap" καθώς από τα 300,000,000 αποτελέσματα που επιστρέφονται τα περισσότερα παραπέμπουν σε σαπούνια (soaps) και σαπουνόπερες (soap operas) αναγκάζοντας το χρήστη να ελέγξει αρκετές σελίδες αποτελεσμάτων προκειμένου να εντοπίσει εκείνο που ήθελε.

Ο λόγος για τον οποίο παρουσιάζονται καταστάσεις παρόμοιες με αυτή του παραδείγματος είναι ότι οι μηχανές αναζήτησης υλοποιούνται με βάση την κεντρική ιδέα *"ποια αρχεία περιέχουν τη δοθείσα λέξη-κλειδί;"*. Αν ένα αρχείο περιλαμβάνει τον όρο του ερωτήματος, θα συμπεριληφθεί στο σύνολο των υποψήφιων αποτελεσμάτων στο οποίο, πλέον, εφαρμόζονται μέθοδοι ταξινόμησης με βάση κριτήρια που καθορίζουν πόσο πιθανόν ένα αρχείο να ταιριάζει με τον επιδιωκόμενο σκοπό της αναζήτησης. Παρά την προσπάθεια εντοπισμού του βέλτιστου ταιριάσματος προσδοκώμενου και τελικού αποτελέσματος, συνεχίζει να πέφτει μεγάλο βάρος στο χρήστη προκειμένου να διαβάσει και να ερμηνεύσει το περιεχόμενο των διάφορων αρχείων και να αντλήσει κάθε χρήσιμη πληροφορία.

Ενσωμάτωση Πληροφοριών (Information Integration)

Η ενσωμάτωση πληροφοριών αφορά στον εντοπισμός πληροφοριών από διάφορες πηγές και ενδεχομένως σε διαφορετική μορφή με σκοπό τον συνδυασμό τους για την παραγωγή του επιθυμητού αποτελέσματος. Χαρακτηριστικό παράδειγμα αποτελεί η οργάνωση ενός πλάνου διακοπών όπου πρέπει, δεδομένων των ενδιαφερόντων και του οικονομικού προϋπολογισμού, να επιλεγεί το μέρος, ο τόπος διαμονής και να βρεθούν εισιτήρια ανάλογα με τον τρόπο μετάβασης. Η συλλογή των πληροφοριών αυτών απαιτεί ώρες αναζήτησης και συνδυασμό πολλών διαδικτυακών υπηρεσιών αλλά και δεν εγγυάται ότι οι τελικές επιλογές είναι οι βέλτιστες, για παράδειγμα ότι δεν υπάρχει πιο φθηνό ξενοδοχείο.

Είναι φανερό ότι μια διαδικασία σαν αυτή που περιγράφηκε θα ήταν πιο εύκολα υλοποιήσιμη με τη βοήθεια μιας εφαρμογής όπου θα δινόταν στο χρήστη η δυνατότητα να διευκρινίσει τις ανάγκες του και βάσει αυτών να του παρουσιάζονται ολοκληρωμένες προτάσεις συνοδευόμενες, πιθανόν, από άλλες χρήσιμες πληροφορίες. Μια τέτοια εφαρμογή θα παρείχε οφέλη από άποψης ταχύτητας, οργάνωσης των διαθέσιμων δεδομένων και ενώ θα ήταν εύκολα συντηρήσιμη.

[3] http://www.w3.org/TR/soap/

Εξόρυξη Δεδομένων στο Διαδίκτυο (Web Data Mining)

Η εξόρυξη δεδομένων αφορά στην καθόλου ευκαταφρόνητη εξαγωγή χρήσιμων πληροφοριών από μεγάλα και συνήθως κατανεμημένα σύνολα ή βάσεις δεδομένων. Στην περίπτωση που εφαρμόζεται στο Διαδίκτυο, δεδομένου του παραλληλισμού του με μια τεράστια κατανεμημένη βάση δεδομένων, σχετίζεται με τον εντοπισμό χρήσιμων πληροφοριών από έγγραφα που υπάρχουν σε αυτό. Πιθανόν να μην ακούγεται τόσο ενδιαφέρον όσο η αναζήτηση αλλά αποτελεί καθημερινότητα για όσους δουλεύουν ως αναλυτές, προγραμματιστές και για ερευνητικά κέντρα. Χαρακτηριστικό παράδειγμα αποτελεί η συγκέντρωση στοιχείων που αφορούν μια συγκεκριμένη μετοχή από όλες τις αξιόπιστες σελίδες για το χρηματιστήριο.

Εικόνα 1.3

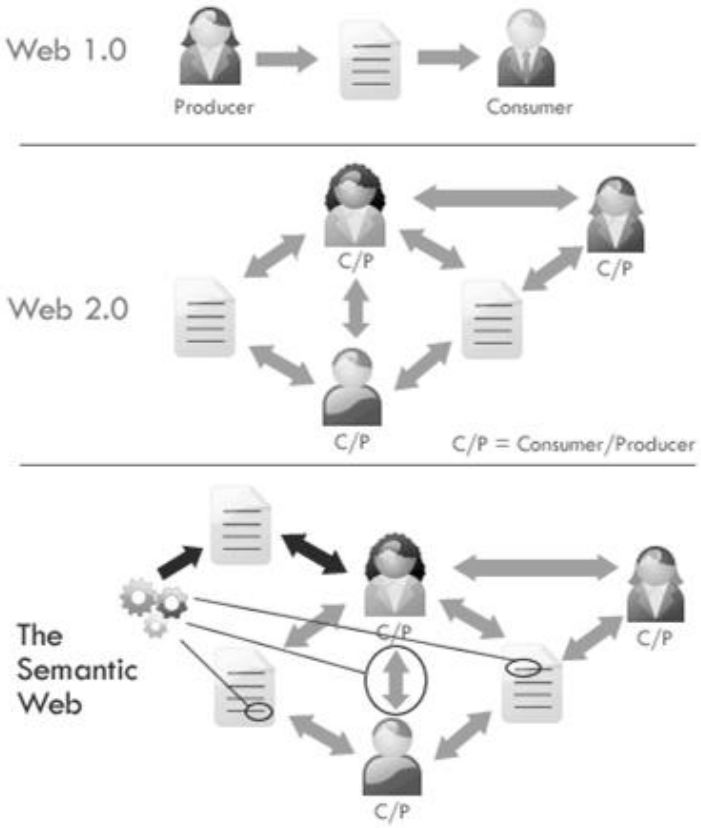
Μια εφαρμογή υπεύθυνη για εξόρυξη δεδομένων έχει το μειονέκτημα ότι δημιουργείται κατά περίπτωση, δηλαδή αναπτύσσεται προκειμένου να επισκέπτεται συγκεκριμένες σελίδες στον ιστό και να αντλεί από αυτές τις ζητούμενες πληροφορίες. Πιο σημαντικό, όμως, είναι το γεγονός ότι αν λάβει χώρα μια σημαντική αλλαγή σε κάποια από τις σελίδες που επισκέπτεται θα πρέπει να γίνει αναπρογραμματισμένης της εφαρμογής.

1.2. Η ιδέα ενός Ιστού Δεδομένων (Web of Data)

Σε καθεμία από τις χρήσεις του διαδικτύου που περιγράφηκαν εντοπίστηκε η ύπαρξη ενός σημείου μετά το οποίο απαιτούνταν η ενεργή συμμετοχή του χρήστη στον έλεγχο της καταλληλότητας των αποτελεσμάτων (αναζήτηση), στο συνδυασμό των πληροφοριών (ενσωμάτωση πληροφορίας) ή στην αναδόμηση εφαρμογών για συγκέντρωση χρήσιμων στοιχείων (εξόρυξη δεδομένων). Ο λόγος ύπαρξης του σημείου αυτού οφείλεται στο γεγονός ότι ο σημερινός Ιστός στοχεύει μόνο στην εξυπηρέτηση των ανθρώπινων αναγνωστών (human readers), όντας κυρίως προσανατολισμένος στην προβολή της πληροφορίας. Δίνεται, δηλαδή, περισσότερη έμφαση στην προβολή των πληροφοριών στο χρήστη παρά στην κατανόησή τους από τη μηχανή.

Ωστόσο, προσθέτοντας σε κάθε πόρο του ιστού νόημα κατανοητό από τον υπολογιστή του δίνουμε τη να εκτελέσει αυτόματα εργασίες για τις οποίες η ευθύνη έπεφτε το χρήστη. Η σκέψη αυτή οδήγησε στο ερώτημα "*Μήπως είναι δυνατό να αναδομήσουμε τον σημερινό ιστό έτσι ώστε το περιεχόμενό του να είναι σε μορφή ευκολότερα επεξεργάσιμη από μία μηχανή (machine processable)*;".

Ο Σημασιολογικός Ιστός αποτελεί μια πρωτοβουλία εμπνευσμένη από τον Tim Berners – Lee, τον ίδιο άνθρωπο που επινόησε τον Παγκόσμιο Ιστό στα τέλη της δεκαετίας του 1980, και προωθείται από την Κοινοπραξία Παγκόσμιου Ιστού (World Wide Web Consortium, W3C), ένα διεθνή οργανισμό προτυποποίησης για το διαδίκτυο. Σκοπός της πρωτοβουλίας αυτής είναι η τοποθέτηση της πληροφορίας στο επίκεντρο, όπως άλλωστε ήταν και το αρχικό όραμα, και η δόμηση της έτσι ώστε να μπορεί να είναι επεξεργάσιμη από υπολογιστές.



Εικόνα 1.4

Ο ιστός εξακολουθεί να είναι προσανατολισμένος στο χρήστη και η προσπάθεια ένταξης σημασιολογίας δεν προϋποθέτει την κατάργηση του τρόπου με τον οποίο αναπαρίσταται η πληροφορία σήμερα. Ωστόσο, δημιουργείται η ανάγκη σύνδεσης των διαθέσιμων δεδομένων (Εικόνα 1.4) και καταγραφής των παραγόμενων σχέσεων με τρόπο τέτοιο ώστε να παραχθεί ένα σύνολο δομημένων προτάσεων τις οποίες οι εφαρμογές θα μπορούν να συλλέγουν και να επεξεργάζονται.

1.3. RDF (Resource Description Framework)

Το Resource Description Framework (RDF) είναι ένα πρότυπο – πολλές φορές θα χαρακτηρίζεται και ως γλώσσα – που αναπαριστά πληροφορία που αφορά πόρους στον Παγκόσμιο Ιστό (web resources). Η κύρια χρήση του αφορά περιπτώσεις στις οποίες δίνουμε περισσότερη έμφαση στην επεξεργασία της πληροφορίας από εφαρμογές παρά στην απλή προβολή της σε ανθρώπινο χρήστη. Εξυπηρετεί στην ανταλλαγή αυτής μέσω εφαρμογών χωρίς να υπάρξει απώλεια νοήματος αλλά και στη χρήση της πέρα από τα όρια της εφαρμογής για την οποία αρχικά δημιουργήθηκε.

Το RDF προορίζεται κυρίως για την αναπαράσταση μεταδεδομένων (metadata) όπως ο τίτλος, ο συγγραφέας και η ημερομηνία τροποποίησης μιας ιστοσελίδας, πληροφορίες αδειοδότησης και πνευματικών δικαιωμάτων ενός δικτυακού αρχείου (web document) κτλ. Ωστόσο, επιχειρώντας μια γενίκευση του διαδικτυακού πόρου, διευκρινίζουμε ότι το εν λόγω πρότυπο μπορεί να χρησιμοποιηθεί προκειμένου να αναπαραστήσει πληροφορία για πράγματα που μπορούν να ταυτοποιηθούν στο διαδίκτυο ακόμα και αν δε μπορούν άμεσα να ανακτηθούν στον Ιστό. Μπορούν, δηλαδή να περιγραφούν άτομα (individuals), όπως ο Γαλιλαίος, είδη αντικειμένων, όπως Φιλόσοφοι, ιδιότητες αντικειμένων, όπως η απόσταση μιας πόλης από την πρωτεύουσα, και τιμές των ιδιοτήτων, οι οποίες μπορεί να αντιστοιχούν και σε συμβολοσειρές, τιμές από άλλους τύπους δεδομένων, όπως οι ακέραιοι.

1.3.1. Η βασική ιδέα

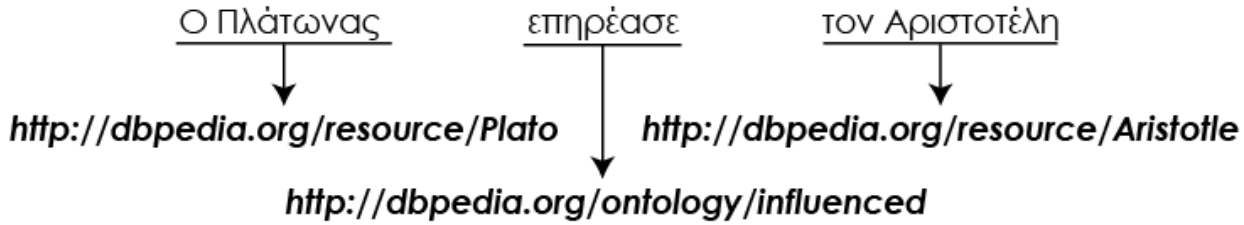
Κεντρική ιδέα αποτελεί η θεώρηση ότι όλες οι οντότητες μπορούν να περιγραφούν βάσει απλών ιδιοτήτων που τους συνδέουν με άλλες οντότητες που τους αποδίδουν χαρακτηριστικά. Δομικό στοιχείο στην περιγραφή αυτή είναι η πρόταση (statement) η οποία έχει συγκεκριμένη μορφή, αποτελούμενη από τη βασική συντακτική τριπλέτα (triple) υποκείμενο – κατηγόρημα – αντικείμενο. Το *υποκείμενο (subject)* είναι η οντότητα εκείνη στην οποία αναφέρεται η πρόταση. Ο όρος *κατηγόρημα (predicate)* αποδίδεται στην ιδιότητα (property) ή στο χαρακτηριστικό (characteristic) του υποκειμένου. Το *αντικείμενο (object)* άλλοτε είναι ένας πόρος με τον οποίο συνδέεται το υποκείμενο και άλλοτε αντιστοιχεί στην τιμή του χαρακτηριστικού στο οποίο αναφέρεται το κατηγόρημα. Παραθέτουμε το ακόλουθο παράδειγμα:

Εικόνα 1.5

1.3.2. Το Ενιαίο Αναγνωριστικό Πόρου (Uniform Resource Identifier, URI)

Η ανάγκη χρήσης του RDF από μηχανές προϋπέθετε την ύπαρξη ενός διαδικτυακού αναγνωριστικού (Web identifier) ικανό να ταυτοποίησει ένα αντικειμένου στον Ιστό χωρίς να

υπάρχει πιθανότητά σύγχυσής του με κάποιο άλλο. Ο Ενιαίος Εντοπιστής Πόρων (*Uniform Resource Locator*), γνωστός σε όλους ως *URL*, αντιστοιχεί σε μια συμβολοσειρά που χρησιμοποιείται για την αναγνώριση ενός πόρου στο διαδίκτυο αναπαριστώντας τον πρωταρχικό μηχανισμό πρόσβασης σε αυτό (ουσιαστικά, την θέση του στο ιστό). Ωστόσο, σκοπός του RDF είναι η περιγραφή οποιασδήποτε οντότητας στην οποία μπορεί να γίνει αναφορά σε μια πρόταση χωρίς να περιορίζεται σε πράγματα που έχουν δικτυακές θέσεις. Δημιουργήθηκε, λοιπόν, το Ενιαίο Αναγνωριστικό Πόρου (*Uniform Resource Identifier, URI)* εξασφαλίζοντας τη δυνατότητα μονοσήμαντης περιγραφής ενός πόρου.



Εικόνα 1.6

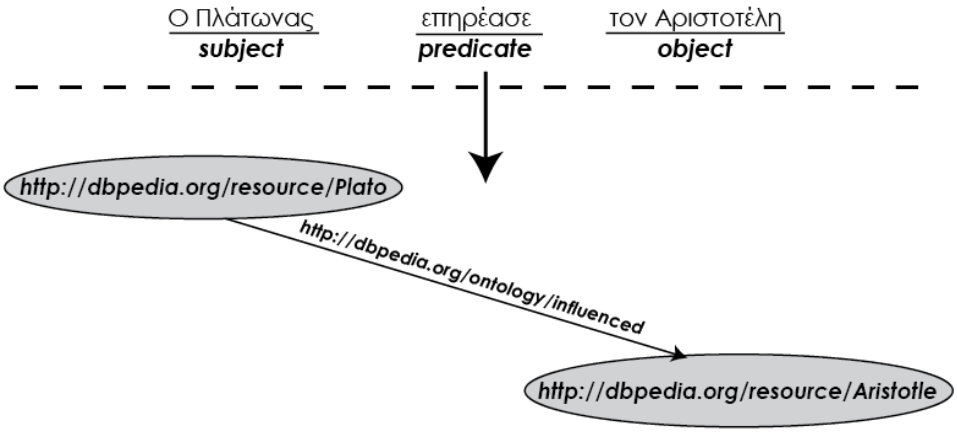
Σημαντική ιδιότητα του URI είναι ότι μπορεί ανεξάρτητα να δημιουργηθεί από διαφορετικούς ανθρώπους ή οργανισμούς γεγονός που το καθιστά λειτουργικό. Εμφανίζονται, όμως, ορισμένα προβλήματα στη διαδικασία ταυτοποίησης καθώς είναι πιθανό να δημιουργηθούν διαφορετικά αναγνωριστικά για τον ίδιο πόρο. Ενθαρρύνεται, για τον λόγο αυτό, η δημιουργία λεξιλογίων (vocabularies) με την έννοια των ομάδων από URI τα οποία προορίζονται για συγκεκριμένο σκοπό, για παράδειγμα η κατασκευή ενός συνόλου αναγνωριστικών για τους υπαλλήλους μίας εταιρείας. Ακόμη, ενθαρρύνεται περισσότερο η αναζήτηση του URI ενός πόρου σε υπάρχοντα λεξιλόγια πριν ο χρήστης προβεί στη δημιουργία νέου.

Κλείνοντας την υποενότητα αυτή, αξίζει να αναφέρουμε ότι η πλήρης καταγραφή του URI οδηγεί σε γραμμές μεγάλου μήκους κατά την καταγραφή των προτάσεων. Προς αποφυγή αυτού, έχει υιοθετηθεί ένας συντομότερος τρόπος γραψίματος σύμφωνα με τον οποίο κάθε URI μπορεί να όπως ένα Qname της XML.[4] Ένα Qname αποτελείται από ένα πρόθεμα (prefix) το οποίο αντιστοιχεί στο URI του τόπου ονομάτων, συνοδευόμενο από το σύμβολο "/", και από ένα τοπικό όνομα (local name). Ανακατασκευή του αρχικού URI επιτυγχάνεται προσαρτώντας το τοπικό όνομα στο πρόθεμα. Έτσι, αν το πρόθεμα foo έχει αντιστοιχηθεί στον τόπο ονομάτων http://example.org/somewhere/, το Qname foo:bar αποτελεί τη σύντομη μορφή του URI http://example.org/somewhere/bar.

[4] http://www.w3.org/XML/

1.3.3. Γραφοειδές μοντέλο του RDF (RDF's graph model)

Η δομική μονάδα του προτύπου RDF είναι η πρόταση (statement), μονάδα αρκετά ευέλικτη ώστε να μπορεί να περιγράψει οποιοδήποτε γεγονός σε μορφή εύκολα κατανοητή από τον υπολογιστή. Η έννοια της σύνδεσης σε μία πρόταση μπορεί να αναπαρασταθεί πιο απτά με τη βοήθεια ενός κατευθυνόμενου γραφήματος όπου το κατηγόρημα μοντελοποιείται ως βέλος (arc) με φορά από το υποκειμένου προς το αντικείμενό τα οποία αντιστοιχούν σε κορυφές (nodes). Η αναπαράσταση με μορφή γράφου της πρότασης που χρησιμοποιήθηκε στα προηγούμενα παραδείγματα φαίνεται στην ακόλουθη εικόνα:



Εικόνα 1.7

Ένα μεγαλύτερο σύνολο προτάσεων μπορεί με την ίδια ευκολία να αναπαρασταθεί σε μορφή κατευθυνόμενου γράφου ο οποίος θα αποτελείται από μεγαλύτερο αριθμό κόμβων και βελών. Η σημασία της ιδιότητας αυτής του RDF μοντέλου δεν αφορά τόσο την ανάγκη για απεικόνισή του όσο το ότι καθιστά καλύτερη την κατανόησή του, την εποπτεία των σχέσεων που το συνιστούν αλλά και δίνει τη δυνατότητα χρήσης αλγορίθμων της θεωρίας γραφημάτων προκειμένου να εκμεταλλευτούμε διάφορες εφαρμογές του.

1.4. RDFS (Resource Description Framework Schema)[5]

1.4.1. Η βασική ιδέα

Μετά την αναφορά που έγινε στο RDF παρατηρούμε ότι αποτελεί συντακτικό εφόδιο για την περιγραφή του "κόσμου", δηλαδή του συνόλου οντοτήτων που έχει επιλεγεί να περιγραφεί, με την έννοια ότι επιβάλλει οι προτάσεις που δημιουργούνται να αποτελούν τριάδες της μορφής υποκείμενο – κατηγόρημα – αντικείμενο. Προκύπτουν, όμως, εύλογα, ορισμένα ερωτήματα. Οποιεσδήποτε οντότητες μπορούν να συνδεθούν μεταξύ τους; Αν όντως είναι δυνατή η σύνδεσή

[5] www.w3.org/TR/rdf-schema

τους μπορεί να γίνει με οποιοδήποτε τρόπο; Δημιουργήθηκε, δηλαδή, η ανάγκη να ελέγχεται εάν η πρόταση που καταγράφεται για έναν πόρο είναι δυνατό να ειπωθεί.

Ένα πεδίο ενδιαφέροντος αποτελείται από αντικείμενα που μπορούν να ενταχθούν σε ομάδες τις οποίες ονομάζουμε κλάσεις (classes). Οι πόροι (resources) – μέλη κάθε κλάσης καλούνται στιγμιότυπά (instances) ενώ η ίδια αποτελεί επίσης πόρο καθώς ορίζεται μονοσήμαντα μέσω ενός URI και μπορεί να περιγραφεί από ιδιότητες που θα μελετηθούν αργότερα. Μια οντολογία περιγράφει ένα πεδίο ενδιαφέροντος με τυπικό τρόπο και αποτελείται από μια πεπερασμένη λίστα κλάσεων συνοδευόμενες από τις σχέσεις ιεραρχίας μεταξύ αυτών. Επιπλέον, οι οντολογίες μπορεί να περιέχουν πληροφορίες όπως ιδιότητες που συνδέουν δύο στιγμιότυπα (instances), καθορισμό του πεδίου ορισμού (domain) και του συνόλου τιμών (range) των ιδιοτήτων, διευκρινίσεις μη επικάλυψης κλάσεων, προδιαγραφές λογικών σχέσεων μεταξύ αντικειμένων.

Το RDFS (Resource Description Framework Schema) δημιουργήθηκε από την Κοινοπραξία Παγκόσμιου Ιστού (W3C) ως μια γλώσσα περιγραφής λεξιλογίου με σκοπό της διευκόλυνσης αποτύπωσης των οντολογιών σε μορφή κατανοητή από τον υπολογιστή. Ο χαρακτηρισμός αυτός έχει να κάνει με το ότι απαριθμεί τα κατηγορήματα που μπορούν να χρησιμοποιηθούν για τη σύνδεση οντοτήτων, τις κατηγορίες στις οποίες μπορούν να ενταχθούν οι οντότητες και τις ιεραρχικές μεταξύ τους σχέσεις. Ακόμα, υποδεικνύει, για κάθε ορισμένο κατηγόρημα, τις κατηγορίες στις οποίες πρέπει να ανήκει το υποκείμενο και το αντικείμενο από τα οποία συνοδεύεται.

1.4.2. Χρήσιμες ιδιότητες στο RDFS

rdfs: range

Η ιδιότητα αυτή χρησιμοποιείται σε προτάσεις όπου το υποκείμενο είναι μία ιδιότητα και το αντικείμενο είναι μία κλάση. Μία τριπλέτα (triple) της μορφής *P rdfs:range C* υποδηλώνει ότι στις προτάσεις που θα χρησιμοποιηθεί η ιδιότητα P ως κατηγόρημα, το αντικείμενο θα πρέπει να είναι στιγμιότυπο της κλάσης C.

rdfs: domain

Η ιδιότητα αυτή είναι συμπληρωματική της rdfs:range και, ομοίως, χρησιμοποιείται σε προτάσεις όπου το υποκείμενο είναι μία ιδιότητα και το αντικείμενο είναι μία κλάση. Μία τριπλέτα (triple) της μορφής *P rdfs:domain C* υποδηλώνει ότι στις προτάσεις που θα χρησιμοποιηθεί η ιδιότητα P ως κατηγόρημα, το υποκείμενο θα πρέπει να είναι στιγμιότυπο της κλάσης C.

rdfs:type

Μία τριπλέτα της μορφής *A rdfs:type B* δηλώνει ότι ο πόρος A είναι στιγμιότυπο της κλάσης B.

rdfs: subClassOf
Μέσω μίας τριπλέτας της μορφής *C1 rdfs:subClassOf C2* διατυπώνεται ότι κάθε στιγμιότυπο της κλάσης C1 ανήκει και στην κλάση C2.

rdfs: label
Μια τριπλέτα της μορφής *R rdfs:label L* δηλώνει ότι το λεκτικό (literal) L αποτελεί μία ταυτότητα του πόρου R αναγνώσιμη από ανθρώπινο χρήστη.

rdfs:comment
Κατά ανάλογο τρόπο με την ιδιότητα rdfs:label, η rdfs:comment παρέχει περιγραφή ενός πόρου κατανοητή από ανθρώπινο χρήστη. Χρησιμοποιείται κυρίως για να διευκρινίσει τη σημασία και τη χρήση κλάσεων και ιδιοτήτων.

1.5. OWL: Γλώσσα Οντολογιών Ιστού (Web Ontology Language)

1.5.1. Μετάβαση στην OWL

Η εκφραστικότητα των γλωσσών RDF και RDFS που περιγράψαμε στις προηγούμενες ενότητες είναι πολύ περιορισμένη και επιτρέπουν την αναπαράσταση μόνο ενός μέρους της οντολογικής γνώσης. Τα κύρια θεμελιώδη στοιχεία μοντελοποίησης των RDF/RDFS αφορούν την οργάνωση των λεξιλογίων σε τυποποιημένες ιεραρχίες (σχέσεις υποκλάσης και υποϊδιότητας), περιορισμό πεδίου ορισμού και συνόλου τιμών, και την καταγραφή των στιγμιότυπων κλάσεων και συνδέσεων μεταξύ αυτών. Ωστόσο, παρατηρήθηκε η έλλειψη δυνατοτήτων οι οποίες θα διευκόλυναν σημαντικά την εφαρμογή συλλογιστικής (reasoning) σε μία οντολογία. Ενδεικτικά αναφέρονται οι εξής:

Τοπική εμβέλεια
Η ιδιότητα rdfs:range ορίζει το σύνολο τιμών μίας ιδιότητας, δηλαδή την κλάση στις οποίες πρέπει να ανήκουν τα αντικείμενα των προτάσεων στις οποίες χρησιμοποιείται ανεξαρτήτως του υποκειμένου με αποτέλεσμα να μη μπορεί εφαρμοστεί ο περιορισμός στο σύνολο τιμών μίας ιδιότητας μόνο για επιλεγμένο αριθμό κλάσεων. π.χ. *Δεν μπορούμε να πούμε ότι οι αγελάδες τρώνε μόνο φυτά ενώ τα υπόλοιπα ζώα τρώνε και κρέας*.

Μη επικάλυψη κλάσεων
Παρότι μπορούν να καταγραφούν σχέσεις υποκλάσεων δεν είναι να δυνατό να ορίσουμε ότι δύο κλάσεις είναι ξένες μεταξύ τους, δεν έχουν δηλαδή κανένα κοινό στοιχείο. π.χ. *Δεν μπορούμε να εξασφαλίσουμε ότι κάθε αντικείμενο που ανήκει στην κλάση Άνδρας δε θα ανήκει και στην κλάση Γυναίκα.*

Λογικοί συνδυασμοί κλάσεων
Δεδομένου ότι μία κλάση ορίζεται από το σύνολο των στιγμιότυπων που ανήκουν σε αυτή, δεν είναι δυνατή η παραγωγή νέων κλάσεων χρησιμοποιώντας πράξεις συνόλων όπως η ένωση ή η τομή. *π.χ. Δεν μπορεί να οριστεί η κλάση Άνθρωπος ως η ένωση των κλάσεων Άνδρας και Γυναίκα.*

Περιορισμοί Πληθικότητας
Δε δίνεται η δυνατότητα επιβολής περιορισμών στο πλήθος των διακριτών τιμών που θα μπορούσε να πάρει μία ιδιότητα, δηλαδή, στον αριθμό των προτάσεων με κοινό υποκείμενο στις οποίες μπορεί να χρησιμοποιηθεί. *π.χ. Δεν υπάρχει τρόπος να καταγραφεί ότι ένα άτομο μπορεί να έχει ακριβώς δύο γονείς*.

Ειδικά χαρακτηριστικά ιδιοτήτων
Δεν γίνεται να δηλωθεί μία ιδιότητα ως μεταβατική η μοναδική ή αντίστροφη μιας άλλης ιδιότητας. π.χ. Δεν υπάρχει τρόπος να οριστεί ότι το ζεύγος ιδιοτήτων “influenced” και “influencedBy” είναι αντίστροφες.

Η ανάγκη ορισμού ενός προτύπου με μεγαλύτερη ευελιξία οδήγησε στην OWL η οποία σήμερα αποτελεί την πιο δημοφιλή γλώσσα οντολογιών. Ο σκοπός της δε διαφοροποιείται από εκείνον του RDFS: ο ορισμός οντολογιών που περιέχουν κλάσεις, ιδιότητες και μεταξύ τους σχέσεις για ένα συγκεκριμένο πεδίο ενδιαφέροντος. Το στοιχείο που την διαφοροποιεί είναι η δυνατότητά της να επιτρέπει τον ορισμό περίπλοκων και πλούσιων σχέσεων καθώς οδηγεί στη δημιουργία εφαρμογών με πολύ πιο ισχυρή συλλογιστική ικανότητα. Το RDFS παραμένει μια έγκυρη επιλογή αλλά ο προφανής περιορισμός που επιβάλλει το καθιστούν δεύτερη επιλογή.

1.6. Περιγραφή της OWL

1.6.1. Κεφαλίδα

Τα έγγραφα OWL αποκαλούνται συνήθως οντολογίες OWL και είναι έγγραφα RDF. Το στοιχείο – ρίζας τους είναι ένα στοιχείο rdf: RDF το οποίο καθορίζει έναν αριθμό από χώρους ονομάτων, όπως φαίνεται στο ακόλουθο παράδειγμα.[6]

```
<rdf:RDF
    xml:base="http://dbpedia.org/ontology/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns="http://dbpedia.org/ontology/">
```

[6] ΠΑΡΑΔΕΙΓΜΑ ΑΠΟ DBPEDIA OWL

1.6.2. Στοιχεία κλάσεων

Κάθε κλάση ορίζεται με τη χρήση ενός στοιχείου owl:Class ενώ δίνεται η δυνατότητα να διευκρινιστούν ισοδύναμες κλάσεις αλλά και ξένες προς αυτήν. Προκαθορισμένες κλάσεις είναι η owl:Thing, που περιέχει όλους τους πόρους, και η owl:Nothing, που είναι κενή. Σημειώνεται ότι κάθε κλάση είναι υποκλάση της owl:Thing και υπέρκλάση της owl:Nothing. Για παράδειγμα:

```
<owl:Class rdf:about="http://dbpedia.org/ontology/Philosopher">
        <rdfs:label xml:lang="en">philosopher</rdfs:label>
        <rdfs:label xml:lang="fr">philosophe</rdfs:label>
        <rdfs:label xml:lang="el">φιλόσοφος</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Person"> </rdfs:subClassOf>
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"></rdfs:subClassOf>
</owl:Class>
```

1.7. Στοιχεία Ιδιοτήτων

Στην OWL υπάρχουν δύο είδη ιδιοτήτων τα οποία επιτρέπουν και διαφορετικά είδη συνδέσεων. Οι ιδιότητες αντικειμένου (Object Properties) συσχετίζουν μεταξύ τους δύο στιγμιότυπα διαφορετικών ή και ίδιων κλάσεων. Για παράδειγμα:

```
<owl:ObjectProperty rdf:about="http://dbpedia.org/ontology/notableIdea">
        <rdfs:label xml:lang="en">notableIdea</rdfs:label>
        <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"></rdfs:domain>
</owl:ObjectProperty>
```

Οι ιδιότητες τύπου Δεδομένων (DataType Properties) συσχετίζουν ένα στιγμιότυπο μίας κλάσης με τιμή ενός τύπου δεδομένων. Σημειώνεται ότι η OWL δεν έχει προκαθορισμένους τύπους δεδομένων, ούτε και παρέχει ειδικές λειτουργίες ορισμού. Αντιθέτως, επιτρέπει τη χρήση των τύπων δεδομένων της γλώσσα XML Schema, αξιοποιώντας έτσι τη διαστρωματωμένη αρχιτεκτονική του Σημασιολογικού Ιστού. Ως παράδειγμα παραθέτουμε την ιδιότητα:

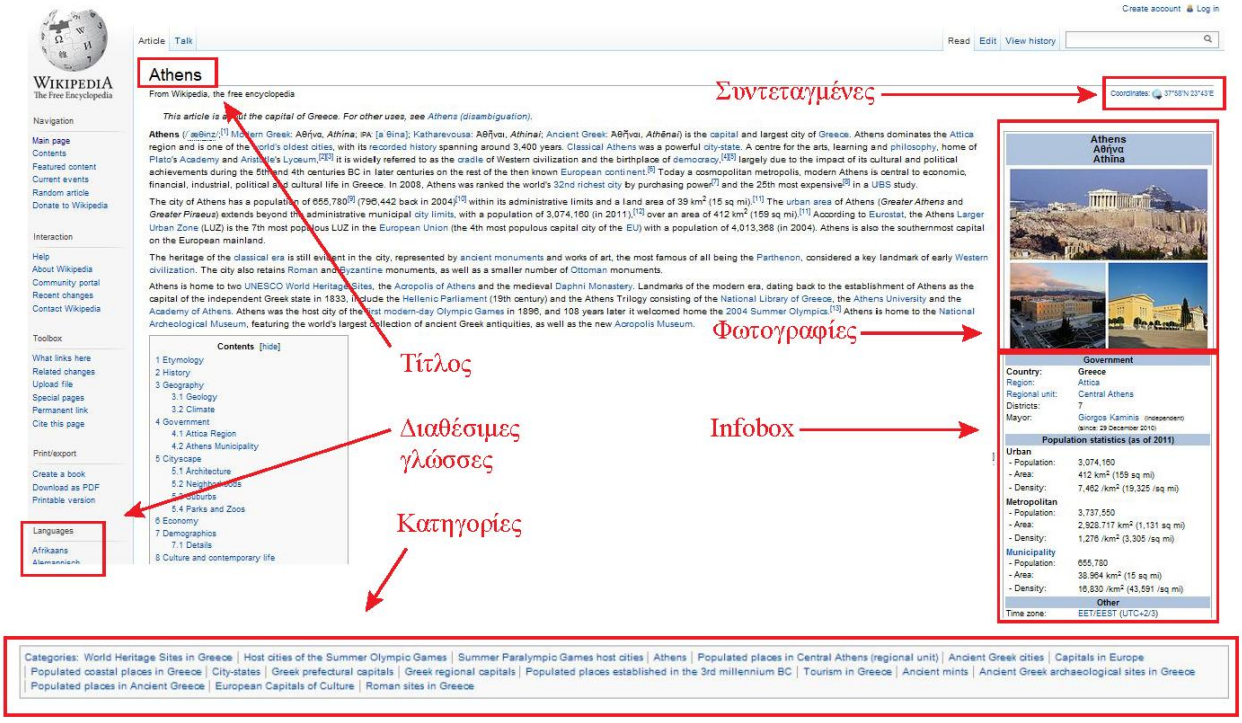
```
<owl:DatatypeProperty rdf:about="http://dbpedia.org/ontology/visitorsTotal">
    <rdfs:label xml:lang="en">visitors total</rdfs:label>
    <rdfs:label xml:lang="el">επιβατική κίνηση</rdfs:label>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/ArchitecturalStructure"></rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"></rdfs:range>
</owl:DatatypeProperty>
```

Κεφαλαιο 2
Το σύνολο δεδομένων και η προεπεξεργασία του

2.1. Το σύνολο δεδομένων

Η **Wikipedia**[7] αποτελεί ένα συλλογικό, παγκόσμιο, πολύγλωσσο εγκυκλοπαιδικό εγχείρημα που έχει συσταθεί στο Διαδίκτυο που λειτουργεί με την αρχή του wiki[8] γεγονός που τη διαφοροποιεί από κάθε παραδοσιακή εγκυκλοπαίδεια. Σύμφωνα με την αρχή αυτή, η Wikipedia δεν βασίζεται αποκλειστικά σε ειδικούς οι οποίοι δημιουργούν και δημοσιεύουν περιοδικά άρθρα, όπως για παράδειγμα η διαδικτυακή έκδοση της Britannica. Η ανάπτυξή της πηγάζει από την εμπιστοσύνη σε ανώνυμους χρήστες οι οποίοι διαρκώς και με μεγάλες ταχύτητες παράγουν περιεχόμενο. Έχει εξελιχθεί σε μια από τις βασικότερες πηγές γνώσης και έχει ως στόχο να παρέχει ελεύθερα επαναχρησιμοποιήσιμο περιεχόμενο, με αντικειμενικά και επαληθεύσιμα στοιχεία τα οποία ο καθένας μπορεί να τροποποιήσει και να βελτιώσει.

Παρότι η Wikipedia αποτελεί το πιο αντιπροσωπευτικό παράδειγμα συνεργατικά παραγόμενου περιεχομένου – 23 εκατομμύρια άρθρα, 285 γλώσσες, πάνω από 100 χιλιάδες συνεισφέροντες – στερείται δομημένου περιεχομένου. Η έλλειψη αυτή περιορίζει τις δυνατότητες της αναζήτησής της οι οποίες περιορίζονται στον ταύτιση μιας λέξης κλειδί με τον τίτλο ενός άρθρου ή μίας κατηγορίας. Ερωτήσεις όπως "Ποια πανεπιστήμια στην Ευρώπη έχουν ιδρυθεί πριν το 1980;" είναι πολύ δύσκολο να απαντηθούν καθώς προαπαιτείται η ύπαρξη ενός μηχανισμού ικανού να ενσωματώσει δεδομένα από διάφορα άρθρα.

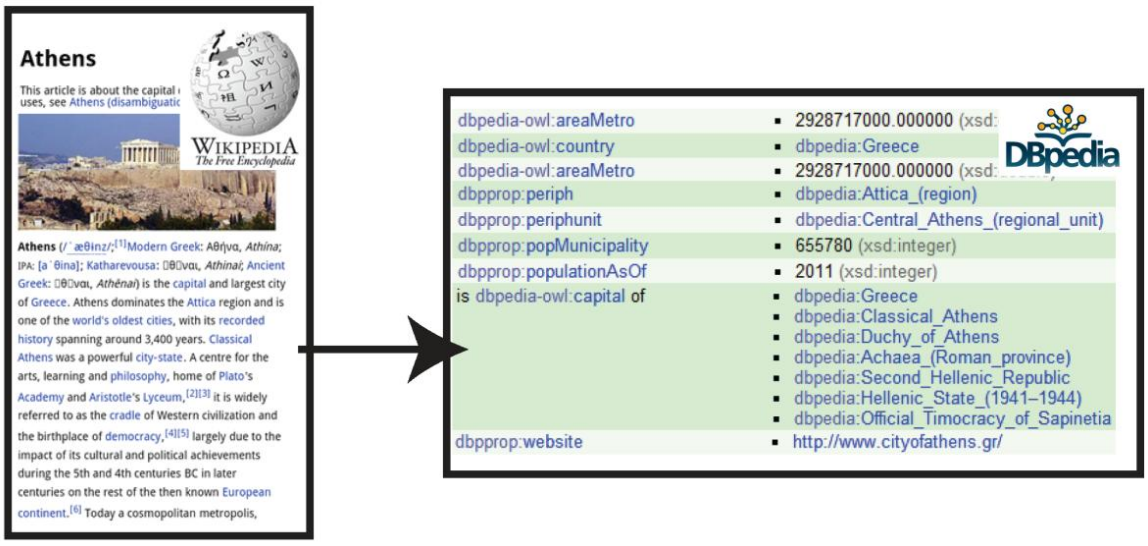


Εικόνα 2.1

[7] http://en.wikipedia.org/wiki/Main_Page

[8] http://wiki.org/wiki.cgi?WhatIsWiki

Αναζητώντας μορφές οργάνωσης της πληροφορίας παρατηρήθηκε ότι κάθε άρθρο αποτελείται κυρίως από ελεύθερο κείμενο ενώ παράλληλα περιλαμβάνει διαφορετικά είδη δομημένης πληροφορίας που αφορούν για παράδειγμα εικόνες, συντεταγμένες και συνδέσμους σε άλλες σελίδες, πίνακες infobox[9] ή στοιχεία κατηγοριοποίησης (Εικόνα 2.1). Ακόμα, τα διαθέσιμα άρθρα είναι κατανεμημένα βάση ενός συστήματος κατηγοριών που έχει εισαχθεί από τη Wikipedia με σκοπό την ευκολότερη πλοήγηση των χρηστών καθώς γνωρίζοντας βασικά χαρακτηριστικά του θέματος που επιθυμούν να μελετήσουν μπορούν εύκολα να βρουν ένα σύνολο σχετικών άρθρων.



Εικόνα 2.2

Η **DBpedia**[10] αποτελεί μια προσπάθεια με σκοπό την εξαγωγή των διαφόρων ειδών δομημένης πληροφορίας από την Wikipedia, όπως φαίνεται στην εικόνα 2.2, και το συνδυασμό τους σε μια τεράστια ενοποιημένη βάση γνώσης στην οποία θα μπορούν οι χρήστες να θέτουν ερωτήματα. Το έργο αυτό έχουν αναλάβει το Ελεύθερο Πανεπιστήμιο του Βερολίνου και το Πανεπιστήμιο του Leipzig σε συνεργασία με την OpenLink Software. Το εξαγώμενο σύνολο δεδομένων αναπαρίσταται με χρήση του Resource Description Framework (RDF) αντιστοιχίζοντας κάθε πόρος σε ένα άρθρο της Wikipedia ή σε μία κατηγορία άρθρων της. Δημοσιεύεται στο διαδίκτυο και, πέραν του ότι μπορούν να ότι μπορούν να τεθούν σε αυτό ερωτήματα, είναι δυνατόν να συνδεθεί με άλλα σύνολα δεδομένων που ακολουθούν το ίδιο πρότυπο.

[9] Ένα infobox είναι ένας καθορισμένης μορφής πίνακας σχεδιασμένος ώστε να τοποθετείται στην πάνω δεξιά γωνία ενός άρθρου προκειμένου να παρουσιάσει με συνέπεια μια περίληψη κάποιας ενοποιημένης πτυχής μεταξύ των άρθρων και για να βελτιώσει την πλοήγηση σε άλλα διασυνδεδεμένα άρθρα.

[10] http://wiki.dbpedia.org/About

Παρότι το DBpedia Data Set περιγράφει 3.64 εκατομμύρια «πράγματα» και μισό εκατομμύριο «γεγονότα», κατά την κατασκευή του συνόλου των δεδομένων που θα χρησιμοποιούνταν στα πειράματα επιλέχθηκαν προς επεξεργασία συγκεκριμένα αρχεία της DBpedia από τα οποία θα μπορούσαν να κατασκευαστούν συνεκτικοί γράφοι με μεγαλύτερο ενδιαφέρον. Τα εν λόγω αρχεία είναι κατά πλειοψηφία γραμμένα σε μορφή τριπλετών (triples) και είναι τα εξής:

A. instance_types_en.nt[11]

Στο αρχείο αυτό διευκρινίζονται οι κλάσεις (classes) στις οποίες ανήκουν τα υπάρχοντα στιγμιότυπα (instances). Σε κάθε RDF πρότασή του συναντάται ένα στιγμιότυπο στη θέση υποκειμένου και μία κλάση στη θέση αντικειμένου ενώ το κατηγόρημα σταθερά παραμένει http://www.w3.org/1999/02/22-rdf-syntax-ns#type.

π.χ. *http://dbpedia.org/resource/Toronto*
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://dbpedia.org/ontology/City

B. article_categories_en.nt[12]

Στο αρχείο αυτό καταγράφονται οι κατηγορίες (categories) στις οποίες ανήκουν τα υπάρχοντα στιγμιότυπα με τη βοήθεια του κατηγορήματος http://purl.org/dc/terms/subject. Σημειώνεται ότι οι κατηγορίες, όπως και οι κλάσεις, στοχεύουν στην οργάνωση των στιγμιοτύπων. Η διαφορά τους είναι ότι η ταξινόμηση στην οποία συμβάλλουν οι πρώτες είναι προϊόν της Wikipedia με σκοπό την διευκόλυνση των χρηστών ενώ η ύπαρξη των δεύτερων είναι εγγενώς καθορισμένη από το πρότυπο RDF.

π.χ. *http://dbpedia.org/resource/Logic*
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Axiology

Γ. mapping_based_properties_en.nt[13]

Στο αρχείο αυτό περιέχονται προτάσεις RDF όπου καταγράφονται σχέσεις μεταξύ στιγμιότυπων και σχέσεις μεταξύ στιγμιότυπων και λεκτικών ή τύπου δεδομένων.

π.χ. *http://dbpedia.org/resource/Aristotle*
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Ptolemy

[11] http://downloads.dbpedia.org/preview.php?file=3.8_sl_en_sl_instance_types_en.nt.bz2
[12] http://downloads.dbpedia.org/preview.php?file=3.8_sl_en_sl_article_categories_en.nt.bz2
[13] http://downloads.dbpedia.org/preview.php?file=3.8_sl_en_sl_mappingbased_properties_en.nt.bz2

http://dbpedia.org/resource/Animal_Farm
http://dbpedia.org/ontology/isbn
"ISBN 0-452-28424-4 (present) ISBN 978-0-452-28424-1"@en .

http://dbpedia.org/resource/Autism
http://dbpedia.org/ontology/omim
"209850"^^<http://www.w3.org/2001/XMLSchema#integer> .

Δ. labels_en.nt[14]

Στις RDF προτάσεις του αρχείου αυτού καταγράφεται η ετικέτα που έχει αντιστοιχηθεί σε κάθε διαθέσιμο πόρο που αφορά άρθρο της Wikipedia. Το αντικείμενο της πρότασης είναι λεκτικό (literal).

π.χ. *http://dbpedia.org/resource/AxiomOfChoice*
http://www.w3.org/2000/01/rdf-schema#label
"AxiomOfChoice"@en.

Ε. category_labels_en.nt[15]

Στις RDF προτάσεις του αρχείου αυτού καταγράφεται η ετικέτα που έχει αντιστοιχηθεί σε κάθε διαθέσιμο πόρο αφορά κατηγορία άρθρων της Wikipedia. Το αντικείμενο της πρότασης είναι λεκτικό (literal).

π.χ. *http://dbpedia.org/resource/Category:British_monarchs*
http://www.w3.org/2000/01/rdf-schema#label
"British monarchs"@en .

ΣΤ. dbpedia_3.7.owl[16]

Στο αρχείο αυτό είναι καταγεγραμμένη σε owl η οντολογία βάση της οποίας έχει κατασκευαστεί το σύνολο δεδομένων της dbpedia. Ορίζονται οι υπάρχουσες κλάσεις αλλά και οι ιδιότητες που μπορούν χρησιμοποιηθούν για σύνδεση δύο στιγμιοτύπων (Object Properties) ή για τη σύνδεση ενός στιγμιοτύπου με ένα λεκτικό (Datatype Properties) – στα πλαίσια της παρούσας εργασίας δε δόθηκε έμφαση στον τελευταίο τύπο ιδιοτήτων. Όπως φαίνεται και στα παραδείγματα που ακολουθούν, στον ορισμό κάθε κλάσης καταγράφεται η ετικέτα που της αντιστοιχεί, σε μία ή περισσότερες γλώσσες καθώς και οι υπερκλάσεις της. Ακόμα, στον ορισμό μίας ιδιότητας καταγράφεται μία ετικέτα, το πεδίο ορισμού και το σύνολο τιμών της. Όποτε είναι απαραίτητο, σημειώνεται κάποιο διευκρινιστικό σχόλιο. Ακολουθούν παραδείγματα για καθεμία από τις προαναφερόμενες περιπτώσεις.

[14] http://downloads.dbpedia.org/preview.php?file=3.8_sl_en_sl_labels_en.nt.bz2
[15] ΛΙΝΚ ΓΙΑ ΠΡΙΒΙΟΥ
[16] http://downloads.dbpedia.org/preview.php?file=3.8_sl_dbpedia_3.8.owl.bz2

```
<owl:Class rdf:about="http://dbpedia.org/ontology/GovernmentType">
    <rdfs:label xml:lang="en">Government Type</rdfs:label>
    <rdfs:label xml:lang="fr">régime politique</rdfs:label>
    <rdfs:label xml:lang="el">Είδη Διακυβέρνησης</rdfs:label>
    <rdfs:comment xml:lang="en">a form of government</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"></rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:about="http://dbpedia.org/ontology/parent">
    <rdfs:label xml:lang="en">parent</rdfs:label>
    <rdfs:label xml:lang="fr">parent</rdfs:label>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"></rdfs:domain>
    <rdfs:range rdf:resource="http://dbpedia.org/ontology/Person"></rdfs:range>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="http://dbpedia.org/ontology/mass">
    <rdfs:label xml:lang="en">mass (g)</rdfs:label>
    <rdfs:label xml:lang="el">μάζα (g)</rdfs:label>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"></rdfs:range>
</owl:DatatypeProperty>
```

2.1.1. Επεξεργασία το συνόλου δεδομένων

Η επεξεργασία του συνόλου δεδομένων της DBpedia πριν χρησιμοποιηθεί ως είσοδος στους αλγορίθμους που υλοποιήθηκαν κρίθηκε απαραίτητη, αρχικά, για την διευκόλυνση μετέπειτα προγραμματιστικών επιλογών. Οι συμβολοσειρές που αντιστοιχούσαν στα URI των πόρων της βάσης γνώσης καταλάμβαναν μεγάλες ποσότητες αποθηκευτικού χώρου και αποφασίστηκε να δημιουργηθεί ένας μηχανισμός γραφής τους με συμπτυγμένο τρόπο. Η κατασκευή ενός τέτοιο μηχανισμού απαιτούσε την διατήρηση μιας ενιαίας μορφής στο URI των υπαρχόντων στιγμιότυπων, κλάσεων και κατηγοριών γεγονός.

Στο αρχείο instance_types_en.nt παρατηρήθηκε ότι με την εξαίρεση της κλάσης *Thing* (*http://www.w3.org/2002/07/owl#Thing*) υπήρχαν δύο μορφές για το URI τον κλάσεων. Συγκεκριμένα, υπήρχαν κλάσεις με URI της μορφής *http://dbpedia.org/ontology/ClassName* ενώ εμφανίζεται, σε λιγότερες περιπτώσεις, η *http://schema.org/ClassName*, η οποία, όπως διαπιστώθηκε, με τη βοήθεια του αρχείου dbpedia_3.8.owl, χρησιμοποιούνταν για κλάσεις ισοδύναμες άλλων που περιγράφονταν με τον πρώτο τρόπο. Στην κατεύθυνση της υιοθέτησης ενός ενιαίου μοτίβου, επιλέχθηκε να αφαιρεθούν οι προτάσεις που αφορούσαν κλάσεις με URI της μορφής *http://dbpedia.org/ontology/*ClassName, καθώς όπως εξηγήθηκε δε θα υπήρχε απώλεια πληροφορίας.

Στόχος της εργασίας αυτής, όπως διευκρινίζεται και στη συνέχεια, δεν είναι τόσο ο εντοπισμός χρήσιμων πληροφοριών αλλά η δυνατότητα κατανόησης του τρόπου με τον οποίο συνδέονται αντικείμενα. Η χρήση των λεκτικών ως αντικειμένων και η ένταξή τους σε δομές της οντολογίας θεωρήθηκε δευτερευούσης σημασίας ενώ αποτέλεσαν κομβικό ρόλο σε τμήματα του αφορούσαν ανάκτηση πληροφορίας (information retrieval). Η επιλογή αυτή οδήγησε στην αφαίρεση των προτάσεων RDF που περιέχονταν στο αρχείο mappibg_based_properties_en.nt και που αφορούσαν ιδιότητες τύπων δεδομένων (DataType Properties) που συνδέουν ένα στιγμιότυπο με ένα λεκτικό, το οποίο δεν αντιστοιχεί στην ετικέτα του πόρου, ή με ένα αντικείμενο γνωστού τύπου δεδομένων.

Πέρα από τις περιπτώσεις που ήδη περιγράφηκαν, επιλέχθηκε να αφαιρεθούν, από όλα τα διαθέσιμα από την DBpedia αρχεία οι προτάσεις εκείνες που αναφέρονταν σε πόρους των οποίων ο URI, και κατά συνέπεια η ετικέτα, περιείχε χαρακτήρες όπως "*" ή ".", εμφανιζόμενους, διαδοχικά, μία ή περισσότερες φορές. Η διαδικασία ανάλυσης που εφαρμόζεται σε κάθε λέξη-κλειδί καθιστά αδύνατο την εισαγωγή ερωτήματος του οποίου κάποιος όρος να περιείχε τους παραπάνω χαρακτήρες ενώ ταυτόχρονα η επιθυμία του χρήστη να οδηγηθεί στην επιλογή αυτή φαίνεται να έχει μικρές πιθανότητες.

Κλείνοντας την παράγραφο αυτή, σημειώνεται ότι οι ενέργειες που περιγράφηκαν παραπάνω ανάγονται σε επεξεργασία συμβολοσειρών και υλοποιήθηκαν με τη βοήθεια κώδικα γραμμένου σε Perl. Η Perl[17] επιλέχθηκε καθώς διαθέτει πολύ ισχυρές και ευέλικτες τεχνικές επεξεργασίας κειμένου ενώ ταυτόχρονα επιτρέπει την αποφυγή γραφής μακροσκελών προγραμμάτων για το σκοπό αυτό.

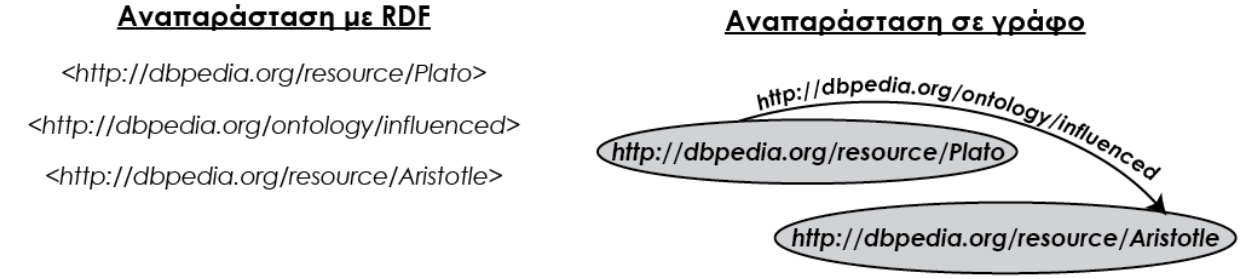
2.2. Αναπαράσταση Γράφου

Στα πλαίσια της εφαρμογής των αλγορίθμων που επιλέχθηκαν, ήταν απαραίτητη η αναπαράσταση τη βάση γνώσης, δηλαδή του δοθέντος συνόλου RDF προτάσεων, με μορφή γράφου. Το υποκείμενο και το αντικείμενο κάθε τριπλέτας (triple), αντιστοιχήθηκαν σε κορυφές του γράφου ενώ το κατηγόρημα αντιστοιχήθηκε σε ένα βέλος, φοράς από τον κόμβο του υποκειμένου προς εκείνον του αντικειμένου (Εικόνα 2.3). Στις τριπλέτες που αναφέρονταν στις ετικέτες πόρων προτιμήθηκε το αντικείμενο, το οποίο ήταν μια σταθερή τιμή (constant value), να μην αντιστοιχηθεί σε κόμβο αλλά να χρησιμοποιηθεί κατά την αντιστοίχηση πόρων με λέξεις-κλειδιά.

Λαμβάνοντας υπόψη τις παραδοχές αυτές, συγκεντρώσαμε σε ένα αρχείο (dbPediaStatements.nt) όλες τις RDF τριπλέτες που περιέχονταν στα αρχεία της DBpedia article_categories_en.nt, instance_types_en.nt και mapping_based_properties_en.nt ενώ παραλείψαμε να συμπεριλάβουμε το περιεχόμενο των category_labels_en.nt και labels_en.nt. Η παράλειψη αυτή έγινε καθώς διαπιστώθηκε ότι η προσθήκη των παραπάνω αρχείων δεν επέφερε μεταβολές στον αριθμό των κόμβων. Αυτό συνέβαινε αφενός διότι τα αντικείμενα των προτάσεων που περιείχαν ήταν σταθερές τιμές (literals), με αποτέλεσμα να μην αποτελούν

[17] http://www.perl.org/

κόμβους, και αφετέρου διότι τα αντίστοιχα υποκείμενα είχαν ενταχθεί στο γράφο μέσω κάποιου από τα συμπεριληφθέντα αρχεία.



Εικόνα 2.3

2.2.1. Δημιουργία ακέραιων αναγνωριστικών

Η κατασκευή του γράφου που περιγράφηκε προϋπέθετε την ύπαρξη ενός χαρακτηριστικού το οποίο θα επέτρεπε να αποφευχθεί η σύγχυση μεταξύ των κόμβων. Δεδομένου ότι κάθε κόμβος αντιστοιχήθηκε με έναν πόρο, όπως επέβαλε η γραφοειδής μορφή του RDF, η πρώτη σκέψη ήταν το ζητούμενο χαρακτηριστικό να ταυτιστεί με τον Unified Resource Indentifier (URI) ο οποίος προσδιορίζει μονοσήμαντα κάθε πόρο. Ωστόσο, ο URI αντιστοιχεί σε μια συμβολοσειρά (string) αρκετών χαρακτήρων και η ύπαρξη μεγάλου αριθμού κόμβων δημιουργούσε προβλήματα όσο αφορά την ευελιξία του γράφου αλλά και το χώρο που απαιτούνταν για την αποθήκευσή του. Παρουσιάστηκε, λοιπόν, η ανάγκη δημιουργίας ενός μηχανισμού ο οποίος θα αντιστοιχούσε το URI κάθε πόρου με ένα στοιχείο του συνόλου των ακεραίων καθώς αποτελούν έναν εύχρηστο τύπο δεδομένων και καταλαμβάνουν σταθερό χώρο στη μνήμη. Σε αυτή την κατεύθυνση, δημιουργήθηκε ένας πίνακα κατακερματισμού (Hash Table[18]), ο οποίος και υλοποιήθηκε με την κλάση HashMap[19] της Java και έδινε τη δυνατότητα να αποθηκευτούν οι διαθέσιμες συμβολοσειρές σε θέσεις κλειδιού (key) και ως τιμές (values) οι ακέραιοι με τους οποίους αντιστοιχούνταν.

Παρατηρήθηκε, όμως, ότι λόγω του μεγάλου όγκου των δεδομένων δημιουργήθηκαν εκ νέου προβλήματα ως προς τον απαιτούμενο χώρο αποθήκευσης του παραπάνω πίνακα κατακερματισμού και ως προς τη συνύπαρξή του στη μνήμη με άλλες δομές που χρειάζονταν. Αναζητώντας τρόπο να αποφευχθεί η διατήρηση της πλήρους συμβολοσειράς που αντιστοιχεί σε κάθε πόρο που περιγράφεται στο DBpedia Data Set, χρησιμοποιήθηκε ως βάση η ομοιομορφία στον τρόπο δόμησης του URI, για την οποία είχαν ήδη μπει θεμέλια κατά την επεξεργασία των διαθέσιμων αρχείων. Στον πίνακα που ακολουθεί καταγράφεται η μορφή που έχει το URI στις περιπτώσεις των στιγμιοτύπων, των κλάσεων, των κατηγοριών και των ιδιοτήτων. Με πλάγια

[18] http://en.wikipedia.org/wiki/Hash_table
[19] http://docs.oracle.com/javase/1.4.2/docs/api/java/util/HashMap.html

γραφή σημειώνεται το τμήμα του αναγνωριστικού που παραμένει αμετάβλητο ανεξάρτητα με το όνομα που έχει αποδοθεί στον πόρο.

Πόρος	Μορφή URI
Στιγμιότυπο	*http://dbpedia.org/resource/*Instance_Name
Κατηγορία	*http://dbpedia.org/resource/Category:*CategoryName
Κλάση	*http://dbpedia.org/ontology/*ClassName
Ιδιότητα	*http://dbpedia.org/ontology/*PropertyName

Όπως γίνεται αντιληπτό, ο URI έχει τη μορφή http://dbpedia.org/(.*?)/*Name* όπου το (.*?) τοποθετείται για να σηματοδοτήσει την ύπαρξη μιας υποσυμβολοσειράς η οποία δε θα μας απασχολήσει παρά μόνο όταν προκύψει η ανάγκη για ανακατασκευή. Η κατάληξη *Name* αντιστοιχεί στο τμήμα του URI που έπεται της τελευταίας εμφάνισης του συμβόλου «/» με εξαίρεση την περίπτωση των κατηγοριών όπου επιλέχθηκε να αντιστοιχηθεί στο τμήμα μετά το σύμβολο «:» προκειμένου να αποφευχθεί η αποθήκευση της λέξης «Category» η οποία δεν προσέφερε κάποια διαφοροποίηση. Σε αυτό το σημείο αξίζει να σημειώθεί ότι στην περίπτωση των στιγμιοτύπων και των κατηγοριών, η κατάληξη (suffix) μετά το σύμβολο «/» δεν αποτελεί μια τυχαία ακολουθία χαρακτήρων αλλά αποσπάται κάθε φορά από το URL του άρθρου ή της κατηγορίας άρθρων της Wikipedia που συνδέεται με τον συγκεκριμένο πόρο αντίστοιχα.[20]

Η παραπάνω παρατήρηση οδήγησε σε επιλογές που πιθανόν να παραπέμπουν στην αναπαράσταση ενός URI ως ένα Qname της XML, με την έννοια ότι η συμβολοσειρά διασπάται σε δύο μέρη, το πρόθεμα (prefix) και την κατάληξη (suffix). Βάσει της μορφής http://dbpedia.org/(.*?)/*Name*, διατηρήθηκε, για κάθε πόρο, ως κατάληξη το τμήμα *Name* συνοδευόμενο από ένα πρόθεμα ικανό να καταδείξει τον τύπο του. Συγκεκριμένα, όπως φαίνεται και στον πίνακα που ακολουθεί, χρησιμοποιήθηκαν τα προθέματα IN, CL, CA, PR για τα στιγμιότυπα, τις κλάσεις, τις κατηγορίες και τις ιδιότητες αντίστοιχα.

Παράδειγμα URI	Αναπαράσταση
(Στιγμιότυπο**)** *http://dbpedia.org/resource/*Abraham_Lincoln	IN: Abraham_Lincoln
(Κατηγορία) *http://dbpedia.org/resource/Category:*Ethics	CA: Ethics
(Κλάση) *http://dbpedia.org/ontology/*Philosopher	CL: Philosopher
(Ιδιότητα) *http://dbpedia.org/ontology/*mainInterest	PR: mainInterest

Η επιλογή χρήσης αντιπροσωπευτικών προθεμάτων προέκυψε, κατά κύριο λόγο, με σκοπό τη διευκόλυνσή κατά τη διεξαγωγή των πειραμάτων. Ωστόσο, αποδείχθηκε αναγκαία καθώς η αποκοπή της υποσυμβολοσειράς «Category» από το URI των κατηγοριών οδήγησε

[20] Αντλώντας το κατάλληλο τμήμα από το URI *http://dbpedia.org/resource/*Abraham_Lincoln, παράγουμε το URL http://en.wikipedia.org/wiki/Abraham_Lincoln το οποίο μας παραπέμπει στο άρθρο της wikipedia που αφορά τον Αβραάμ Λίνκολν.

στην ύπαρξη περιπτώσεων όπου μία κατηγορία και ένα στιγμιότυπο μοιράζονταν το ίδιο Name. Ως παράδειγμα αναφέρεται η περίπτωση των πόρων με URI http://dbpedia.org/resource/Aristotle και http://dbpedia.org/resource/Category:Aristotle για τους οποίους το Name είναι *Aristotle* παρότι ο πρώτος αντιστοιχεί σε κατηγορία και ο δεύτερος σε στιγμιότυπο.

Πρέπει να διευκρινιστεί ότι υπήρξαν τρεις περιπτώσεις πόρων των οποίων ο URI δεν ακολουθούσε τη μορφή που μέχρι στιγμής περιγράφηκε. Καθένας από αυτούς αντιστοιχήθηκε με μια συμβολοσειρά που να παραπέμπει στο νόημά τους, όπως φαίνεται στον ακόλουθο πίνακα, και η συμβολοσειρά αυτή με τη σειρά της αντιστοιχήθηκε με ένα ακέραιο.

URI	Συμβολοσειρά
http://www.w3.org/2002/07/owl#Thing	Thing
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	belongsTo
http://purl.org/dc/terms/subject	categAs

Τέλος, σημειώνεται ότι η ύπαρξη του μηχανισμού αντιστοίχησης μίας συμβολοσειράς σε έναν ακέραιο δημιούργησε την ανάγκη κατασκευή ανάλογου μηχανισμού προκειμένου να υλοποιείται η αντίστροφη διαδικασία. Δημιουργήθηκε, έτσι, ένας ακόμα πίνακας κατακερματισμού ο οποίος είχε ως κλειδιά τους ακεραίους και ως τιμές τις συμβολοσειρές.

2.2.2. Υλοποίηση του γράφου

Η υλοποίηση του γράφου της οντολογίας έγινε με τη βοήθεια της βιβλιοθήκης JgraphT[21], η οποία παρέχει αντικείμενα και αλγορίθμους της Θεωρίας Γραφημάτων. Αρχικά, επιλέχθηκε μία κλάση που καθόριζε το είδος των ακμών του γράφου, αν δηλαδή θα είχαν κατεύθυνση και αν θα συνοδεύονταν από ένα είδος βάρους ή ετικέτας. Η χρήση της κλάσης DefaultDirectedWeightedGraph[22], όπου ως βάρος στις ακμές/βέλη θα χρησιμοποιούνταν το αναγνωριστικό (ID) των αντίστοιχων ιδιοτήτων, δεν ήταν αποτελεσματική. Δημιουργούνταν προβλήματα κατά την ενσωμάτωση τριπλετών με κοινό υποκείμενο, κοινό αντικείμενο και διαφοροποιημένο κατηγόρημα καθώς η κλάση αυτή διευκρινίζει ότι επιτρέπεται μόνο ένα βέλος/ακμή μεταξύ δύο κόμβων. Για παράδειγμα, κατά την καταγραφή των τριπλετών

(α) http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Western_philosoph

(β) http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/region
http://dbpedia.org/resource/Western_philosophy

η προσθήκη της δεύτερης οδήγησε σε εξάλειψη της ακμής που είχε δημιουργηθεί χάρις την πρώτη. Προς αποφυγή τέτοιου είδους απωλειών, ο γράφος της οντολογίας υλοποιήθηκε ως

[21] http://jgrapht.org/
[22] http://jgrapht.org/javadoc/org/jgrapht/graph/DefaultDirectedWeightedGraph.html

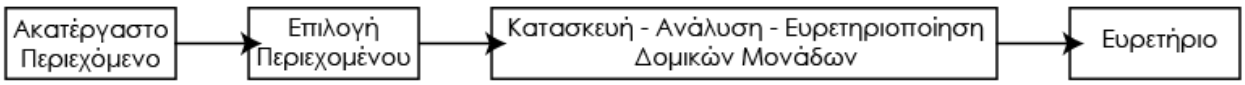
αντικείμενο της κλάσης DirectedMultigraph[23] η οποία παρείχε μεγαλύτερη ευελιξία. Παρότι υπήρχαν αντίστοιχες με αυτήν οι οποίες μεριμνούσαν για την ύπαρξης μιας μορφής ετικέτας ή βάρους των ακμών, δεν προτιμήθηκαν ενώ επιλέχθηκε το ζήτημα αυτό να καλυφθεί κατά την κατασκευή της κλάσης που θα αντιστοιχούσε στις ακμές/βέλη.

Μετά την ολοκλήρωση της διαδικασίας που περιγράφηκε στην προηγούμενη ενότητα, ήταν φανερό ότι κάθε κόμβος του γράφου, ο οποίος θα αντιστοιχούσε σε ένα πόρο της οντολογίας, θα αναπαρίσταντο με ένα αντικείμενο της κλάσης των ακεραίων ενώ έπρεπε να κατασκευαστεί ανάλογη κλάση για τις ακμές/βέλη. Δημιουργήθηκε η κλάση RDFTTriple και βασική απαίτηση ήταν η ύπαρξη ενός πεδίου όπου θα μπορούσε να καταχωρείται το αναγνωριστικό (ID) της αναπαριστώμενης ιδιότητας. Ωστόσο, δεδομένου ότι υπήρξαν στιγμές όπου έπρεπε να δοθεί βαρύτητα περισσότερο στην ύπαρξη της ακμής, χρειάστηκε να μετατραπεί ο γράφος σε μη κατευθυνόμενο με τη βοήθεια της κλάσης AsUndirectedGraph[24]. Η μετατροπής αυτή είχε ως συνέπεια την απώλεια της πληροφορίας που αφορούσε την κατεύθυνση του βέλους και η οποία ήταν απαραίτητο να διατηρείται. Προστέθηκαν, έτσι, δύο ακόμα πεδία όπου καταχωρούνταν τα αναγνωριστικά των πόρων που αντιστοιχούσαν στο υποκείμενο και το αντικείμενο της αναπαριστώμενης τριπλέτας προκειμένου να μπορεί άμεσα να εντοπίζεται η αφετηρία, το τέρμα μιας ακμής.

2.3. Αντίστοιχηση των λέξεων-κλειδιά με αντικείμενα του γράφου

Απαραίτητο στάδιο της αναζήτησης με λέξεις-κλειδιά σε δεδομένα με γραφοειδή δομή είναι η αντιστοίχηση των όρων του ερωτήματος του χρήστη με αντικείμενα του γράφου της βάσης γνώσης. Το στάδιο αυτό εντοπίστηκε σε όλους τους αλγορίθμους που είχαν επιλεγεί προς υλοποίηση και για τον λόγο αυτό κρίθηκε προτιμότερο να ενταχθεί στο στάδιο προεπεξεργασίας των δεδομένων. Με τον τρόπο αυτό εξασφαλιζόταν η καλύτερη οργάνωση του κώδικα αλλά και η μείωση του χρόνου εκτέλεσης σημαντικών διαδικασιών.

Ο μηχανισμός που θα συνέδεε μια λέξη – κλειδί με έναν αντικείμενο του γράφου, κατασκευάστηκε με τη βοήθεια της ευρέως διαδεδομένης βιβλιοθήκης Lucene[25], κατάλληλης για εφαρμογές ανάκτησης πληροφορίας (Information Retrieval). Ουσιαστικά, η Lucene έδωσε τη δυνατότητα προσθήκης λειτουργίας αναζήτησης στην πλατφόρμα που επρόκειτο να δημιουργηθεί και για να υλοποιηθεί αυτό έπρεπε να ακολουθηθούν συγκεκριμένα βήματα τα οποία περιγράφονται στην εικόνα 2.4.



Εικόνα 2.4

[23] http://jgrapht.org/javadoc/org/jgrapht/graph/DirectedMultigraph.html
[24] http://jgrapht.org/javadoc/org/jgrapht/graph/AsUirectedGraph.html
[25] http://lucene.apache.org/core/

2.3.1. Επιλογή περιεχομένου

Το στάδιο της επιλογής περιεχομένου αφορούσε τον εντοπισμό όλων των συμβολοσειρών που εμφανίζονται στη βάση γνώσης και που θα μπορούσαν να αποτελέσουν όρους ενός ερωτήματος του χρήστη. Η διαδικασία αυτή έγινε παράλληλα με τη διαδικασία αντιστοίχησης ακεραίων στους URIs των πόρων της οντολογίας και οδήγησε στη δημιουργία ενός πίνακα κατακερματισμού ο οποίος είχε ως κλειδιά τις εν δυνάμει λέξεις-κλειδιά και ως τιμές ακέραιους αριθμούς που είχαν ρόλο αναγνωριστικού.

Οι συμβολοσειρές που θεωρήθηκε ότι θα μπορούσαν να εισαχθούν από το χρήστη αντλήθηκαν, κατά κύριο λόγο, από τα αρχεία category_labels_en.nt και labels_en.nt όπου σε κάθε πρόταση RDF το υποκείμενο – ένα στιγμιότυπο ή μια κατηγορία – προβαλλόταν σε ένα κόμβο του γράφου ενώ το αντικείμενο αντιστοιχούνταν σε μια σταθερή τιμή (literal), όπως φαίνεται στα ακόλουθα παραδείγματα.

(α) http://dbpedia.org/resource/AxiomOfChoice
http://www.w3.org/2000/01/rdf-schema#label
"AxiomOfChoice"@en .

(β) http://dbpedia.org/resource/Category:Mathematics
http://www.w3.org/2000/01/rdf-schema#label
"Mathematics"@en .

Πέρα, όμως, από τις κατηγορίες και τα στιγμιότυπα ετικέτες διέθεταν οι κλάσεις. Οι ετικέτες των τελευταίων εντοπίζονταν στο αρχείο dbpedia_3.8.owl όπου, κατά τον ορισμό τους, αντιστοιχούνται με μία συμβολοσειρά, διατυπωμένη πολλές φορές σε περισσότερες από μία γλώσσες. Στα πλαίσια της παρούσας εργασίας, χρησιμοποιήθηκε μόνο η αγγλική εκδοχή προκειμένου να υπάρχει συμβατότητα και με τα υπόλοιπα αρχεία που χρησιμοποιήθηκαν και τα οποία είναι γραμμένα στη γλώσσα αυτή. Πρέπει να διευκρινιστεί ότι στο αρχείο της οντολογίας, κάθε ιδιότητα που ορίζεται συνοδεύεται από μια σταθερή συμβολοσειρά (literal) που την περιγράφει αλλά η αντιστοιχία αυτή, παρότι υπαρκτή, δε θα ληφθεί υπόψη καθώς έχει γίνει η παραδοχή ότι μετά από μια διαδικασία αναζήτησης με λέξεις – κλειδιά στην υπό μελέτη βάση γνώσης θα μελετούνται μόνο αποτελέσματα που, δεδομένης της γραφοειδούς δομής, θα παραπέμπουν τελικώς σε κόμβους του γράφου και όχι σε ακμές.

Μετά την ολοκλήρωση της προσπέλασης των αρχείων category_labels_en.nt και labels_en.nt, παρατηρήθηκε ότι υπήρχαν πόροι, που αντιστοιχούσαν σε στιγμιότυπα ή κατηγορίες, για τους οποίους δεν είχαν καταγραφεί RDF προτάσεις που να τους συνδέουν με κάποια ετικέτα. Αυτό, ουσιαστικά, σήμαινε ότι θα υπήρχαν κόμβοι του γράφου οι οποίοι δε θα αντιστοιχούνταν ποτέ με κάποια λέξη-κλειδί πιθανού ερωτήματος του χρήστη. Δεδομένου ότι κάτι τέτοιο δεν ήταν επιθυμητό, αποφασίστηκε ότι για τους πόρους στους οποίους δε γίνεται αναφορά σε ένα από τα προαναφερθέντα αρχεία, το τμήμα του URI τους που έπεται του τελευταίου συμβόλου «/», αφού υποστεί κάποιου είδους επεξεργασία (αντικατάσταση συμβόλων «_» με κενά), θα αναλάμβανε ρόλο ετικέτας. Ως παράδειγμα αναφέρεται ο πόρος με URI http://dbpedia.org/resource/Los_Angeles_International_Airport στον οποίο αντιστοιχήθηκε η ετικέτα “ Los Angeles International Airport ”.

2.3.2. Κατασκευή – Ανάλυση - Ευρετηριοποίηση Δομικών Μονάδων

Η εύρεση του τρόπου με τον οποίο συνδέονται οι όροι ενός ερωτήματος απαιτεί την προβολή τους σε αντικείμενα του γράφου της οντολογίας. Το στάδιο κατασκευής δομικών μονάδων αφορούσε σε πρώτο τόνο την επιλογή των μονάδων εκείνων που θα επιστρέφονταν ως αποτελέσματα της αναζήτησης. Στα πλαίσια της εργασίας αυτής, επιλέχθηκε να περιοριστεί η αντιστοίχηση αυτή σε κόμβους του γράφου και όχι σε ακμές καθώς διαπιστώθηκε ότι δημιουργούνταν έδαφος για μεγαλύτερη ποικιλία δοκιμών και ότι διεξάγονταν αποτελέσματα με μεγαλύτερο ενδιαφέρον. Ωστόσο, πρέπει να σημειωθεί ότι έχουν γίνει απαραίτητες ενέργειες έτσι ώστε η επιλογή αυτή να αλλάζει εύκολα.

Α. Κατασκευή Δομικών Μονάδων

Κάθε δομική μονάδα αναπαρίσταται με τη βοήθεια της κλάσης Document την οποία θα μπορούσαμε να παρομοιάσουμε σαν μία συλλογή από πληροφορίες που θα επιτρέψουν την σύνδεση ενός κόμβου του γράφου με μια λέξη-κλειδί. Οι πληροφορίες αυτές αποθηκεύονται σε αντικείμενα της κλάσης Field, συνδέονται με το Document το οποίο αφορούν και καθίστανται, έτσι, τα σημεία όπου ο μηχανισμός αναζήτησης της βιβλιοθήκης Lucene θα προσπαθούσε να εντοπίσει τις λέξεις–κλειδιά που θα εισήγαγε ο χρήστης.

Κατά την διαδικασία κατασκευής επιλέχθηκε η δημιουργία πεδίων που θα τοποθετούνταν σε κάθε δομική μονάδα ανεξάρτητα με τον αν αντιστοιχούσε σε κόμβο που αναπαριστούσε στιγμιότυπο, κλάση ή κατηγορία. Στο πεδίο *resID* αποθηκεύτηκε, βάση των πινάκων κατακερματισμού, το μοναδικό ακέραιο αναγνωριστικό που είχε αντιστοιχηθεί σε κάθε πόρο. Στο πεδίο *resType* καταχωρήθηκε ένας ακέραιος – 0 για τα στιγμιότυπα, 1 για τις κλάσεις, 2 για τις κατηγορίες – ικανός να καταδείξει τον τύπο του πόρου που αντιστοιχούσε στον κόμβο και στο πεδίο *contents* τοποθετούνταν όλες οι συμβολοσειρές με τις οποίες ο κόμβος συνδεόταν.

Στην κατεύθυνση δημιουργίας μιας εφαρμογής προσαρμοστικής, με περισσότερες επιλογές στην αναζήτηση, δημιουργήθηκαν και πεδία που διαφοροποιούνταν ανάλογα με τον τύπο του πόρου που αναπαριστούσε κάθε κόμβος. Στα Document των στιγμιοτύπων εμφανίζονταν τα επιπρόσθετα πεδία *categAs, belongsTo, hasDataTypeProp* τα οποία σηματοδοτούσαν τις κατηγορίες στις οποίες ανήκε, τις κλάσεις στις οποίες ανήκε και τις ιδιότητες μέσω των οποίων συνδεόταν με άλλα στιγμιότυπα. Η προσθήκη των πεδίων αυτών έδινε τη δυνατότητα, ανά πάσα στιγμή, μέσω μιας απλής αναζήτησης να συγκεντρωθούν όλοι οι κόμβοι που αντιστοιχούσαν σε πόρους που είτε ανήκαν σε μια επιλεγμένη κλάση ή κατηγορία είτε βρίσκονταν στο ένα άκρο μιας συγκεκριμένης ακμής. Ακόμα, στα Document των κλάσεων είχε τοποθετηθεί το επιπρόσθετο πεδίο *subClassOf* όπου καταχωρούνταν όλες τις υπερκλάσεις, όπως αυτές καταγράφονταν κατά τον ορισμό τους στο αρχείο της οντολογίας.

Κατά την προσπέλαση των αρχείων της DBpedia κάθε πόρος συναντιόταν περισσότερες από μία φορές με αποτέλεσμα οι πληροφορίες που τον αφορούσαν και οι οποίες έπρεπε να καταγραφούν να εντοπίζονται σε πολλά σημεία. Σύμφωνα με την αρχική προσέγγιση που είχε

υιοθετηθεί, είχαν κατασκευαστεί δομές, συγκεκριμένα πίνακες κατακερματισμού, που φιλοξενούσαν προσωρινά τις απαραίτητες πληροφορίες και αφού είχε ολοκληρωθεί η συγκέντρωση όλων, δηλαδή αφού είχαν προσπελαστεί όλα τα αρχεία, κατασκευάζονταν τα αντικείμενα της κλάσης Document.

Η διαδικασία αυτή ήταν χρονοβόρα και καταλάμβανε μεγάλη ποσότητα μνήμης με αποτέλεσμα να εγκαταλειφθεί. Στη θέση της επιλέχθηκε να χρησιμοποιηθεί η δυνατότητα ανανέωσης των δομικών μονάδων που κατασκευάζονταν και προσθήκη σε αυτές κάθε νέας πληροφορίας που προέκυπτε. Έτσι, την πρώτη φορά που κάθε πόρος-κόμβος συναντάται παράγεται το Document που του αναλογεί με τα στοιχεία που εκείνη τη στιγμή είναι διαθέσιμα ενώ αν ο ίδιος πόρος εμφανιστεί ξανά τότε το ήδη υπάρχον Document ανανεώνεται προκειμένου να ενσωματωθεί οποιοδήποτε νέο στοιχείο.

Β. Ανάλυση – Ευρετηριοποίηση Δομικών Μονάδων

Σκοπός του σταδίου της ανάλυσης είναι η επεξεργασία της πληροφορίας που διατίθεται στα αντικείμενα της κλάσης Field που απαρτίζουν ένα Document σε μορφή πιο ευέλικτη, η οποία θα ενισχύει τις δυνατότητες της ανάκτησης πληροφορίας. Τα πεδία *resID* και *resType* έχουν δημιουργηθεί με σκοπό τον μονοσήμαντο προσδιορισμό ενός Document αλλά και την ταυτοποίηση του κόμβου στον οποίο αντιστοιχεί. Περιέχουν ακέραιους αριθμούς και δεν επιδέχονται περαιτέρω τροποποιήσεις. Αντίθετα, το πεδίο *contents* αποτελεί το σημείο όπου είναι δυνατόν να εντοπιστούν οι συμβολοσειρές εκείνες που ενδέχεται να αποτελέσουν λέξεις-κλειδιά ενός ερωτήματος του χρήστη.

Η διαδικασία της ανάλυσης επιλέχθηκα να υλοποιηθεί με τη βοήθεια του EnglishAnalyzer[26] και αυτό είναι το μόνο σημείο παρέμβασης. Ο EnglishAnalyzer προϋποθέτει την ύπαρξη ενός συνόλου λέξεων (stop words) τις οποίες χαρακτηρίζει ως ασήμαντες και τις αγνοεί κάθε φορά που εμφανίζονται σε ένα Field. Το σύνολο αυτό ορίζεται από το χρήστη και τείνει να περιλαμβάνει λέξεις κοινότυπες στην αγγλική γλώσσα, που δε θα επέφεραν διαφοροποιήσεις σε μία αναζήτηση (π.χ. afterwards, because κτλ). Η υπολειπόμενη συμβολοσειρά, βάσει κανόνων του analyzer, διασπάται σε υποσυμβολοσειρές από τις οποίες αποκόπτονται οι καταλήξεις και διατηρείται μόνο το θέμα. Για παράδειγμα, η πρόταση *"The Academy of Athens was the first institution of higher learning in the Western World"* αφού υποστεί ανάλυση ανάγεται στην πρόταση "*academi athen institut higher learn western world*"

Ως έξοδο, η διαδικασία που περιγράφηκε παρέχει αντικείμενα της κλάσης Token[27]. Κάθε αντικείμενο της κλάσης αυτής συνδέεται με την εκάστοτε υποσυμβολοσειρά, που αναπαρίσταται ως αντικείμενο της κλάσης Term[28], συνοδευόμενο από δύο ακεραίους που αντιστοιχούν στη θέση (offset) του πρώτου και του τελευταίου χαρακτήρα του στην αρχική. Κάθε αντικείμενο της κλάσης Term αποτελεί την μονάδα της αναζήτησης. Αποτελείται από το κείμενο που αντιστοιχεί

[26] http://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene/analysis/en/EnglishAnalyzer.html
[27] http://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene/analysis/Token.html
[28] http://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene/index/Term.html

στο τμήμα της αρχικής συμβολοσειράς που αναπαριστά και από το όνομα του Field στο οποίο εντοπίστηκε.

Όσο αφορά το τμήμα της ευρετηριοποίησης αξίζει να γίνει αναφορά στο ότι η βιβλιοθήκη Lucene κατά την υλοποίηση της αναζήτησης χρησιμοποιεί το λεγόμενο ανεστραμμένο ευρετήριο (inverted index). Αποτελεί έναν τρόπο οργάνωσης όπου οι όροι (Terms) κατέχουν κεντρική θέση και επιτρέπεται άμεσος εντοπισμός των δομικών μονάδων (Documents) που περιέχουν καθέναν από αυτούς.

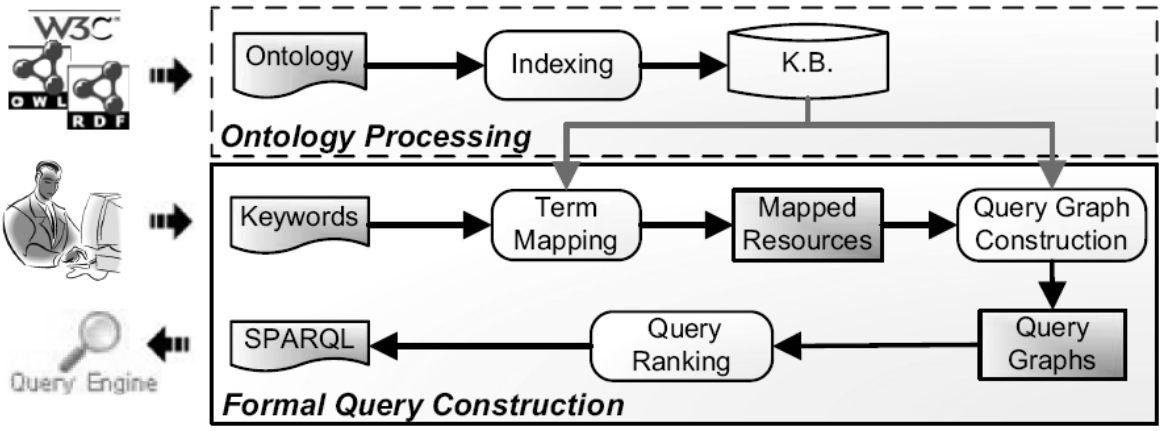
Κεφαλαιο 3

SPARK: Προσαρμόζοντας Ερωτήματα με λέξεις-κλειδιά στο Σημασιολογικό Ιστό

3.1. Η προσέγγιση του SPARK

Ο SPARK είναι ένα πρωτότυπο σύστημα που αναπτύχθηκε στην κατεύθυνση της προσαρμογής των ερωτημάτων με λέξεις-κλειδιά στην αναζήτηση στο Σημασιολογικό Ιστό. Βασίζεται στην διαπίστωση ότι η εύρεση της απάντησης σε ένα σημασιολογικό ερώτημα μπορεί να αναχθεί σε πρόβλημα εντοπισμού μιας ομάδας αντικειμένων που συνδέονται μεταξύ τους με συγκεκριμένες σχέσεις και περιορισμούς. Έτσι, σε μια οντολογία, ένα σημασιολογικό ερώτημα ισούται με ένα γράφο όπου επιβάλλονται περιορισμοί στις κορυφές του που αντιστοιχούν σε αντικείμενα – στιγμιότυπα (instances), κλάσεις (classes) – και βέλη του που αντιστοιχούν σε ιδιότητες.

Στην αρχική μορφή του, αποτελείται από δύο βασικές μονάδες (Εικόνα 3.1): την μονάδα επεξεργασίας της οντολογίας (ontology processing module) και τη μονάδα κατασκευής τυπικών ερωτημάτων (formal query construction module). Η μονάδας επεξεργασίας της οντολογίας είναι υπεύθυνη αρχικά για τη ευρετηριοποίηση (Indexing) των πόρων της οντολογίας όπου, με τη βοήθεια τεχνικών ανάκτησης πληροφορίας (Information Retrieval), καθένας από αυτούς αντιστοιχείται με συμβολοσειρές που θα μπορούσαν να αποτελέσουν όρους ερωτήματος του χρήστη. Στη συνέχεια, αναλαμβάνει την κατασκευή της βάσης γνώσης, δηλαδή, του κατευθυνόμενου γράφου που βασίζεται στην επιλεγμένη οντολογία. Η μονάδα κατασκευής τυπικών ερωτημάτων δέχεται λέξεις-κλειδιά ως είσοδο και δίνει ως έξοδο μια ταξινομημένη λίστα ερωτημάτων σε SPARQL.

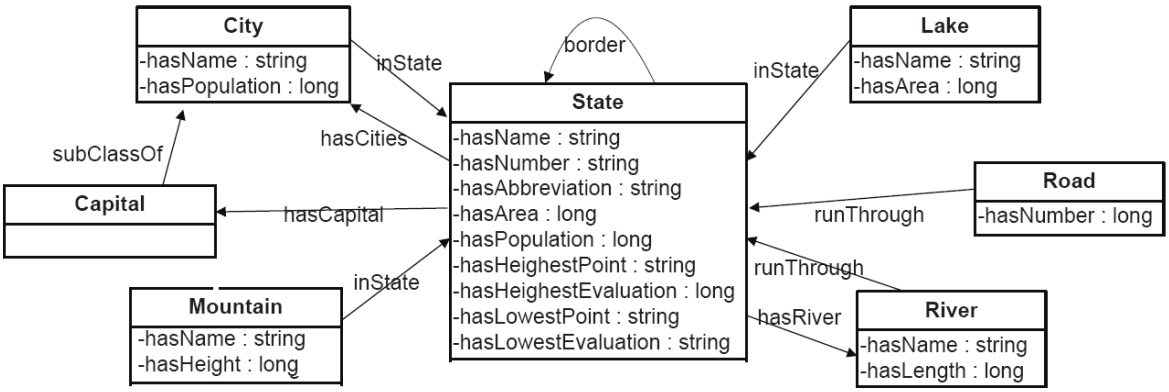


Εικόνα 3.1

Μόλις ο χρήστης εισάγει ένα ερώτημα αποτελούμενο από λέξεις-κλειδιά, το στάδιο της αντιστοίχησης όρων (term mapping) αναλαμβάνει να εντοπίσει τους κόμβους του γράφου της οντολογίας που συνδέονται με κάθε όρο του ερωτήματος. Στη συνέχεια, το στάδιο κατασκευής γράφων ερωτημάτων (query graph construction) απαριθμεί όλους τους πιθανούς συνδυασμούς από κόμβους, όπως αυτοί προέκυψαν στο προηγούμενο στάδιο. Για κάθε συνδυασμό, εφαρμόζοντας αλγόριθμο εύρεσης Ελάχιστου Συνδετικού Δέντρου, προσπαθεί να κατασκευάσει

πλήρεις γράφους, με διαφορετικές ερμηνείες, που να συνδέουν κόμβους του συνδυασμού. Τέλος, το στάδιο της κατάταξης ερωτημάτων (query ranking) αξιολογεί τα τυπικά ερωτήματα που κατασκευάστηκαν και τα επιστρέφει ταξινομημένα στο χρήστη. Στις επόμενες ενότητες, θα γίνει αναλυτική αναφορά στα στάδια *αντιστοίχησης όρων* και *κατασκευής γράφων ερωτημάτων* ενώ δε θα το στάδιο *κατάταξης ερωτημάτων* καθώς δεν συμπεριλήφθηκε στην τροποποιημένη μορφή του αλγορίθμου η οποία και υλοποιήθηκε.

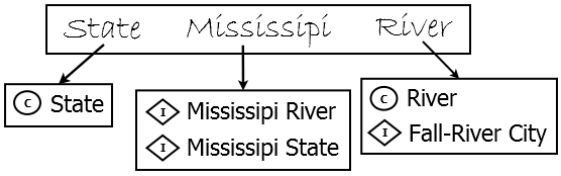
Στην ανάλυση που ακολουθεί, προκειμένου να γίνουν πιο κατανοητοί οι όροι και οι διαδικασίες που παρουσιάζονται, θεωρήθηκε απαραίτητο να χρησιμοποιηθεί ένα παράδειγμα στο οποίο θα εφαρμόζονται οι διαδικασίες που κάθε φορά περιγράφονται. Το σχήμα (schema) της οντολογίας του παραδείγματος απεικονίζεται στην εικόνα 3.2 και το ερώτημα που θα θεωρείται ότι έχει εισαχθεί από το χρήστη είναι το "*state mississippi river"*. Αποτελείται μόνο από τρεις (3) όρους, για λόγους απλότητας, και θεωρείται ότι ο χρήστης επιδιώκει να λάβει απαντήσεις ανάλογες με εκείνες που απαντούν στην ερώτηση *"Find all the states that the Mississippi river runs through"*.



Εικόνα 3.2

A. Αντιστοίχηση Όρων (Term Mapping)

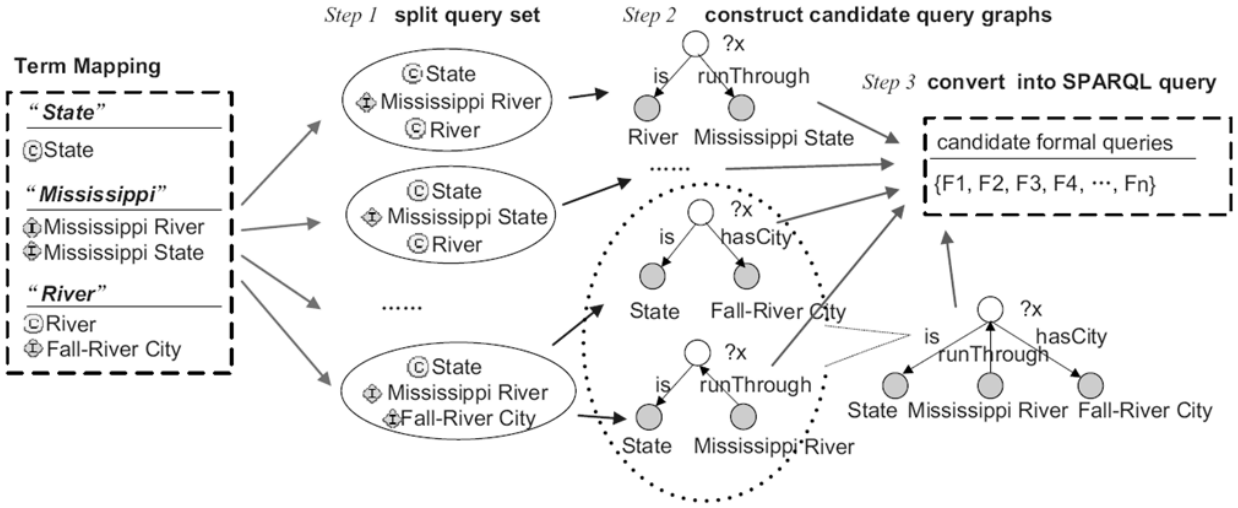
Ο ρόλος της διαδικασίας Αντιστοίχησης Όρων είναι να εντοπίσει τους πόρους (resources) της οντολογίας – κλάσεις (classes), κατηγορίες (categories), στιγμιότυπα (instances) – που αντιστοιχούν σε κάθε όρο του ερωτήματος που έθεσε ο χρήστης (Εικόνα 3.3). Στη συγκεκριμένη υλοποίηση, χρησιμοποιούνται δύο μέθοδοι αντιστοίχησης: i) *μορφολογική αντιστοίχηση (morphological mapping)*, όπου εφαρμόζονται τεχνικές σύγκρισης συμβολοσειρών (stemming, Sub-String κτλ) ii) *σημασιολογική αντιστοίχηση (semantic mapping)*, όπου με τη βοήθεια λεξικών για τον εντοπισμό λέξεων που σχετίζονται σημασιολογικά (π.χ. συνώνυμα). Γίνεται αντιληπτό, λοιπόν, ότι η εν λόγω διαδικασία αντιστοιχεί κάθε λέξη-κλειδί με στοιχεία της βάσης γνώσης με αποτέλεσμα, μετά την ολοκλήρωσή της, κάθε όρος του ερωτήματος να μην αντιστοιχεί σε μια συμβολοσειρά αλλά να αντιμετωπίζεται ως ένα σύνολο πόρων που υποδεικνύουν τι είδους στοιχεία επιθυμεί ο χρήστης.



Εικόνα 3.3

Β. Κατασκευή Γράφων Ερωτημάτων (Query Graph Construction)

Η διαδικασία κατασκευής γράφων κατασκευάζει υποψήφιους γράφους ερωτημάτων οι οποίοι περιλαμβάνουν τους πόρους που εντοπίστηκαν κατά την αντιστοίχηση όρων. Αρχικά, οι πόροι αυτοί χωρίζονται σε διαφορετικά σύνολα ερωτημάτων (query sets). Στη συνέχεια, εφαρμόζεται ο Αλγόριθμος Ελάχιστου Συνδετικού δέντρου σε καθένα από αυτά για να παραχθούν οι αντίστοιχοι γράφοι. Τέλος, κάθε γράφος ερωτήματος ερμηνεύεται ως ένα SPARQL ερώτημα βάσει κανόνων μετατροπής. (Εικόνα 3.4)

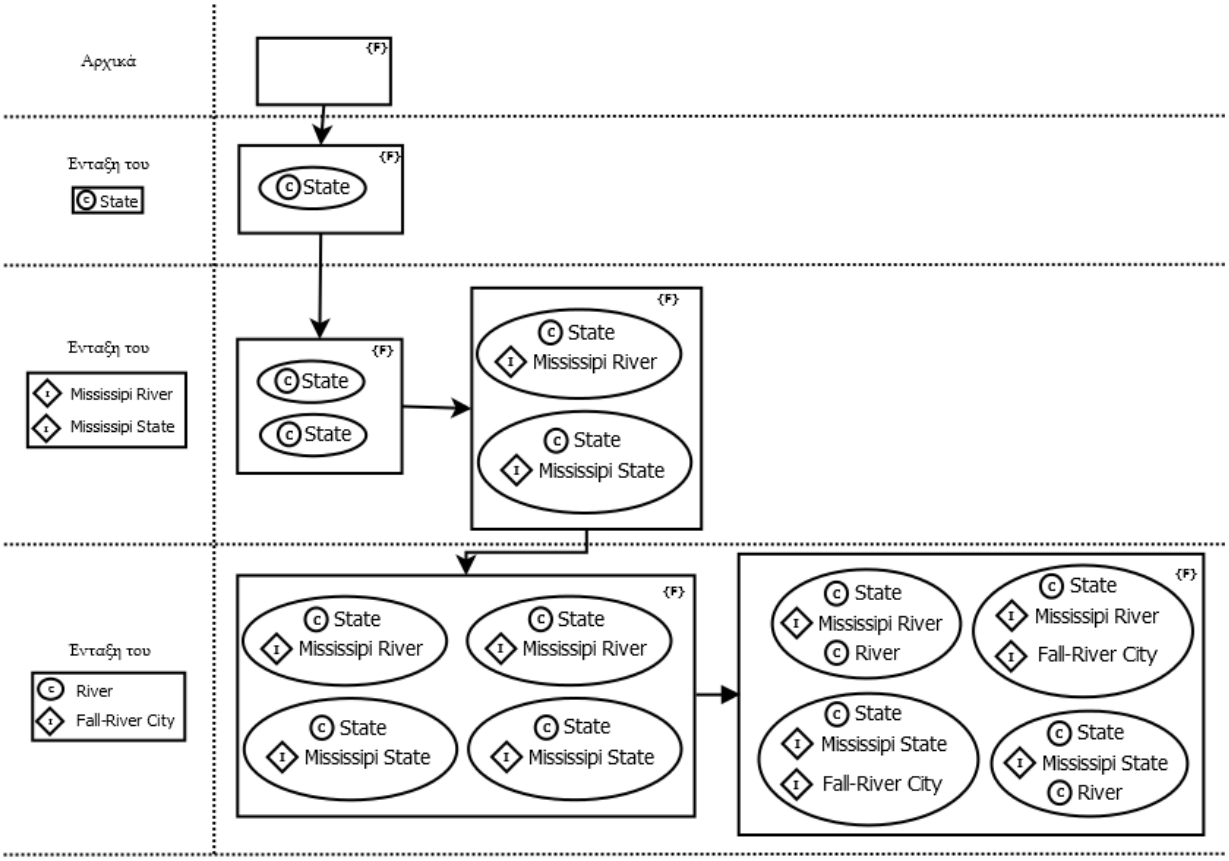


Εικόνα 3.4

Η διαδικασία διαχωρισμού των συνόλων ερωτημάτων (query set split) είναι η εξής: Δημιουργούμε το σύνολο {F} όπου θα τοποθετείται κάθε συνδυασμός F που παράγεται. Αν ο όρος t_i του ερωτήματος K αντιστοιχείται μόνο με έναν πόρο P, τοποθετούμε τον P σε κάθε σύνολο ερωτήματος F· αλλιώς, αν υπάρχουν N πόροι με τους οποίους ο t_i συνδέεται δημιουργούμε N αντίτυπα των στοιχείων του {F} και αντιστοιχούμε μονοσήμαντα έναν πόρο σε κάθε αντίτυπο. (Εικόνα 3.5)

Παρατηρώντας τον τρόπο κατασκευής των συνόλων ερωτημάτων, διαπιστώνουμε ότι περιγράφεται ουσιαστικά η κατασκευή του *καρτεσιανού γινομένου* των συνόλων που προέκυψαν στο στάδιο αντιστοίχησης όρων. Ανεξάρτητα, όμως, από το ποια μέθοδος θα χρησιμοποιηθεί, σκοπός είναι να απαριθμηθούν όλοι οι δυνατοί συνδυασμοί και να παραχθούν στη συνέχεια

γράφοι που να καλύπτουν όλες τις διαφορετικές ερμηνείες των λέξεων-κλειδιά που εισήγαγε ο χρήστης.



Εικόνα 3.5

Όταν δεν υπάρχουν πια νέα στοιχεία να εντάξουμε στο σύνολο {F}, ο SPARK χρησιμοποιεί τον αλγόριθμο του Kruskal για εύρεση ελάχιστου συνδετικού δέντρου προκειμένου να δημιουργηθεί τελικά για κάθε σύνολο ερωτήματος F ένας κατευθυνόμενος υπογράφος της βάσης γνώσης που θα περιέχει όλους τους πόρους του F. Αν μετά τη διαδικασία αυτή υπάρχουν σχέσεις μεταξύ κόμβων που λείπουν, γίνεται διερεύνηση του σχήματος (schema) της οντολογίας.

Στο ερώτημα ‘state mississippi river’ του παραδείγματός μας, ένα από τα σύνολα F περιέχει τις κλάσεις ‘State’ και ‘River’ συνοδευόμενες από το στιγμιότυπο ‘Mississippi River’. Παρότι δεν εντοπίζονται δυσκολίες στη σύνδεση του στιγμιότυπου ‘Mississippi river’ με την κλάση ‘River’, καθώς ανήκει σε αυτή, δεν υπάρχει σαφής σχέση μεταξύ αυτού και της κλάσης ‘State’. Με τη βοήθεια του σχήματος της οντολογίας (Εικόνα 2.8), διαπιστώνεται ότι είναι δυνατόν να συνδέσουμε ένα στιγμιότυπο που ανήκει στην κλάση ‘River’ με ένα άλλο που να ανήκει στην κλάση ‘State’ μέσω της σχέσης ‘run through’, ο οποία θα αποτελέσει βέλος στον τελικό γράφο ερωτήματος που θα παραχθεί. Καθώς δε γνωρίζουμε ποιο στιγμιότυπο ακριβώς

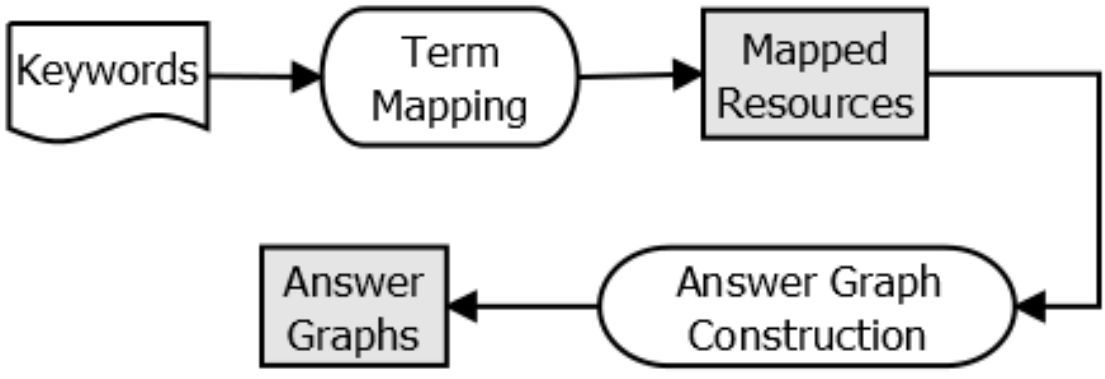
είναι αυτό που μόλις περιγράφηκε, θα αποτελέσει κόμβο μεταβλητής (variable node) στον γράφο.

Αν ένας συνεκτικός γράφος μπορεί, με τη μέθοδο που μόλις παρουσιάστηκε, να δεθεί με όλους τους πόρους του συνόλου F, τον εντάσσουμε αμέσως ως υποψήφιο τυπικό ερώτημα· αλλιώς, κρατάμε τη μεγαλύτερη συνεκτική συνιστώσα του και παράγουμε τον γράφο ερωτήματος ως εξής: 1) Στιγμιότυπα που ανήκουν σε κλάσεις που περιέχονται στο σύνολο F ή κλάσεις που εντοπίστηκαν κατά τη διάσχιση του γράφου αποτελούν κόμβους μεταβλητών (variable nodes) 2) Στιγμιότυπα ή λεκτικά που διατίθενται αποτελούν τερματικούς κόμβους. 3) Πόροι που αντιστοιχούν σε ιδιότητες αποτελούν βέλη του γράφου. Δεδομένου ότι η SPARQL έχει γραφοειδές μοτίβο, είναι εύκολο να παράγουμε, βάσει του γράφου που κατασκευάστηκε, το αντίστοιχο SPARQL ερώτημα.

3.2. Η τροποποιημένη προσέγγιση του SPARK

Ο SPARK επιλέχθηκε προς υλοποίηση διότι η απλότητα στη δομή τον καθιστά κατάλληλο για την κατανόηση της διαδικασίας αναζήτησης με λέξεις-κλειδιά σε σημασιολογικά δεδομένα ενώ παράλληλα επιτρέπει πληθώρα μετατροπών και βελτιώσεων. Στόχος υπήρξε η βελτίωση του χρόνου εκτέλεσης (time complexity), η αποδοτική εκμετάλλευση της διαθέσιμης μνήμης (space complexity) και η χρήση των αποτελεσμάτων στη διεξαγωγή συμπερασμάτων σχετικά με την εξατομίκευση (personalization) και τη διαφοροποίηση (diversification) τους.

Η μονάδα επεξεργασίας οντολογίας αποκόπηκε από το κύριο μέρος του αλγορίθμου αποτελώντας, πλέον, τμήμα του σταδίου προεπεξεργασίας των δεδομένων. Η μονάδα κατασκευής τυπικών ερωτημάτων (formal query construction module), της οποίας ο αρχικός της ρόλος ήταν να παρέχει ως έξοδο ένα σύνολο SPARQL ερωτημάτων, ανάχθηκε σε μονάδα κατασκευής γράφων απαντήσεων (answer graph construction module) η οποία δέχεται ως είσοδο ένα σύνολο από λέξεις-κλειδιά και δίνει ως έξοδο γράφους οι οποίοι συνδέουν τους κόμβους που αντιστοιχούν στα στοιχεία του συνόλου της εισόδου. (Εικόνα 3.6)



Εικόνα 3.6

Μόλις ο χρήστης εισάγει ένα ερώτημα αποτελούμενο από λέξεις-κλειδιά, το στάδιο της αντιστοίχησης όρων (term mapping) εκμεταλλεύεται την ύπαρξη του συγκεντρωτικού Hash Table που κατασκευάσαμε κατά την προεπεξεργασία των δεδομένων προκειμένου να εντοπίσει τους κόμβους του γράφου της οντολογίας που συνδέονται με τους όρους του ερωτήματος. Στη συνέχεια, το στάδιο κατασκευής γράφων ερωτημάτων (query graph construction) απαριθμεί όλους τους πιθανούς συνδυασμούς από κόμβους, όπως αυτοί προέκυψαν στο προηγούμενο στάδιο. Για κάθε συνδυασμό επιλέγει έναν κόμβο ως αφετηρία και, εφαρμόζοντας αλγόριθμο αναζήτησης κατά πλάτος, προσπαθεί να κατασκευάσει ένα συνεκτικό γράφο που θα συνδέει την αφετηρία με τους υπόλοιπους κόμβους του συνδυασμού.

3.2.1. Μονάδα επεξεργασίας της Οντολογίας **(Ontology Processing Module)**

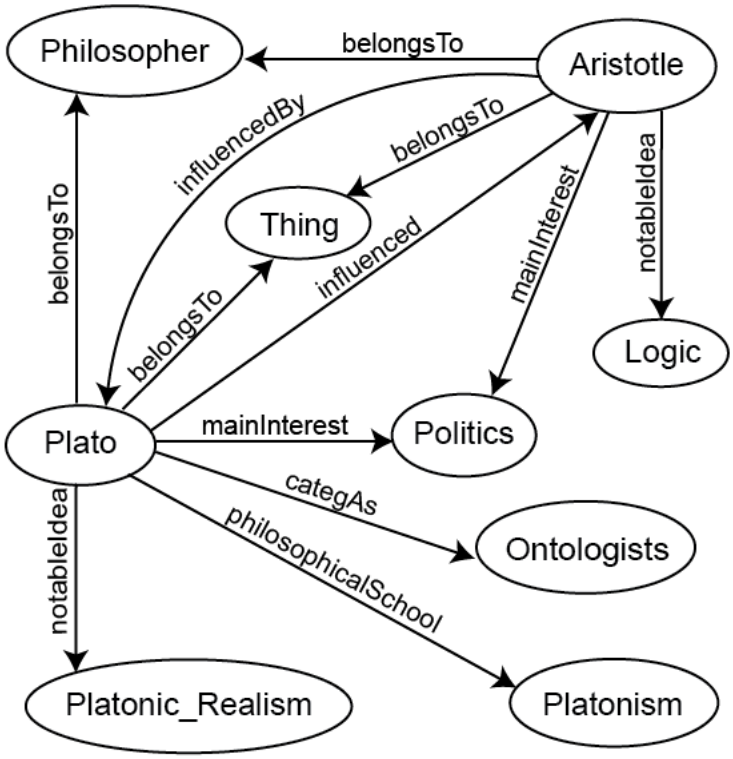
Όλα τα πειράματά που υλοποιήθηκαν στα πλαίσια αυτής της εργασίας έχουν διεξαχθεί στο DBpedia Data Set – σύνολο προτάσεων γραμμένες σε RDF, το οποίο έχει κατασκευαστεί μέσω της εξαγωγής των διαφόρων ειδών δομημένης πληροφορίας από την Wikipedia και το συνδυασμό αυτών σε μια τεράστια ενοποιημένη βάση γνώσης. Αποτελείται από ένα σύνολο πόρων (resources) και ένα σύνολο λεκτικών (literals) και έχει αναπαρασταθεί, στο στάδιο της προεπεξεργασίας των δεδομένων μας, ως ένας κατευθυνόμενος γράφος. Οι κορυφές του γράφου αυτού αντιστοιχήθηκαν σε στιγμιότυπα (instances), κλάσεις (classes) και κατηγορίες (categories) ενώ οι κατευθυνόμενες ακμές του σε ιδιότητες (properties).

Η διαφοροποίηση που παρατηρείται συγκριτικά με την αρχική υλοποίηση της βάσης γνώσης αφορά στην απόφαση να μην ενταχθούν τα λεκτικά με τη μορφή κορυφών στο γράφο της οντολογίας και άμεση συνέπεια αυτού ήταν να παραλειφθούν οι ιδιότητες που τα συνέδεαν με πόρους, οι λεγόμενες DataType Properties. Ο κομβικός ρόλος των λεκτικών που αποτελούσαν ετικέτες (labels) πόρων στην κατασκευή του μηχανισμού αντιστοίχησης πόρων με λέξεις – κλειδιά κατέστησε οποιαδήποτε άλλη χρήση τους πλεονασμό ενώ παράλληλα δε δόθηκε έμφαση σε αυτά που αναπαριστούσαν μια αριθμητική τιμή (π.χ. ο πληθυσμός μιας χώρας).

Η κατασκευή του γράφου της οντολογίας που μελετήθηκε αλλά και η ευρετηριοποίηση (indexing) των πόρων της είναι διαδικασίες κοινές για όλους τους αλγορίθμους που υλοποιήθηκαν και ταυτόχρονα ικανές να αποκοπούν από το κύριο μέρος τους. Λαμβάνοντας υπόψη την διαπίστωση αυτή, προτιμήθηκε η κατάργηση της μονάδας επεξεργασίας της οντολογίας (ontology processing module), όπως αυτή ορίστηκε στην αρχική μορφή του SPARK, και η αντικατάστασή της από μια ενισχυμένη μονάδα προεπεξεργασίας, κοινή για όλους τους αλγορίθμους. Η επιλογή αυτή οδήγησε σε σημαντικές βελτιώσεις στο χρόνο εκτέλεσης των αλγορίθμων, σε αποφυγή επανάληψης κοινών διαδικασιών και άνοιξε το δρόμο για τη δημιουργία περισσότερο οργανωμένου και περιεκτικού κώδικα.

3.2.2. Μονάδα Κατασκευής Γράφων Απαντήσεων (Answer Graph Construction Module)

Στην ανάλυση που ακολουθεί, στην κατεύθυνση της καλύτερης κατανόησης, θεωρήθηκε απαραίτητο να χρησιμοποιηθεί ένα παράδειγμα στο οποίο θα βλέπουμε να εφαρμόζονται οι διαδικασίες που κάθε φορά περιγράφονται. Στην εικόνα 3.7 απεικονίζεται ο γράφος μιας οντολογίας που κατασκευάσαμε χρησιμοποιώντας ένα υποσύνολο προτάσεων του DBpedia Data Set. Οι κορυφές έχουν αναπαρασταθεί με τη βοήθεια συμβολοσειρών που αντιστοιχούν στο τμήμα του URI τους που ακολουθεί μετά την τελευταία εμφάνιση του συμβόλου "/" – πρακτική που έχει υιοθετηθεί για να διευκολυνθεί ο αναγνώστης. Τέλος, το ερώτημα που θα θεωρείται ότι έχει εισαχθεί από το χρήστη είναι το "*platonism logic philosopher"*.



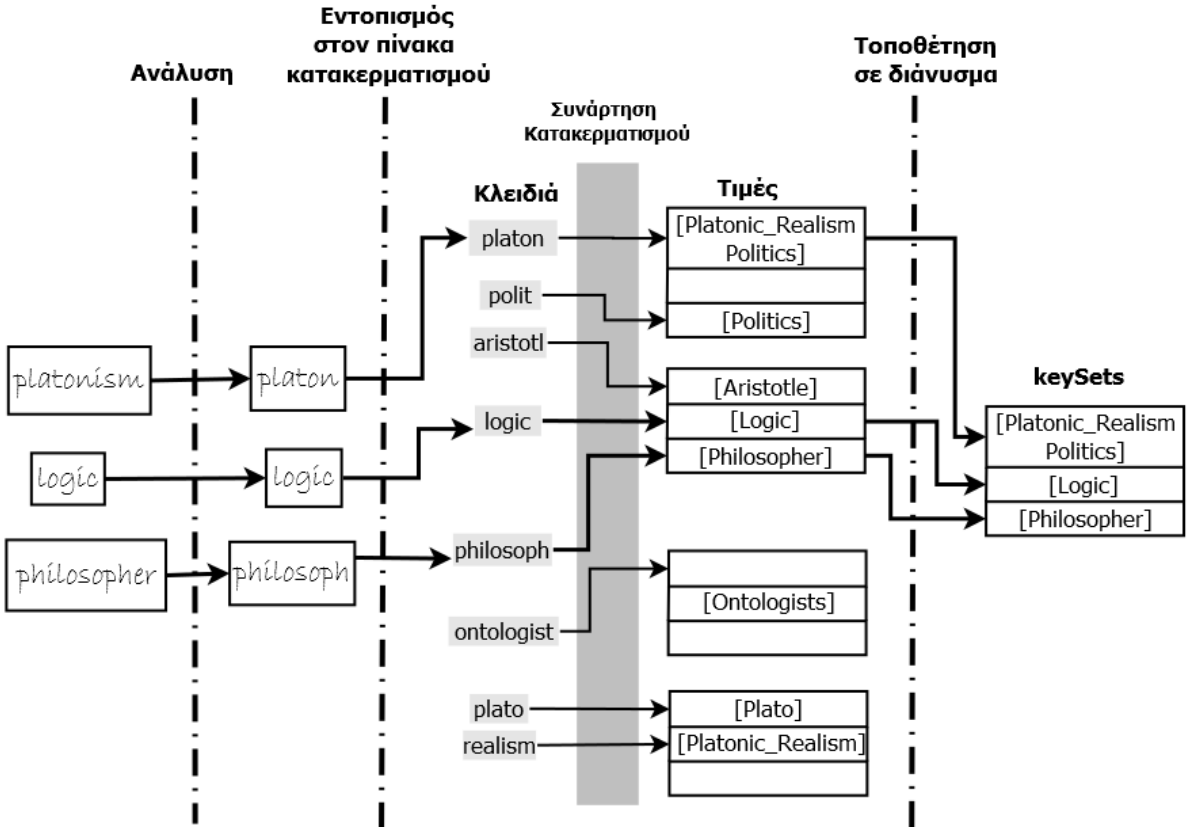
Εικόνα 3.7

A. Αντιστοίχηση Όρων (Term Mapping)

Ο ρόλος της διαδικασίας Αντιστοίχησης Όρων, όπως αυτός ορίζεται στην πρωτότυπη έκδοση του SPARK, είναι να εντοπίσει τους πόρους (resources) της οντολογίας που αντιστοιχούν σε κάθε όρο του ερωτήματος που έθεσε ο χρήστης. Διαπιστώνοντας ότι η εν λόγω διαδικασία επιδέχεται βελτιώσεις στο χρόνο εκτέλεσής της, δημιουργήθηκε ο πίνακας κατακερματισμού kToNset (**k**eyword**To**N**o**des**S**et) στο στάδιο της προεπεξεργασίας.

Υλοποιημένος ως αντικείμενο της κλάσης HashMap, η οποία εξασφαλίζει απόδοση σταθερού χρόνου όχι μόνο για την εισαγωγή αλλά και για την εύρεση ενός αντικειμένου, ο

πίνακας αυτός περιλαμβάνει για κάθε εν δυνάμει λέξη – κλειδί μια καταχώρηση με το σύνολο των αναγνωριστικών των κορυφών του γράφου με τις οποίες συνδέεται. Λαμβάνοντας υπόψη το περιεχόμενο του πίνακα αυτού, διαπιστώνεται ότι η Αντιστοίχηση Όρων απλοποιείται καθώς ανάγεται σε μια διαδικασία εντοπισμού κλειδιών (keys) σε έναν Hash Table και καταγραφής των τιμών (values) που τους αντιστοιχούν. (Εικόνα 3.8)



Εικόνα 3.8

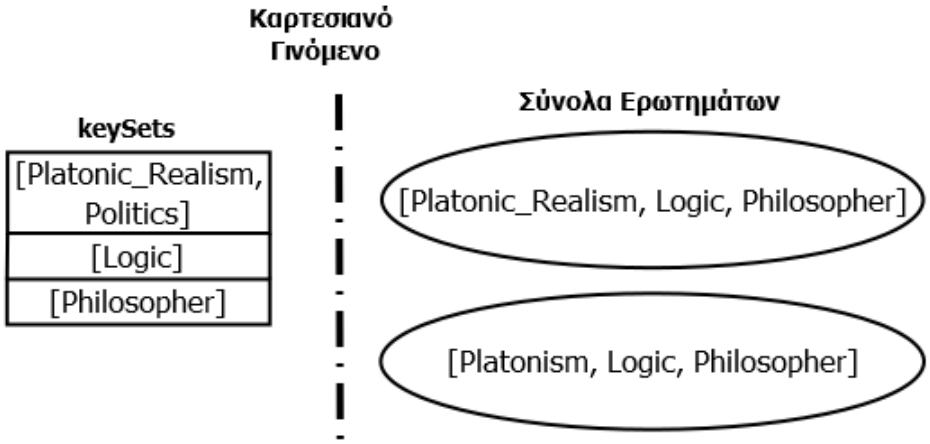
Έστω, λοιπόν, ότι ο χρήστης εισάγει m όρους στο ερώτημά του. Με τη βοήθεια ενός EnglishAnalyzer, διατηρείται μόνο το θέμα καθενός από τους όρους αυτούς, ενώ απορρίπτεται η κατάληξή τους. Η ενέργεια αυτή είναι απαραίτητη προκειμένου να μπορέσει να γίνει ταύτιση με μία από τις πιθανές-λέξεις κλειδιά στις οποίες είχε επίσης εφαρμοστεί ανάλυση πριν ενταχθούν στον πίνακα κατακερματισμού.

Στη συνέχεια, για κάθε όρο, να εντοπίσουμε την καταχώρηση του kToNset στην οποία ο εντοπίζεται ως κλειδί και τοποθετήσουμε το σύνολο που του αντιστοιχεί στο διάνυσμα (array) *keySets* μήκους m. Δεδομένου ότι ο kToNset έχει υλοποιηθεί ως αντικείμενο της κλάσης Hash Map η οποία έχει κατασκευαστεί έτσι ώστε η μέθοδος εντοπισμού κλειδιού (get) να απαιτεί χρόνο O(1) και λαμβάνοντας υπόψη ότι η τοποθέτηση στοιχείου σε πίνακα υλοποιείται σε σταθερό χρόνο, συμπεραίνουμε ότι η διαδικασία Αντιστοίχησης Όρων στοιχίζει 2*m*O(1). Ο

αριθμός των όρων που εισάγει ο χρήστης είναι σημαντικά μικρότερος από τον αριθμό των κορυφών του γράφου της οντολογίας και, συνεπώς, ο χρόνος 2*m*O(1) θεωρείται σταθερός.

Β. Κατασκευή Γράφων Ερωτημάτων (Query Graph Construction)

Σε αυτό το σημείο, για κάθε όρο του ερωτήματος του χρήστη, είναι διαθέσιμο ένα σύνολο το οποίο περιέχει τους κόμβους του γράφου με τους οποίους ο όρος έχει αντιστοιχηθεί, μέσω της διαδικασίας ανάκτησης πληροφορίας. Υπολογίζοντας το καρτεσιανό γινόμενο των συνόλων αυτών, των στοιχείων δηλαδή του διανύσματος keysets, παράγονται όλα τα δυνατά σύνολα ερωτημάτων (query sets), τα οποία ορίσαμε στην παράγραφο 3.1, και εξασφαλίζεται ότι κάθε στοιχείο τους αντιστοιχεί σε διαφορετική λέξη-κλειδί. (Εικόνα 3.9)

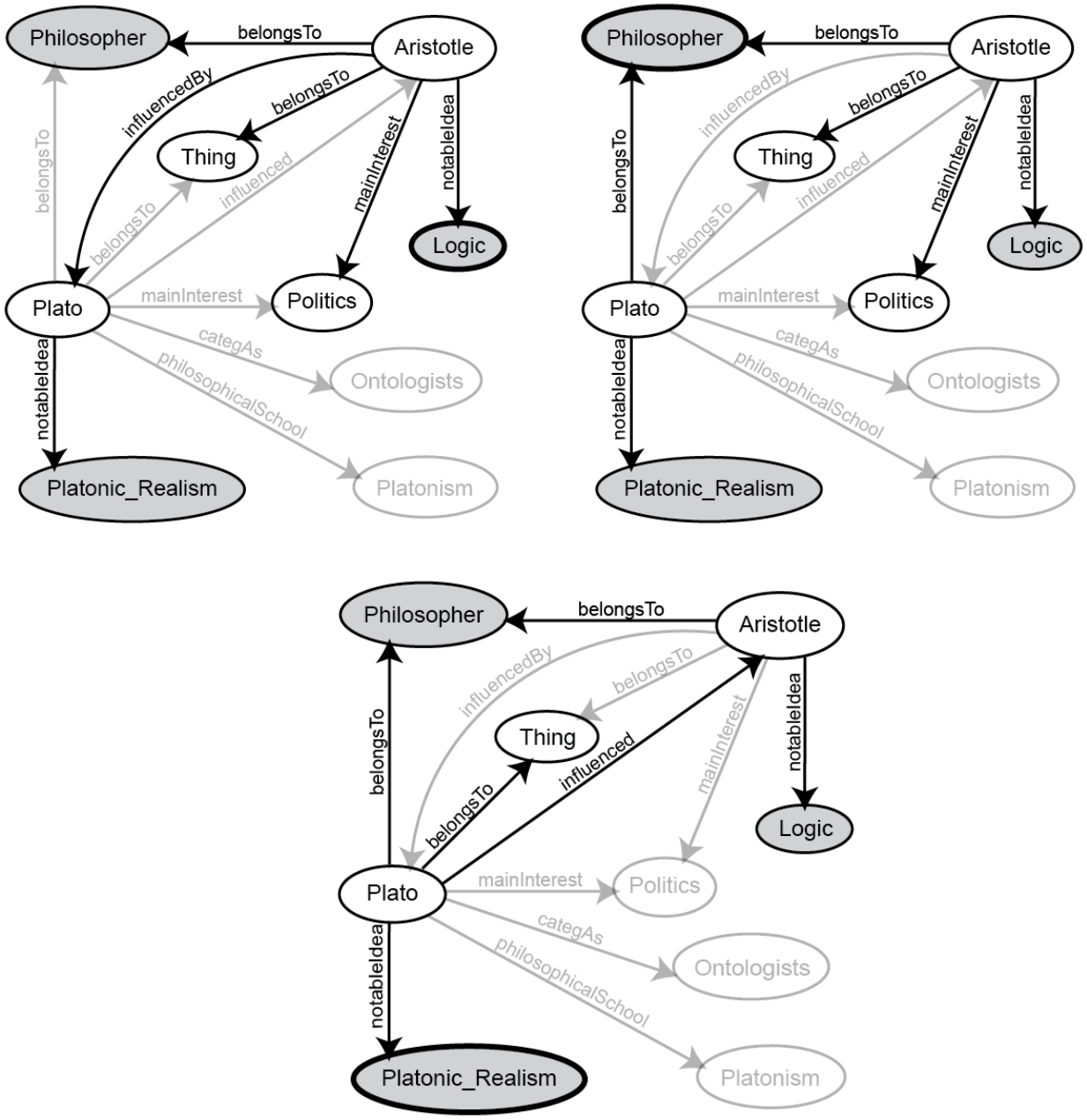


Εικόνα 3.9

Σύμφωνα με την αρχική μορφή του SPARK, εφαρμόζεται σε κάθε σύνολο ερωτήματος ο αλγόριθμος του Kruskal για την εύρεση του Ελάχιστου Συνδετικού Δέντρου που θα συνδέει τα μέλη του. Στην τροποποιημένη μορφή του αλγορίθμου, επιλέχθηκε να δοθεί λιγότερη έμφαση στην έννοια του ελαχίστου και να αποτελέσει κύριο μέλημα ο εντοπισμός όσο περισσότερων λύσεων που θα μπορούσαν να μελετηθούν. Ο γράφος της οντολογίας μετατρέπεται σε μη κατευθυνόμενο και η σύνδεση των στοιχείων-μελών ενός συνόλου ερωτήματος γίνεται με τη βοήθεια του αλγορίθμου αναζήτησης κατά πλάτος (**B**readth **F**irst **S**earch). Ορίστηκε ένα μέγιστο βάθος, maxDepth, στο οποίο ο αλγόριθμος BFS θα μπορούσε να φτάσει καθώς θεωρήθηκε ότι πιθανόν να υπάρχουν περιπτώσεις όπου η υπέρβαση του βάθους αυτού να οδηγεί σε λύσεις μηδαμινής σημασίας.

Συγκεκριμένα, κάθε κόμβος ενός συνόλου ερωτήματος αποτέλεσε αφετηρία ενός περιορισμένου BFS αλγορίθμου ο οποίος είχε σκοπό να παράγει ένα συνεκτικό γράφο που θα περιείχε τους υπόλοιπους κόμβους του συνόλου. Ο λόγος για τον οποίο ελέγχονται πολλαπλές αφετηρίες για το ίδιο σύνολο αφορούσε την προσπάθειά παραγωγής περισσότερων αποτελεσμάτων. Σε περίπτωση που για μία από τις αφετηρίες αυτές η κατά πλάτος αναζήτηση έφτανε σε βάθος μεγαλύτερο από maxDepth χωρίς να επιτύχει το στόχο που έχει τεθεί, δεν

γινόταν παραγωγή γράφου. Παρατίθεται ως παράδειγμα τα δέντρα που παράγονται από το σύνολο ερωτήματος {Platonic_Realism, Logic, Philosopher}. (Εικόνα 3.10)



Εικόνα 3.10: Ο κόμβος με το έντονο περίγραμμα αντιστοιχεί στην αφετηρία του BFS ενώ με γκρι χρώμα σημειώθηκαν οι κόμβοι του συνόλου ερωτήματος. Οι ακμές του αρχικού γράφου που δεν συμπεριλαμβάνονται στον εκάστοτε γράφο ερωτήματος σημειώνονται με μεγαλύτερη διαφάνεια για να γίνεται αισθητή η διαφορά.

Κεφαλαιο 4
SLINKS: Καταταγμένα ερωτήματα με λέξεις-κλειδιά σε γράφους

4.1. Η προσέγγιση του SLINKS

Πολλοί αλγόριθμοι που έχουν αναπτυχθεί στην κατεύθυνση της προσαρμογής των ερωτημάτων με λέξεις-κλειδιά κατά την αναζήτηση σε δεδομένα με γραφοειδή δομή (graph-structured data) εφαρμόζουν ευριστικούς (heuristic) αλγορίθμους για τη διάσχιση του γράφου με αποτέλεσμα να στερούνται τη δυνατότητα εγγύησης υψηλής απόδοσης καθώς υπάρχουν περιπτώσεις γράφων όπου οι ευρυστικές που έχουν επιλεγεί δεν αποδίδουν. Σύνηθες χαρακτηριστικό αποτελούν οι απαιτήσεις μνήμης ενώ δε γίνεται πλήρης εκμετάλλευση των δυνατοτήτων των ευρετηρίων χρησιμοποιώντας τα μόνο για τον εντοπισμό των κόμβων που περιέχουν κάθε λέξη κλειδί και ανάγοντας την εύρεση των υπογράφων που συνδέουν τους κόμβους αυτούς σε πρόβλημα διάσχισης του γράφου.

Ο BLINKS (**B**i-**L**evel **IN**dexing for **K**eyword **S**earch) αναπτύχθηκε με σκοπό να προτείνει τρόπους να προσπεραστούν οι παραπάνω δυσκολίες. Αποτελεί ένα σύστημα ευρετηριοποίησης και επεξεργασίας ερωτημάτων σε κατανεμημένες αναζητήσεις με λέξεις-κλειδιά εφαρμοσμένες σε κατευθυνόμενους γράφους με κόμβους στους οποίους έχουν αντιστοιχηθεί ετικέτες. Οι κύριες συνεισφορές του είναι οι εξής:

Καλύτερη Στρατηγική Αναζήτησης
Χρησιμοποιείται περιορισμένη ως προς το κόστος επέκταση κάθε κόμβου (cost-balanced expansion) δίνοντας μια νέα διάσταση στην στρατηγική της αναζήτησης προς τα πίσω (backward search strategy) η οποία αναλαμβάνει την εξερεύνηση του γράφου ξεκινώντας από κόμβους που περιέχουν τις λέξεις-κλειδιά του ερωτήματος.

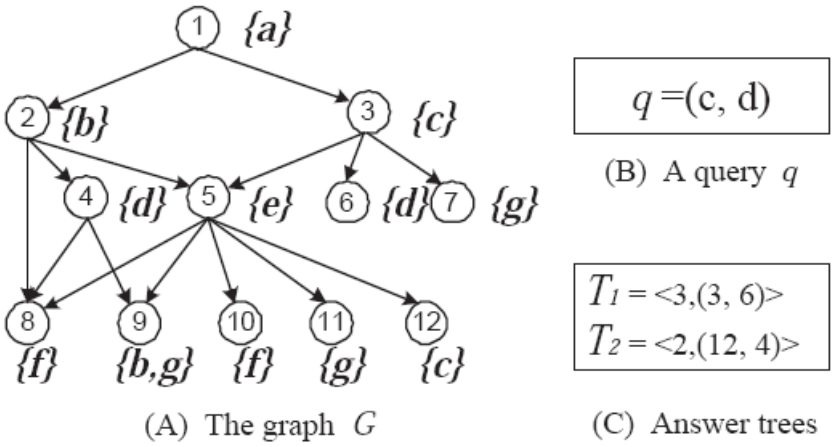
Συνδυασμός ευρετηριοποίησης με αναζήτηση
Η αναζήτηση επιταχύνεται με τη βοήθεια ευρετηρίου το οποίο επιλεκτικά υπολογίζει και οργανώνει εκ των προτέρων πληροφορίες που αφορούν συντομότερα μονοπάτια. Με τον τρόπο αυτό μειώνεται το απαιτούμενο κόστος κατά την εκτέλεση της αναζήτησης προς τα πίσω, η οποία βελτιστοποιείται. Παράλληλα υλοποιείται η αναζήτηση προς τα εμπρός (forward search) η οποία επιταχύνεται επίσης καθώς γίνονται μεγαλύτερα κατευθυνόμενα βήματα.

Ευρετηριοποίηση βασισμένη στη διαίρεση
Ο γράφος στον οποίο πραγματοποιούνται οι αναζητήσεις διαιρείται σε αρκετούς υπογράφους (subgraphs) οι οποίοι αποτελούν τις λεγόμενες συστάδες (blocks). Στο ευρετήριο δύο επιπέδων (bi-level index) αποθηκεύονται αρχικά μόνο απαραίτητες πληροφορίες σε επίπεδο συστάδας προκειμένου να παραχθούν οι απαιτούμενες δομές και να καθοδηγηθεί η μετάβαση μεταξύ των συστάδων κατά την αναζήτηση. Ακόμα, αποθηκεύονται λεπτομερέστερα στοιχεία με σκοπό να επιταχυνθεί η αναζήτηση εντός μίας συστάδας. Ο συμβιβασμός ως προς τον χώρο που απαιτείται για την αποθήκευση των δομών που χρειάζονται επιφέρει οφέλη ως προς την απόδοση της αναζήτησης και τελικά ως προς το χώρο εκτέλεσης.

Στο σημείο αυτό, πρέπει να διευκρινιστεί ότι το ευρετήριο δύο επιπέδων αναμφισβήτητα αποτελεί καινοτομία και ενισχύει την απόδοση του αλγορίθμου. Ωστόσο, η υλοποίηση του διαμερισμού του γράφου, πολύπλοκων δομών για τη διευκόλυνση της αναζήτησης και συνάρτησης κατάταξης των λύσεων ξέφευγε από τα πλαίσια της παρούσας εργασίας καθώς δόθηκε περισσότερη έμφαση στον εντοπισμό των λύσεων και όχι στον χρόνο εκτέλεσης των αλγορίθμων που χρησιμοποιούνταν, παρότι διατηρήθηκε πάντα ένα μέτρο. Επιλέχθηκε, λοιπόν, να υλοποιηθεί και να τροποποιηθεί ο SLINKS (**S**ingle-**L**evel **IN**dexing for **K**eyword **S**earch) ο οποίος αποτελεί την πρώιμη μορφή και στην ανάλυση αυτού θα κινηθεί παρόν κεφάλαιο.

4.2. Αναζητώντας λύσεις με τη βοήθεια ευρετηρίου ενός επιπέδου

Στα πλαίσια της αναζήτησης με τη βοήθεια ευρετηρίου ενός επιπέδου (single-level index) ένας κατευθυνόμενος γράφος G(V,E) αποτελεί τη βάση στην οποία τίθενται τα ερωτήματα. Το σύνολο E περιλαμβάνει τα τόξα (arcs) του γράφου ενός το σύνολο V περιλαμβάνει τις κορυφές του καθεμία από τις οποίες χαρακτηρίζεται από έναν ακέραιο αριθμό που αποτελεί το αναγνωριστικό τις. Επιπλέον, για κάθε κορυφή του γράφου καταγράφεται ένα σύνολο από συμβολοσειρές που αντιστοιχούν στις εν δυνάμει λέξεις-κλειδιά που περιέχει. Για παράδειγμα, στην Εικόνα 4.1, ο κόμβος εννέα (9) συνδέεται με τις λέξεις-κλειδιά {b,g}.



Εικόνα 4.1

Δοθέντος ενός ερωτήματος με λέξεις-κλειδιά $q = (w_1, w_2, \ldots, w_m)$ και ενός κατευθυνόμενου γράφου G, μια απάντηση στο q είναι το ζεύγος $<r,(n_1, n_2, \ldots, n_m)>$ όπου οι r και n_i είναι κόμβοι (όχι απαραίτητα διαφορετικοί μεταξύ τους) που ικανοποιούν τις παρακάτω ιδιότητες:

- ✓ Για κάθε i, ο κόμβος n_i περιέχει την λέξη-κλειδί n_i.
- ✓ Για κάθε i, υπάρχει ένα κατευθυνόμενο μονοπάτι από τον r στον n_i.

Ο κόμβος r χαρακτηρίζεται ρίζα (root) της απάντησης και οι n_i αποτελούν τους κόμβους στους οποίους προβάλλονται οι όροι του ερωτήματος. Η δεύτερη ιδιότητα, που καλύπτει τη

συνεκτικότητα, απαιτεί μία απάντηση να είναι υπόδεντρο του αρχικού γράφου και η ρίζα του να συνδέεται μέσω μονοπατιού με όλους τους κόμβους.

Σκοπός του υπό μελέτη αλγορίθμου, στην αρχική μορφή του ήταν να εξάγει τις k απαντήσεις που θεωρείται ότι ταιριάζουν καλύτερα με τις απαιτήσεις του χρήστη. Παρότι δε θα γίνει αναφορά στην κατάταξη των απαντήσεων, αξίζει να αναφερθεί ότι η παρατήρηση αυτή έχει ως αποτέλεσμα όλα τα υπόδεντρα που θα ληφθούν ως απαντήσεις να έχουν διαφορετικούς κόμβους ως ρίζες τους. Η απαίτηση αυτή εξασφαλίζει την αποφυγή της περίπτωσης όπου ένας κεντρικός κόμβος από τον οποίο εξέρχονται πολλές ακμές λαμβάνεται ως ρίζα σε πολλά δέντρα τα οποία όμως διαφοροποιούνται στο ελάχιστο μεταξύ τους. Ως χαρακτηριστικό παράδειγμα αναφέρεται η περίπτωση όπου ένας συγγραφέας έχει γράψει k_1 δημοσιεύσεις με τίτλο που να περιέχει τη λέξει "privacy", k_2 με τίτλο που να περιέχει τη λέξη "mining" και k_3 δημοσιεύσεις με τη λέξη "sensor" στον τίτλο. Ο συγγραφέας αυτός θα αποτελέσει ρίζα σε $k_1 \times k_2 \times k_3$ απαντήσεις και αν το γινόμενο αυτό προσεγγίζει το k τα δέντρα που θα επιστραφούν στο χρήστη δε θα έχουν ιδιαίτερο ενδιαφέρον.

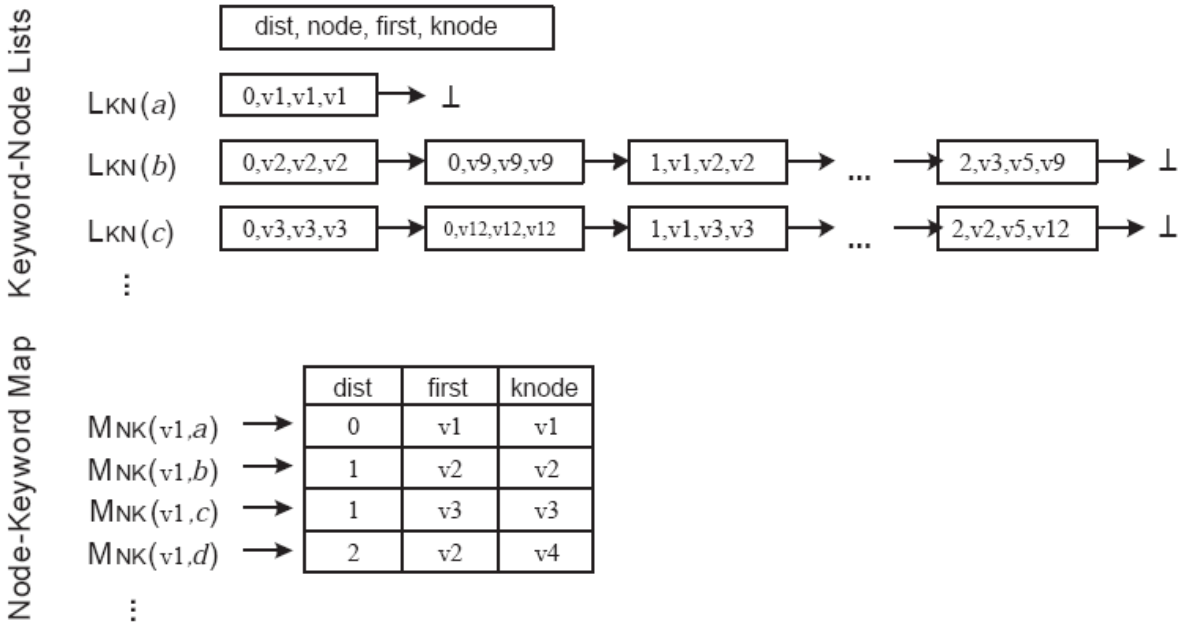
4.2.1. Το ευρετήριο ενός επιπέδου (Single-Level Index)

Το σύνολο O_i περιλαμβάνει όλους τους κόμβους του γράφου που περιέχουν τη λέξη-κλειδί k_i. Ως σύμπλεγμα (cluster) Ei χαρακτηρίζουμε το σύνολο κόμβων που συνδέονται μέσω ενός μονοπατιού με έναν από τους κόμβους που περιέχουν την λέξη-κλειδί k_i χωρίς να είναι απαραίτητο να υπάγονται στο O_i. Στην προσπάθεια να συγκεντρωθούν πληροφορίες που αφορούν όλα τα εν δυνάμει E_i προέκυψε η ανάγκη για ύπαρξη εύκολα διαχειρίσιμων δομών η κατασκευή των οποίων σκοπό θα είχε να βελτιώσει την απόδοση του αλγορίθμου (online performance). Η κοινή προσέγγιση που ακολουθείται αφορά στη δημιουργία ενός σταδίου προεπεξεργασίας όπου θα πραγματοποιείται ένα σύνολο απαραίτητων υπολογισμών (offline computation).

Εικόνα 4.2

Αρχικά, υπολογίζεται εκ των προτέρων για κάθε λέξι-κλειδί, η ελάχιστη απόσταση από κάθε κόμβου του γράφου προς αυτήν, ή με μεγαλύτερη ακρίβεια, προς κάθε κόμβο που την περιέχει. Η $L_{KN}(w)$ (keyword – node list) αποτελεί τη λίστα με κόμβους που μπορούν να φτάσουν στη λέξη-κλειδί w. Κάθε στοιχείο (καταχώρηση) της λίστας αντιστοιχεί σε μία δομή με τέσσερα (4) πεδία: *dist, node, first, knode* (Εικόνα 4.2). Το πεδίο *dist* είναι η ελάχιστη απόσταση μεταξύ του *node* και του *knode*, ο οποίος περιέχει την w, και μπορεί να ισούται με ∞ σε περίπτωση που δεν υπάρχει μονοπάτι μεταξύ των κόμβων. Ο *first* αντιστοιχεί στον πρώτο κόμβο του συντομότερου αυτού μονοπατιού που έχει καταγραφεί και μπορεί να βοηθήσει στην αναδόμηση του μονοπατιού.

Στην εικόνα που ακολουθεί έχει σχεδιαστεί ένα τμήμα της λίστας κόμβων-κλειδιών που κατασκευάστηκε για τον γράφο της εικόνας 4.1 υπό την υπόθεση ότι όλα τα τόξα του έχουν μήκος ένα (1). Ως παράδειγμα, στην λίστα της λέξης-κλειδί *b* βρίσκεται η καταχώρηση $(0,u_2,u_2,u_2)$ που υποδεικνύει ότι ο κόμβος u_2 (= node) φτάνει τον κόμβο u_2 (= knode) που περιέχει την b μέσω ενός μονοπατιού μήκους 0 γεγονός που ερμηνεύει το λόγο για τον οποίο ο πρώτος κόμβος του μονοπατιού είναι ο u_2 (= first). Ανάλογα, η καταχώρηση $(2,u_3,u_5,u_9)$ υποδεικνύει ότι το συντομότερο μονοπάτι από τον κόμβο u_3 στον u_5 που περιέχει την b είναι το $u_3 \rightarrow u_5 \rightarrow u_9$, μήκους 2.



Εικόνα 4.3

Με τη βοήθεια των λιστών Lkn, κατά την εκτέλεση του αλγορίθμου, υπάρχει η δυνατότητα μετάβασης σε σταθερό χρόνο από μία λέξη-κλειδί σε οποιοδήποτε κόμβο του γράφου. Ωστόσο, όπως θα αναλυθεί αργότερα, κρίθηκε εξίσου αναγκαία η ύπαρξη ενός μηχανισμού που θα επιτρέπει τον εντοπισμό όλων των λέξεων-κλειδιά στις οποίες μπορεί να φτάσει ένας κόμβος ανεξάρτητα με την απόστασή του από αυτές.

Στην κατεύθυνση αυτή, υπολογίζεται εκ των προτέρων, για κάθε κόμβο u, το μήκος του συντομότερου μονοπατιού μεταξύ αυτού και κάθε λέξης-κλειδί και οι πληροφορίες που συγκεντρώνονται τοποθετούνται στον πίνακα κατακερματισμού M_{NK} (node-keyword map). Δοθέντος ενός κόμβου u και μιας λέξης-κλειδί w, η καταχώρηση $M_{NK}(u,w)$ περιέχει μία δομή (*dist, first, knode*) τα πεδία της οποίας διατηρούν σημασία ίδια με εκείνη που έχουν στις λίστες L_{KN}. Όπως είναι προφανές, η πληροφορία που περιέχεται στην καταχώρηση $M_{NK}(u,w)$ μπορεί να διεξαχθεί από τη λίστα $L_{KN}(w)$ στην οποία μπορούμε σε γραμμικό χρόνο να εντοπίσουμε την

ελάχιστη απόσταση μεταξύ του κόμβου u και της λέξης-κλειδί w. Ο λόγος ύπαρξης του πίνακα κατακερματισμού είναι ότι η αναζήτηση αυτή αποφεύγεται και το επιθυμητό αποτέλεσμα είναι διαθέσιμο σε σταθερό χρόνο O(1).

Το σύνολο των λιστών Lkn σε συνδυασμό με τον πίνακα κατακερματισμού Mnk αποτελούν το ευρετήριο (index) ενός επιπέδου (single-level) καθώς ορίζεται στα πλαίσια όλου του υπό μελέτη γράφου και όχι υπογράφου του. Τόσο οι λίστες όσο και ο πίνακας περιέχουν N*K καταχωρήσεις όπου N είναι ο αριθμός των εν δυνάμει λέξεων-κλειδιών και K ο αριθμός των κόμβων του γράφου. Σε πολλές εφαρμογές ο K είναι της ίδιας τάξης με τον N οδηγώντας σε χωρική πολυπλοκότητα (space complexity) $O(N^2)$, η οποία είναι προφανώς αδύνατη για μεγάλους γράφους.

Το ευρετήριο ενός επιπέδου μπορεί να συμπληρωθεί με αναζητήσεις προς τα πίσω οι οποίες θα έχουν ως αφετηρίες τους κόμβους που περιέχουν λέξεις-κλειδιά. Προκειμένου να υπολογιστούν οι αποστάσεις μεταξύ κόμβων και λέξεων-κλειδιών τρέχουν παράλληλα N αντίγραφα του αλγόριθμου Dijkstra για την εύρεση συντομότερου μονοπατιού από μία κορυφή με αφετηρία κάθε φορά έναν από τους N κόμβους του γράφου. Η διαδικασία αυτή έχει πολυπλοκότητα $O(N^2)$.

4.2.2. Αλγόριθμος αναζήτησης με το ευρετήριο ενός επιπέδου

Στο σημείο αυτό περιγράφεται ο τρόπος με τον οποίο συνδυάζονται όλες οι δομές που έχουν κατασκευαστεί με σκοπό τον εντοπισμό των απαντήσεων εκείνων που θα ενδιέφεραν το χρήστη. Η περιγραφή συνοδεύεται από τον αντίστοιχο ψευδοκώδικα (Εικόνα 4.4) σε γραμμές του οποίου γίνονται παραπομπές με σκοπό την καλύτερη κατανόηση.

Με κατεύθυνση προς τα πίσω (Expanding Backward)

Δοθέντος ενός ερωτήματος με λέξεις-κλειδιά $q = (w_1, w_2, \ldots, w_m)$ παράγουμε έναν κέρσορα για καθεμία από τις m λίστες και τον χρησιμοποιούμε για τη διάσχισή τους. Με κλήση της μεθόδου next (Γραμμή 6) ο κέρσορας c_i επιστρέφει το επόμενο στοιχείο της λίστας $L_{KN}(w_i)$ παρέχοντας πρόσβαση στον κόμβο που του αντιστοιχεί αλλά και στην απόστασή του από τη λέξη-κλειδί w_i. Από κατασκευής, η σειρά με την οποία έχουν τοποθετηθεί οι κόμβοι στη λίστα ακολουθεί την αρχή σύμφωνα με την οποία όσο προχωράμε στη λίστα συναντάμε κόμβους που βρίσκονται σε μεγαλύτερη απόσταση από την αφετηρία (equi-distance expansion). Προκειμένου να πραγματοποιηθεί μετάβαση μεταξύ των συμπλεγμάτων, η επιλογή του κέρσορα που μας υποδεικνύει τη λίστα από την οποία θα αντλήσουμε τον επόμενο κόμβο γίνεται “in a round-robin manner” (Γραμμή 5), δηλαδή ακολουθεί κυκλική σειρά (round-robin) η οποία μας εξασφαλίζει ότι το σύμπλεγμα που κάθε φορά μελετάμε έχει την ελάχιστη πληθικότητα (cost-balanced expansion).

Με κατεύθυνση προς τα εμπρός (Expanding Forward)

Μόλις γίνει επίσκεψη σε έναν κόμβο, ο πίνακας κατακερματισμού M_{NK} βοηθά στην εύρεση των αποστάσεών του από άλλους κόμβους (Γραμμή 17). Με βάσει τις πληροφορίες που αντλούμε από τον πίνακα αυτό μπορεί να καθοριστεί εάν έχει εντοπιστεί. Πιο συγκεκριμένα, για κάθε κόμβο που αποτελεί υποψήφια ρίζα διατηρείται μια δομή της μορφής <root, $dist_1$, $dist_2$, …, $dist_m$> όπου ως *root* καταγράφεται ο υπό επίσκεψη κόμβος και ως $dist_i$ η απόστασή του από τη λέξη-κλειδί w_i. Αν κάποιο $dist_i$ είναι ∞ τότε ο κόμβος *root* δε μπορεί να αποτελεί ρίζα μίας απάντησης επειδή υπάρχει μία λέξη-κλειδί στην οποία δεν έχει πρόσβαση. Στην περίπτωση που όλα τα $dist_i$ είναι πεπερασμένα, οπότε και θα πρέπει το sumDist(u) που αντιστοιχεί στη συνδυασμένη απόσταση από τον κόμβο u προς τις λέξεις κλειδιά, δηλαδή $\sum_i^m disti$, να έχει πεπερασμένη τιμή (Γραμμή 18), έχει εντοπιστεί μια απάντηση στο ερώτημα του χρήστη και θα πρέπει να ελεγχθεί αν θα ενταχθεί στις k ποιο πιθανές απαντήσεις.

Algorithm 1: Searching with the single-level index.

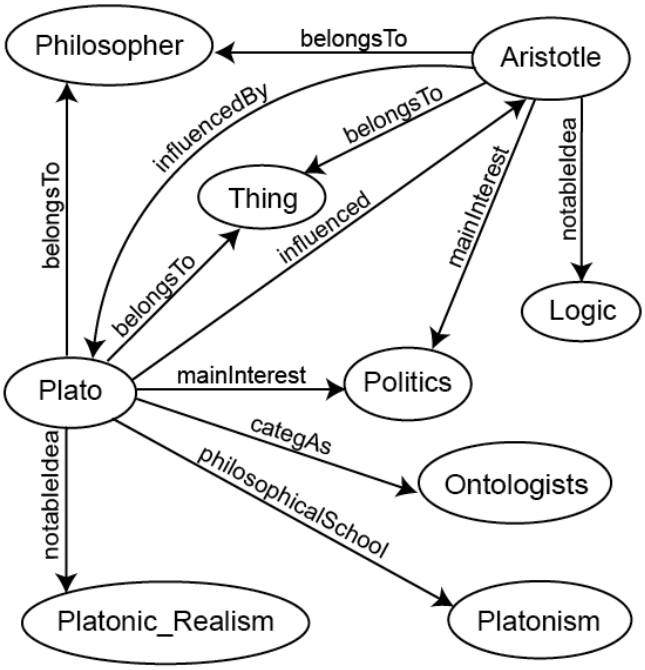
Variables: R: nodes visited; initially ∅.
A: answers found; initially ∅.
τ_{prune}: pruning threshold; initially ∞.

```
1  searchSLINKS(w_1, ..., w_m) begin
2      foreach i ∈ [1, m] do
3          c_i ← new Cursor(L_KN(w_i), 0);
4      while ∃j ∈ [1, m] : c_j.peekDist() ≠ ∞ do
5          i ← pick from [1, m] in a round-robin fashion;
6          ⟨u, d⟩ ← c_i.next();
7          if ⟨u, d⟩ ≠ ⟨⊥, ∞⟩ then visitNode(i, u, d);
8          if |A| ≥ k and Σ_{j=1}^{m} c_i.peekDist() > τ_prune then
9              exit and output the top k answers in A;
10     output up to top k answers in A;
11 end
12 visitNode(i, u, d) begin
13     if R.contains(u) then return;     // already visited
14     R.add(⟨u, ⊥, ..., ⊥⟩);
15     R[u].dist_i ← d;
16     foreach j ∈ [1, i) ∪ (i, m] do      // expand forward
17         R[u].dist_i ← M_NK(u, w_i);
18     if sumDist(u) < ∞ then              // answer found
19         A.add(R[u]);
20         if |A| ≥ k then
           τ_prune ← the k-th largest of {sumDist(v) | v ∈ A}
21 end
```

Εικόνα 4.4

4.3. Ο τροποποιημένος SLINKS

Στην ανάλυση που ακολουθεί, θεωρείται ότι έχει προηγηθεί κατασκευή του γράφου της οντολογίας και εντοπισμός των συμβολοσειρών που αποτελούν εν δυνάμει λέξεις-κλειδιά. Στην κατεύθυνση της καλύτερης κατανόησης, χρησιμοποιείται ένα παράδειγμα στο οποίο εφαρμόζονται οι διαδικασίες που κάθε φορά περιγράφονται. Στην εικόνα 4.5 απεικονίζεται ο γράφος μιας οντολογίας που κατασκευάσαμε χρησιμοποιώντας ένα υποσύνολο προτάσεων του DBpedia Data Set. Οι κορυφές έχουν αναπαρασταθεί με τη βοήθεια συμβολοσειρών που αντιστοιχούν στο τμήμα του URI τους που ακολουθεί μετά την τελευταία εμφάνιση του συμβόλου "/" – πρακτική που έχει υιοθετηθεί για να διευκολυνθεί ο αναγνώστης. Τέλος, το ερώτημα που θα θεωρείται ότι έχει εισαχθεί από το χρήστη είναι το "*platonism logic philosopher"*.



Εικόνα 4.5

4.3.1. Το τροποποιημένο ευρετήριο ενός επιπέδου (Single-Level Index)

Στα πλαίσια αυτής της εργασίας, όπως αναφέρεται και στην ανάλυση των άλλων αλγορίθμων που υλοποιήθηκαν, δεν επιδιώχθηκε να δοθεί έμφαση στην εύρεση των k απαντήσεων που αναμένεται, βάσει κριτηρίων, να είναι οι πλησιέστερες στην προσδοκία του χρήστη. Στόχος είναι η εύρεση όσο το δυνατόν περισσότερων απαντήσεων στις οποίες να μπορούν να εφαρμοστούν μέθοδοι προκειμένου να διεξαχθούν συμπεράσματα όσο αφορά τη διαφοροποίηση (diversification) και την εξατομίκευσή (personalization) τους. Επιλέχθηκε

Η λογική με την οποία έχει λειτουργεί στην αρχική μορφή του ο SLINKS είναι από μόνη της πρωτοποριακή καθώς με τη βοήθεια ενισχυμένων δομών επιτρέπει τον εντοπισμό του μονοπατιού που ενώνει δύο κόμβους του γράφου και αυτός είναι ο λόγος για τον οποίο δεν έγιναν επεμβατικές αλλαγές στον τρόπο με τον οποίο υλοποιούνταν η αναζήτηση. Η άντληση γενικότερων αποτελεσμάτων που επιδιώκει η εργασία αυτή οδήγησε στην ανεξαρτητοποίηση από την έννοια του ελαχίστου η οποία με τη σειρά της οδήγησε στην ανάγκη για επαναπροσδιορισμό των δομών που επρόκειτο να κατασκευαστούν κατά το στάδιο της προεπεξεργασίας, η σημασία της ύπαρξης του οποίου δεν αμφισβητείται.

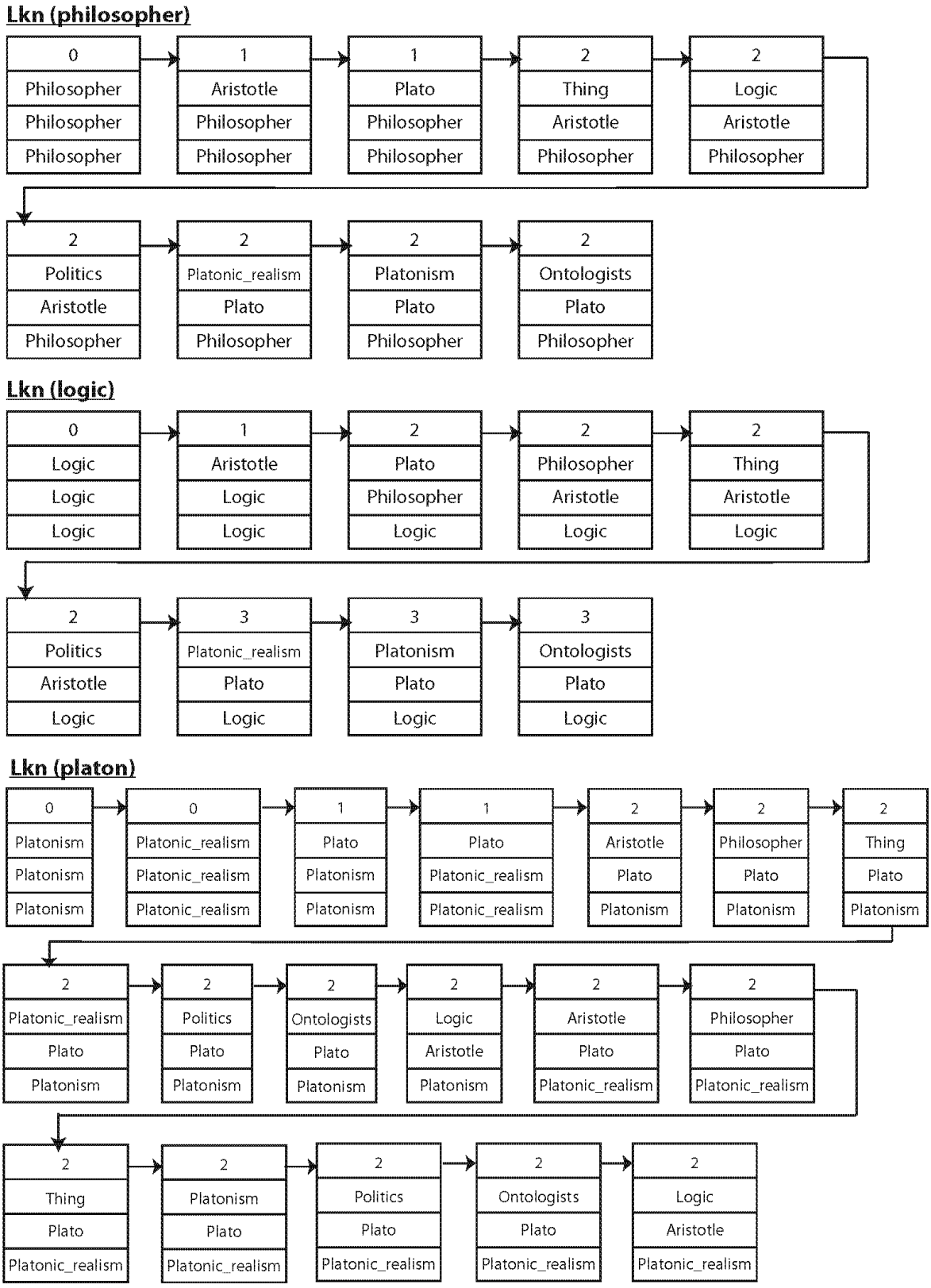
A. Η λίστα $L_{KN}(w)$ (keyword – node list)

Η λίστα $L_{KN}(w)$ στοχεύει στο να καταγράψει όλους τους κόμβους που συνδέονται μέσω μονοπατιού με κόμβο που περιέχει τη λέξη-κλειδί w και κάθε στοιχείο της είναι μία δομή με τέσσερα (4) πεδία: *dist, node, first, knode*. Τα πεδία *node, knode, first* συμπληρώνονται με τα αναγνωριστικά των κόμβων-άκρων (*node, knode*) αλλά και του πρώτου κόμβου (*first*) ενώ στο πεδίο *dist* σημειώνεται το μήκος του εν λόγω μονοπατιού. Στην περίπτωση που η εν λόγω σύνδεση είναι ανέφικτη το πεδίο *dist* λάμβανει την τιμή ∞.

Η διαδικασία αυτή είναι χρονικά ασύμφορη και απαιτεί μεγάλες ποσότητες μνήμης. Το πιο σημαντικό, όμως, μειονέκτημά της είναι ότι, στην προσπάθεια να είναι πλήρης, μεριμνά για την αποθήκευση πληροφοριών που δεν οδηγούν σε ενδιαφέροντα αποτελέσματα. Είναι σαφές ότι εάν επιστραφεί στο χρήστη ένας γράφος όπου οι κόμβοι που θα αντιστοιχούν στους όρους του ερωτήματος του έχουν μεταξύ τους μεγάλη απόσταση το πιο πιθανό είναι να μην αποτελεί ένα αποτέλεσμα που τον ενδιαφέρει. Για παράδειγμα, στο ερώτημα “Java Programmer” η παραγωγή μιας απάντησης που θα καταδεικνύει τον τρόπο με τον οποίο ένας προγραμματιστής συνδέεται με τον καφέ Java[29] δε θα εκτιμηθεί από το χρήστη.

Αξιολογώντας, λοιπόν, τη σημασία των παραγόμενων μονοπατιών, κρίθηκε απαραίτητο ελέγχεται ότι το μήκος των υπαρχόντων μονοπατιών δε θα ξεπερνά το επιλεγμένο ανώτατο όριο. Γίνεται αντιληπτό ότι το ενδιαφέρον δεν εστιάζεται στην εύρεση των συντομότερων μονοπατιών αλλά στον περιορισμό του μήκους τους εντός του επιλεγμένου ορίου το οποίο μπορεί να εκληφθεί ως η ακτίνα ενός κύκλου με κέντρο έναν κόμβο που περιέχει μία εν δυνάμει λέξη-κλειδί. Στην εικόνα 4.6 έχουν καταγραφεί οι λίστες Lkn για τον γράφο της εικόνας Δ με μέγιστο μήκος μονοπατιών ίσο με δύο. Αξίζει να σημειωθεί ότι ο τρόπος με τον οποίο έχουν κατασκευαστεί οι δομές του ευρετηρίου ενός επιπέδου επιτρέπει τον εντοπισμό σχέσεων μεταξύ δύο κόμβων οι οποίοι απέχουν μεταξύ τους κατά το διπλάσιο από τη μέγιστη επιτρεπτή απόσταση.

[29] http://en.wikipedia.org/wiki/Java_coffee


Lkn (philosopher)
0 Philosopher Philosopher Philosopher
1 Aristotle Philosopher Philosopher
1 Plato Philosopher Philosopher
2 Thing Aristotle Philosopher
2 Logic Aristotle Philosopher
2 Politics Aristotle Philosopher
2 Platonic_realism Plato Philosopher
2 Platonism Plato Philosopher
2 Ontologists Plato Philosopher
Lkn (logic)
0 Logic Logic Logic
1 Aristotle Logic Logic
2 Plato Philosopher Logic
2 Philosopher Aristotle Logic
2 Thing Aristotle Logic
2 Politics Aristotle Logic
3 Platonic_realism Plato Logic
3 Platonism Plato Logic
3 Ontologists Plato Logic
Lkn (platon)
0 Platonism Platonism Platonism
0 Platonic_realism Platonic_realism Platonic_realism
1 Plato Platonism Platonism
1 Plato Platonic_realism Platonic_realism
2 Aristotle Plato Platonism
2 Philosopher Plato Platonism
2 Thing Plato Platonism
2 Platonic_realism Plato Platonism
2 Politics Plato Platonism
2 Ontologists Plato Platonism
2 Logic Aristotle Platonism
2 Aristotle Plato Platonic_realism
2 Philosopher Plato Platonic_realism
2 Thing Plato Platonic_realism
2 Platonism Plato Platonic_realism
2 Politics Plato Platonic_realism
2 Ontologists Plato Platonic_realism
2 Logic Aristotle Platonic_realism

Εικόνα 4.6

Β. Ο πίνακας κατακερματισμού M_{NK} (node-keyword map)

Με τη βοήθεια των λιστών L_{KN}, κατά την εκτέλεση του αλγορίθμου, υπάρχει η δυνατότητα μετάβασης σε σταθερό χρόνο από έναν κόμβο που αναμένεται να εντάσσεται σε ένα γράφο-απάντηση, καθώς περιέχει έναν όρο του ερωτήματος, σε οποιοδήποτε κόμβο του γράφου της οντολογίας. Ο πίνακας κατακερματισμού M_{NK} αποτελεί το μηχανισμό που επιτρέπει την υλοποίηση της αντίστροφης διαδικασίας. Επιτρέπει, δηλαδή, τον εντοπισμό όλων των λέξεων-κλειδιά στις οποίες μπορεί να φτάσει ένας κόμβος ανεξάρτητα με την απόστασή του από αυτές. Έτσι, δοθέντος ενός κόμβου u και μιας λέξης-κλειδί w, η καταχώρηση $M_{NK}(u,w)$ περιέχει μία δομή (*dist, first, knode*) τα πεδία της οποίας διατηρούν σημασία ίδια με εκείνη που έχουν στις λίστες L_{KN}.

Σύμφωνα με την αρχική υλοποίηση του SLINKS, υπολογίζεται εκ των προτέρων, για κάθε κόμβο u, το μήκος του συντομότερου μονοπατιού μεταξύ αυτού και κάθε λέξης-κλειδί και οι πληροφορίες που συγκεντρώνονται τοποθετούνται στον πίνακα κατακερματισμού M_{NK} (node-keyword map). Υπενθυμίζεται ότι η πληροφορία που περιέχεται στην καταχώρηση $M_{NK}(u,w)$ μπορεί να διεξαχθεί από τη λίστα $L_{KN}(w)$ στην οποία σε γραμμικό χρόνο εντοπίζεται η ελάχιστη απόσταση μεταξύ του κόμβου u και της λέξης-κλειδί w. Ο λόγος ύπαρξης του πίνακα κατακερματισμού είναι ότι η αναζήτηση αυτή αποφεύγεται και το επιθυμητό αποτέλεσμα είναι διαθέσιμο σε σταθερό χρόνο O(1).

Κατά την τροποποίηση του ευρετηρίου η αναγκαιότητα ύπαρξης του πίνακα κατακερματισμού M_{NK} δεν αμφισβητήθηκε. Ωστόσο, έπρεπε να γίνουν τροποποιήσεις όσο αφορά την πληροφορία που διέθετε αλλά και στην τρόπο με τον οποίο αυτή θα οργανωνόταν. Στα πλαίσια της γενικότερης λογικής ανεξαρτητοποίησης από την έννοια του ελαχίστου για κάθε κόμβο του γράφου επιλέχθηκε να καταγραφούν περισσότεροι του ενός δυνατοί τρόποι με τους οποίους μπορεί ένας κόμβος να προσεγγίσει μία λέξη-κλειδί. Μοναδικός περιορισμός: το μήκος των μονοπατιών που τους αντιστοιχούσε θα έπρεπε να είναι εντός των επιτρεπτών ορίων, για τους λόγους που αναφέρθηκαν κατά την κατασκευή των λιστών L_{KN}. Η επιλογή αυτή είχε ως αποτέλεσμα να αποθηκεύεται σε κάθε καταχώρηση $M_{NK}(u,w)$ ένα σύνολο δομών της μορφής (*dist, first, knode*) για τις οποίες ισχύει ότι ο *knode* είναι κόμβος που περιέχει τον όρο w και ότι η απόσταση *dist* θα είναι το πολύ ίση με το ανώτατο όριο που έχει οριστεί.

Καθίσταται σαφές ότι το μέγεθος του πίνακα κατακερματισμού γίνεται αρκετά μεγαλύτερο από ότι ήδη ήταν στην αρχική μορφή του SLINKS. Δημιουργήθηκαν προβλήματα, δεδομένης της περιορισμένης μνήμης που ήταν διαθέσιμη για την εκτέλεση των αλγορίθμων, και προέκυψε η ανάγκη για ριζική αλλαγή στον τρόπο με τον οποίο θα οργανωνόταν αποτελεσματικά η υπάρχουσα πληροφορία. Αποφασίστηκε, κατά αναλογία με τις λίστες L_{KN}, να δημιουργηθεί ένας πίνακας κατακερματισμού M_{NK}΄ για κάθε λέξη-κλειδί. Με τον τρόπο αυτό εξασφαλίστηκε ότι για κάθε εκτέλεση του αλγορίθμου θα έπρεπε να ληφθούν υπόψη μόνο οι πίνακες εκείνοι που αφορούσαν τους όρους του ερωτήματος που είχε εισάγει ο χρήστης.

Mnk' (philosopher)

Thing	2	Aristotle	Philosopher
Plato	1	Philosopher	Philosopher
Platonic_realism	2	Plato	Philosopher
Logic	2	Aristotle	Philosopher
Aristotle	1	Philosopher	Philosopher
Politics	2	Aristotle	Philosopher
Ontologists	2	Plato	Philosopher
Philosopher	0	Philosopher	Philosopher
Platonism	2	Plato	Philosopher

Mnk' (logic)

Thing	2	Aristotle	Logic
Plato	2	Aristotle	Logic
Platonic_realism	3	Plato	Logic
Logic	0	Logic	Logic
Aristotle	1	Logic	Logic
Politics	2	Aristotle	Logic
Ontologists	3	Plato	Logic
Philosopher	2	Aristotle	Logic
Platonism	3	Plato	Logic

Mnk' (platon)

Thing	2	Plato	Platonic_realism
	2	Plato	Platonism
Plato	1	Platonism	Platonism
	1	Platonic_realism	Platonic_realism
Platonic_realism	0	Platonic_realism	Platonic_realism
	2	Plato	Platonism
Logic	2	Aristotle	Platonism
	2	Aristotle	Platonic_realism
Aristotle	2	Plato	Platonic_realism
	2	Plato	Platonism
Politics	2	Plato	Platonism
	2	Plato	Platonic_realism
Ontologists	2	Plato	Platonism
	2	Plato	Platonic_realism
Philosopher	2	Plato	Platonic_realism
	2	Plato	Platonism
Platonism	0	Platonism	Platonism
	2	Plato	Platonic_realism

Εικόνα 4.7

Στον πίνακα M_{NK}' ως κλειδιά συναντούνται μόνο οι κόμβοι που μπορούσαν να "φτάσουν" σε αυτή. Για κάθε κλειδί u, ως τιμή $M_{NK}'(u)$ αποθηκεύεται ένα σύνολο δομών της μορφής (*dist, first, knode*) για τις οποίες ισχύει ότι ο *knode* είναι κόμβος που περιέχει τον όρο w,

ο first είναι ο πρώτο κόμβος του μονοπατιού μεταξύ των u και knode και ότι η απόσταση *dist* θα είναι το πολύ ίση με το ανώτατο όριο που έχει οριστεί.

4.3.2. Κατασκευή και αποθήκευση του τροποποιημένου ευρετηρίου

Με βάση την περιγραφή του τροποποιημένου ευρετηρίου, γίνεται αντιληπτό ότι σε κάθε εν δυνάμει λέξη-κλειδί αντιστοιχεί μία λίστα L_{KN} και ένας πίνακας κατακερματισμού M'_{NK}. Η ύπαρξη του περιορισμού στο μήκος των μονοπατιών κατέστησε μη αποδοτική τη χρήση του αλγορίθμου Dijksta προκειμένου να επιλεχθούν οι κόμβοι που θα λαμβάνονταν υπόψη στις δομές αυτές καθώς ο Dijkstra επιβάλλει τον υπολογισμό των συντομότερων μονοπατιών από μία κορυφή προς όλους τους κόμβους του γράφου. Επιπλέον, δεν αρκούσε η καταγραφή των συντομότερων μονοπατιών αλλά έπρεπε να γίνει σύγκρισή τους με το ανώτατο μήκος προκειμένου να διαπιστωθεί εάν θα ενταχθούν στις λίστες και στα σύνολα του πίνακα κατακερματισμού.

Ο αλγόριθμος που επιλέχθηκε έναντι του Dijkstra είναι ο αλγόριθμος αναζήτησης κατά πλάτος (Breadth-First Search). Αφού επιλεγεί μία κορυφή u ως εναρκτήρια, γίνεται επίσκεψη των υπόλοιπων κορυφών του γράφου με συγκεκριμένη σειρά η οποία καθορίζεται από την απόστασή της από τη u. Ο BFS προσαρμόζεται με μεγάλη ευκολία προκειμένου να ελέγχει την εκάστοτε απόσταση από την εναρκτήρια κορυφή αλλά και να διακοπεί σε περίπτωση που η απόσταση αυτή ξεπεράσει το όριο που έχει καθοριστεί. Σε αυτό το σημείο, πρέπει να σημειωθεί ότι, όπως αναφέρθηκε και στην ανάλυση του SPARK, κατά την διάσχισή του, ο γράφος της οντολογίας μετατρέπεται σε μη κατευθυνόμενο διότι σκοπός είναι ο εντοπισμός συνδέσεων μεταξύ επιθυμητών κορυφών του και όχι μονοπατιών με τον αυστηρό τρόπο που αυτά ορίζονται στη Θεωρία Γραφημάτων[30].

Η λίστα L_{KN} και ο πίνακας M'_{NK} που αντιστοιχούν σε μια λέξη-κλειδί w συμπληρώνονται παράλληλα. Κάθε κόμβος u που περιέχει την w αποτελεί εναρκτήρια κορυφή για μία εκτέλεση του BFS όπου επισκέπτονται πρώτα οι κόμβοι που συνδέονται με μονοπάτι μήκους ένα με τον u, μετά οι κόμβοι που συνδέονται με μονοπάτι μήκους δύο κ.ο.κ. Κάθε φορά που ο αλγόριθμος επισκέπτεται μία κορυφή v, πρέπει να εξασφαλιστεί ότι η απόστασή της από την κορυφή u είναι το πολύ ίση με το ανώτατο όριο που έχει καθοριστεί. Σε περίπτωση που το κριτήριο αυτό δεν ισχύει ο αλγόριθμος τερματίζει και επιλέγεται μία νέα κορυφή προκειμένου να αρχίσει μια νέα εκτέλεση του BFS.

Αν η απόσταση της κορυφής v από την u είναι εντός των επιτρεπτών ορίων, δημιουργείται αρχικά η απαιτούμενη καταχώρηση στη λίστα L_{KN} καθώς έχει πιστοποιηθεί ότι η λέξη-κλειδί w μπορεί να "φτάσει" την κορυφή v. Χάρις την επιλογή να θεωρηθεί ο γράφος της οντολογίας ως μη κατευθυνόμενος, συμπεραίνεται η εγκυρότητα της αντίστροφης πρότασης. Η κορυφή v μπορεί να "φτάσει" τη λέξη-κλειδί w και, συνεπώς, μπορεί να δημιουργηθεί η ανάλογη καταχώρηση στον πίνακα M'_{NK}.

[30] http://en.wikipedia.org/wiki/Path_(graph_theory)

Η αποθήκευση των δομών που περιγράφηκαν για όλες τις συμβολοσειρές εκείνες που ενδέχεται να αποτελέσουν όρους ερωτήματος του χρήστη έγινε, αρχικά, με τη βοήθεια δύο πινάκων κατακερματισμού όπου χρησιμοποιούνταν ως κλειδιά οι πιθανοί όροι. Στον έναν πίνακα ως τιμή αποθηκευόταν η αντίστοιχη λίστα L_{KN} ενώ στον άλλο ο αντίστοιχος πίνακας M'_{NK}. Η προσέγγιση αυτή δημιούργησε προβλήματα ως προς την μνήμη που έπρεπε να είναι διαθέσιμη σε κάθε εκτέλεση του αλγορίθμου και για τον λόγο αυτό κρίθηκε απαραίτητο να αντικατασταθεί.

Κάθε εν δυνάμει λέξη-κλειδί αντιστοιχείται μονοσήμαντα σε μία λίστα L_{KN} και σε ένα πίνακα M'_{NK}. Όπως θα φανεί και στην επόμενη παράγραφο, όπου περιγράφεται ο κορμός του αλγορίθμου, για τον εντοπισμό των απαντήσεών του ερωτήματος που εισάγει ο χρήστης αρκεί να διατηρούνται στη μνήμη οι λίστες L_{KN} και οι πίνακες M'_{NK} που αντιστοιχούν στους όρους του ερωτήματος. Βάσει της διαπίστωσης αυτής επιλέχθηκε για κάθε λέξη κλειδί w να αποθηκεύονται στο δίσκο τα αρχεία "w_A.Lkn" "w_A.Mnk", όπου w_A η συμβολοσειρά w αφού έχει υποστεί ανάλυση. Τα αρχεία αυτά περιέχουν, αντίστοιχα, τη λίστα και τον πίνακα που αφορούν την w.

Γίνεται αντιληπτό ότι σε κάθε εκτέλεση να γίνεται ανάγνωση των αρχείων εκείνων που χρειάζονται. Το γεγονός ότι η ανάγνωση από αρχείο αποτελεί χρονοβόρα διαδικασία έχει ληφθεί υπόψη και σε περίπτωση που υπήρχε μεγαλύτερη ποσότητα μνήμης διαθέσιμη δε θα προτιμούνταν. Ωστόσο, θα σημειωθεί ότι η επιλογή να αποθήκευσης των λιστών και των σε αρχεία αύξησε σημαντικά το μέγεθος του γράφου στο οποίο μπορούσαν να γίνουν πειράματα σχετικά με την εκτέλεση του SLINKS και το όφελος αυτό υπερκέρασε τις απώλειες σε χρόνο, οι οποίες τελικά ήταν αμελητέες.

4.3.3. Αλγόριθμος αναζήτησης με το ευρετήριο ενός επιπέδου

Στο σημείο αυτό περιγράφεται ο αλγόριθμος που εφαρμόζεται προκειμένου να συνδυαστούν όλες οι τροποποιημένες δομές που έχουν κατασκευαστεί με σκοπό τον εντοπισμό των απαντήσεων εκείνων που θα ενδιέφεραν το χρήστη. . Θεωρείται ότι έχει εισαχθεί ερώτημα με λέξεις-κλειδιά $q = (w_1, w_2, \ldots, w_m)$. Πρώτο βήμα είναι η ανάγνωση από το δίσκο των αρχείων εκείνων που αντιστοιχούν στην μορφή των όρων του ερωτήματος, αφού υποστούν ανάλυση.

Το τμήμα εκείνο που αφορά την αναζήτηση με κατεύθυνση προς τα πίσω (expanding backward) παραμένει ίδιο με την αρχική μορφή του αλγορίθμου. Διασχίζονται "παράλληλα" οι διαθέσιμες λίστες L_{KN} και κάθε στοιχείο τους αφορά μία κορυφή η οποία είναι πιθανόν να συνδέεται μέσω μονοπατιού με κόμβους που αντιστοιχούν σε όλους τους όρους του ερωτήματος του χρήστη. Έστω ότι μελετάται το ενδεχόμενο να διαθέτει την ιδιότητα αυτή κορυφή v η οποία εντοπίστηκε κατά τη διάσχιση της λίστας της λέξης-κλειδί w_2.

Γνωρίζοντας ότι η κορυφή v μπορεί να φτάσει την λέξη-κλειδί w_2, πρέπει να διαπιστωθεί εάν ισχύει το ίδιο για τις υπόλοιπες λέξεις-κλειδιά με τη βοήθεια της αναζήτησης προς τα εμπρός (expanding forward). Ελέγχεται, λοιπόν, εάν στους πίνακες κατακερματισμού M'_{NK} που αντιστοιχούν στις $w_1, w_3, \ldots, w_m$ εντοπίζεται ως κλειδί η κορυφή v .Σε περίπτωση που υπάρχει

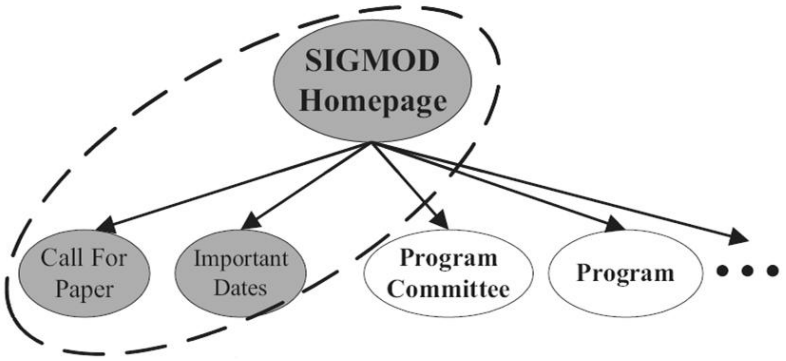
πίνακας στον οποίο δε μπορεί να εντοπιστεί η v, συμπεραίνεται ότι δε μπορεί να αποτελέσει κεντρική κορυφή ενός γράφου-απάντησης. Σε αντίθετη περίπτωση, παρατηρείται ότι τα στοιχεία που έχουν συγκεντρωθεί επιτρέπουν την κατασκευή περισσότερων από ένα γράφων απαντήσεων καθώς είναι διαθέσιμο ένα μονοπάτι μέσω του οποίου η v "φτάνει" την w_2, m_1 μονοπάτια μέσω των οποίων η v "φτάνει" την w_1, m_2 μονοπάτια μέσω των οποίων η v "φτάνει" την w_3 κ.ο.κ. Δημιουργούνται, λοιπόν, $m_1 \times 1 \times m_3 \times \cdots \times m_m$ γράφοι απαντήσεων και τα μονοπάτια από τα οποία αποτελούνται υπολογίζονται με τη βοήθεια του καρτεσιανού γινομένου των συνόλων που αντλήθηκαν από τους πίνακες M'_{NK}.

Κεφαλαιο 5

EASE: Μία αποδοτική μέθοδος αναζήτησης για αδόμητα (unstructured), ημιδομημένα (semi-structured) και δομημένα (structured) δεδομένα

5.1. Η προσέγγιση του EASE

Ο EASE (**E**fficient and **A**daptive Keyword **SE**arch method) αποτελεί έναν ακόμα αλγόριθμο που πραγματεύεται την αναζήτηση με λέξεις-κλειδιά και βασική αρχή του αποτελεί η μοντελοποίηση των δεδομένων έτσι ώστε να μπορούν να αναπαρασταθούν σε μορφή γράφου. Για παράδειγμα, στην περίπτωση αναζήτησης σε αρχεία κειμένου (text documents) αντιστοιχούνται τα αρχεία σε κόμβους του γράφου και οι υπερσύνδεσμοι μεταξύ τους στις ακμές του. Κατά ανάλογο τρόπο, στην περίπτωση μιας βάσης δεδομένων σε XML, αναπαριστούμε τα στοιχεία (elements) ως κόμβους και ως ακμές τις σχέσεις πατέρα-παιδιού (parent-child relationships). Στη μοντελοποίηση σχεσιακών βάσεων δεδομένων (relational databases) οι πλειάδες (tuples) αντιστοιχούν σε κόμβους και οι σχέσεις πρωτεύοντος και δευτερεύοντος κλειδιού (primary-foreign relationships) αντιστοιχούν στις ακμές. Δεδομένης της γραφοειδούς μοντελοποίησης, ο ορισμός του προβλήματος της αναζήτησης και ο τρόπος καθορισμού της λύσης του ακολουθούν ήδη χρησιμοποιημένα μοτίβα: Ο χρήστης εισάγει ένα ερώτημα με λέξεις-κλειδιά και ως απάντηση αναμένεται ένας υπογράφος της βάσης γνώσης που περιέχει κορυφές των οποίων οι ετικέτες αντιστοιχούν στους όρους του ερωτήματος.



Εικόνα 5.1

Πέρα από μια κοινή βάση, ο υπό μελέτη αλγόριθμος προσπαθεί να διαφοροποιηθεί από υπάρχουσες νόρμες και αυτό αποδεικνύεται από την προσαρμοστικότητα που επιδιώκει να επιτύχει προκειμένου να εφαρμόζεται σε δεδομένα ανεξάρτητα από το βαθμό δόμησής τους. Η ενασχόληση με αδόμητα δεδομένα (unstructured data), όπως αρχεία κειμένου (π.χ. αρχεία HTML), προέκυψε από το γεγονός ότι προηγούμενες μελέτες που έχουν πραγματοποιηθεί με θέμα τα ερωτήματα με λέξεις-κλειδιά παρουσιάζουν ως αποτέλεσμα μεμονωμένες σελίδες. Σε περίπτωση που δεν βρεθεί σελίδα που να περιέχει όλους τους όρους του ερωτήματος με βάση υπάρχουσες μηχανές επιστρέφονται σελίδες που περιέχουν κάποιους από αυτούς και κατατάσσονται βάση της σχετικότητας τους. Δεν υπάρχει καμία μέριμνα για το ενδεχόμενο εντοπισμού των λέξεων-κλειδιά σε ενδοσυνδεδεμένες (interrelated) σελίδες και την ενσωμάτωσή τους σε μία ενιαία και ουσιώδη απάντηση. Ως παράδειγμα αναφέρεται το ερώτημα “Conference 2008 Canada Data Integration” όπου ως απάντηση αναμένεται το SIGMOD 2008 που

φιλοξενήθηκε στον Καναδά με κύριο ερευνητικό θέμα "Data Integration" το οποίο, όμως, δεν εντοπίζεται στις 100 πρώτες απαντήσεις. Ο λόγος είναι ότι το SIGMOD 2008 υποδιαιρεί μεθοδικά τις πληροφορίες του σε πολλές σελίδες όπως φαίνεται στην εικόνα 5.1 με αποτέλεσμα η σελίδα IMPORTANT DATES να περιέχει τις λέξεις-κλειδιά "conference" και "2008" ενώ η "Data Integration" να περιέχεται στη σελίδα CALL FOR PAPERS.

Προηγούμενες μελέτες σε δομημένα ή ημιδομημένα δεδομένα κρίνουν απαραίτητο τον εντοπισμό των κόμβων που περιέχουν τους όρους του ερωτήματος του χρήστη και τη μετέπειτα εύρεση των πλησιέστερων σε αυτούς κοινών προγόνων (LCAs, Lowest Common Ancestors). Το ενδιαφέρον επικεντρώνεται στην επιλογή των υπόδεντρων που τελικά στέκουν περισσότερο εννοιολογικά ως απαντήσεις στο ερώτημα του χρήστη χωρίς να περιλαμβάνουν επιπρόσθετη και άχρηστη πληροφορία. Η εξασφάλιση της βέλτιστης επιλογής παραμένει ανοικτό πρόβλημα και η πρακτική που συνήθως ακολουθείται είναι εκείνη της καταγραφής των υπόδεντρων με το μικρότερο κόστος με χρήση μιας προσέγγισης της λύσης του προβλήματος του δέντρου Steiner (Steiner Tree Problem[31]). Ο τρόπος αυτός αντιμετώπισης εγείρει δυσκολίες καθώς το εν λόγω αλγοριθμικό πρόβλημα θεωρείται δύσκολο να λυθεί σε πολυωνυμικό χρόνο (NP-hard) με αποτέλεσμα να στερείται απόδοσης σε μεγάλους γράφους. Ταυτόχρονα, δεν επιτρέπει την εύρεση ποικιλόμορφων λύσεων με πολύπλοκη γραφοειδή δομή, όπως εκείνη των κύκλων, η οποία παρότι δυσχεραίνει τις απαιτούμενες διαδικασίες ενδέχεται είναι πλούσια σε πληροφορίες.

Στην κατεύθυνση της βελτιστοποίησης διαχείρισης των ερωτημάτων με λέξεις-κλειδιά, ο EASE αποτελεί ένα πρωτότυπο σύστημα που εξασφαλίζει προσαρμοστική και ευέλικτη οργάνωση του υπό επεξεργασία συνόλου δεδομένων. Παράλληλα παρέχει έναν αποδοτικό αλγόριθμο αναζήτησης των k καλύτερων αποτελεσμάτων με τη βοήθεια ενός καινοτόμου ευρετηρίου το οποίο δεν παραπέμπει στην ευρέως χρησιμοποιούμενη ανεστραμμένη εκδοχή που κατά κόρον χρησιμοποιείται σε εφαρμογές όπου απαιτείται ανάκληση πληροφορίας.

5.2. Το πρόβλημα του Γράφου Steiner Ακτίνας r

Ο EASE, όπως αναφέρθηκε και παραπάνω, μοντελοποιεί τα δεδομένα σε μορφή γράφου, ανεξάρτητα από το αν ή κατά πόσο είναι δομημένα, και το πρόβλημα της απάντησης ενός ερωτήματος με λέξεις-κλειδιά ανάγεται σε αναζήτηση σε γράφο. Έχοντας ως έμπνευση το πρόβλημα του δέντρου Steiner (Steiner Tree Problem), εισάγεται το πρόβλημα του γράφου Steiner (Steiner Graph Problem) το οποίο έχει ιδιαίτερο ενδιαφέρον. Το γεγονός ότι αν η απόσταση μεταξύ δύο κόμβων είναι υπερβολικά μεγάλη, η πληροφορία που περιέχεται στο μονοπάτι που τους ενώνει δεν έχει αρκετή χρησιμότητα, οδήγησε στην ανάγκη περιορισμού της. Η παραδοχή αυτή οδήγησε με τη σειρά της στον ορισμό του προβλήματος του γράφου Steiner ακτίνας r (r-Radius Steiner Graph Problem) το οποίο αποτελεί μια πρόκληση στην παραγωγή εννοιολογικά πλούσιων γράφων με αποδεκτό μέγεθος.

Ακολουθούν έννοιες απαραίτητες για την κατανόηση του εν λόγω προβλήματος καθώς και ο αυστηρός ορισμός του. Εφαρμογή αυτών, με σκοπό την καλύτερη κατανόησή τους, γίνεται

[31] http://en.wikipedia.org/wiki/Steiner_tree_problem

στη βάση της εικόνας 5.2, η οποία παρουσιάζεται τόσο στην αρχική όσο και στη μοντελοποιημένη μορφή της.

Authors

AID	Name
a1	J. Shanmugasundaram
a2	L. Guo
a3	V. Hristidis
a4	Y. Papakonstantinou
a5	A. Balmin

Paper-Reference

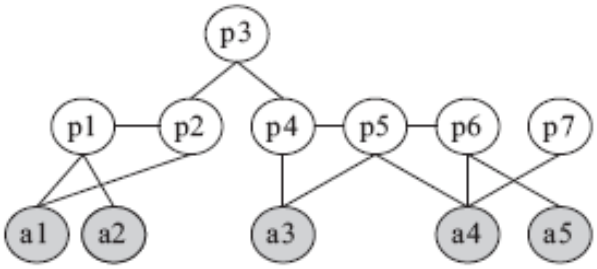
PID	citedPID
p1	p2
p2	p3
p3	p4
p4	p5
p5	p6

Author-Paper

AID	PID
a1	p1
a1	p2
a2	p1
a3	p4
a3	p5
a4	p5
a4	p6
a4	p7
a5	p6

Papers

PID	Title
p1	Topology Search over Biological Databases
p2	XRANK: Ranked Keyword Search over XML Documents
p3	Bidirectional Expansion for Keyword Search on Graphs
p4	Finding *top-k* Answers in Keyword Proximity Search
p5	Efficient IR-Style Keyword Search over Relational Databases
p6	Keyword Proximity Search on XML Graphs
p7	DISCOVER: Keyword Search in Relational Databases



Εικόνα 5.2

Κεντρική Απόσταση (Centric Distance)

Δοθέντος ενός γράφου G, υπολογίζουμε, για τυχαία επιλεγμένο κόμβο υ το μέγεθος D(υ,u) που αντιστοιχεί στην απόστασή του, δηλαδή στο μήκος του συντομότερου μονοπατιού, από κάθε άλλο κόμβο u του γράφου. Η μέγιστη αυτών των αποστάσεων, $\boldsymbol{max_{u \in G}\{D(v,u)\}}$, ονομάζεται κεντρική απόσταση και συμβολίζεται CD(υ).

Ακτίνα (Radius)

Η ακτίνα ενός γράφου G συμβολίζεται ως R(G) και αντιστοιχεί στην ελάχιστη όλων των κεντρικών αποστάσεων κόμβων του G, $\boldsymbol{min_{v \in G}\{CD(v)\}}$. Αν η ακτίνα του G είναι ακριβώς r τότε τον καλούμε γράφο ακτίνας r (r-radius graph).

Γράφος Steiner ακτίνας r

Δοθέντος ενός γράφου G ακτίνας r και ενός ερωτήματος K αποτελούμενο από λέξεις-κλειδιά, καλούνται κόμβοι περιεχομένου (content nodes) οι κόμβοι εκείνοι στους οποίους αντιστοιχεί ετικέτα που περιλαμβάνει μία λέξη-κλειδί. Ένας κόμβος s χαρακτηρίζεται κόμβος Steiner (Steiner node) σε περίπτωση που υπάρχουν δύο κόμβοι περιεχομένου, έστω u και υ, και ο s βρίσκεται στο μονοπάτι που τους ενώνει ενώ μπορεί να ταυτίζεται και με κάποιον από αυτούς. Ο υπογράφος του G που αποτελείται από κόμβους Steiner και τις συσχετιζόμενες ακμές καλείται γράφος Steiner ακτίνας r. Η ακτίνα ενός γράφου Steiner που έχει κατασκευαστεί με βάση τον G μπορεί να είναι μικρότερη, ίση αλλά όχι μεγαλύτερη του r.

Η χρήση ενός απλού υπογράφου του αρχικού ως απάντηση στο ερώτημα του χρήστη αποτελεί έναν ουσιώδη και συμπαγή τρόπο αναπαράστασης χρήσης πληροφορίας ενώ περιέχουν και επιπρόσθετες κορυφές που ενδέχεται να ενισχύσουν την καταλληλότητά τους. Ωστόσο, οι Steiner γράφοι είναι πιο σαφής καθώς έχει απορριφθεί το σύνολο των κορυφών που δεν είναι απαραίτητες και οι οποίες πιθανόν να προκαλούσαν επιβάρυνση. Στην γραφοειδή μορφή της βάσης της εικόνας 5.2, η απόσταση μεταξύ των κορυφών a_1 και a_3 είναι τέσσερα [$D(a_1, a_3) = 4$], η κεντρική απόσταση της κορυφής p_2 είναι πέντε [$CD(p_2)=5$] και η ακτίνα του γράφου είναι τέσσερα [$R(G) = 4$]. Δοθέντος του ερωτήματος "IR Hristidis", υπολογίζεται ως απάντηση ο γράφος Steiner που αποτελείται από τους κόμβους p_4,p_5,a_3 και τις μεταξύ τους ακμές. Σε αντίθεση με το αντίστοιχο δέντρο Steiner που αποτελείται από τις κορυφές p_5,a_3 και που έχει προταθεί σε προηγούμενες μελέτες, ο γράφος αποτρέπει την απώλεια σημαντικής πληροφορίας, ιδιαίτερα σημαντική ιδιότητα στην περίπτωση των βάσεων δεδομένων οι οποίες έχουν πολύπλοκη μορφή.

Το πρόβλημα του Γράφου Steiner Ακτίνας r

Δοθέντος ενός γράφου G ακτίνας r και ενός ερωτήματος K αποτελούμενο από λέξεις-κλειδιά, το πρόβλημα του γράφου Steiner ακτίνας r αφορά στην εύρεση όλων των γράφων Steiner ακτίνας r που παράγονται με βάση το G και οι οποίοι περιέχουν όλους τους όρους του ερωτήματος, ή τους περισσότερους από αυτούς, αλλά και στην ταξινόμησή τους ανάλογα με το πόσο σχετίζονται με το αρχικό ερώτημα.

5.3. Ένας προσαρμοστικός αλγόριθμος αναζήτησης

Η αποδοτικότητα και τα πλεονεκτήματα της χρήσης του ανεστραμμένου ευρετηρίου (inverted index) για την διευκόλυνση του υπολογισμού των "καλύτερων" απαντήσεων σε ερωτήματα με λέξεις-κλειδιά είναι ευρέως αναγνωρισμένα. Ωστόσο, η χρήση τέτοιων δομών δεν είναι το ίδιο αποτελεσματική κατά τον εντοπισμό πλούσιων σχέσεων από πλευράς δομής τις οποίες συναντούμε σε βάσεις δεδομένων. Κρίνεται, λοιπόν, σημαντική, η γρήγορη και πλήρης εύρεση αυτών των δομικών σχέσεων αλλά και η άμεση και ακριβής απάντηση σε ερωτήματα του χρήστη. Η πιο "αφελής" προσέγγιση θα αφορούσε την καταγραφή όλων των συνδυασμών που μπορούν να προκύψουν από τις πιθανές λέξεις-κλειδιά και την κατασκευή των γράφων

Steiner ακτίνας r για καθένα από τους συνδυασμού αυτούς. Ο μεγάλος αριθμός των συνδυασμών σε μία ρεαλιστική βάση δεδομένων είναι τόσο μεγάλος που καθιστά ασύμφορο από πλευράς κόστους έναν τέτοιο υπολογισμό.

Ο EASE, έχοντας λάβει υπόψη τα παραπάνω, προτείνει μια αποδοτική στρατηγική που θα επιτρέπει τον εντοπισμό ενός ποσοστού απλών γράφων ακτίνας r ανάλογου του αριθμού των κορυφών του αρχικού γράφου. Η ευελιξία που παρέχει έγκειται στο ότι απαιτείται μόνο η ευρετηριοποίηση και ο υπολογισμός μόνο αυτού του ποσοστού γράφων το οποίο είναι επαρκές για την κατασκευή όλων των Steiner γράφων που προκύπτουν.

5.3.1. Ο Πίνακας Πρόσπτωσης

Ο Πίνακας Πρόσπτωσης M = $\left(\mathrm{m_{ij}}\right)_{|V|\times|V|}$ αποτελεί τον πιο διαδεδομένο τρόπο αναπαράστασης ενός γράφου G(V,E). Πρόκειται για έναν πίνακα διαστάσεων $|V| \times |V|$ όπου η ποσότητα $|V|$ ισούται με τον πληθάριθμο του συνόλου των κορυφών του γράφου. Κάθε στοιχείο $\mathrm{m_{ii}}$ έχει τιμή ένα (1). Ένα στοιχείο του $\mathrm{m_{ij}}$, όπου $i \neq j$, έχει την τιμή ένα (1) αν και μόνο αν υπάρχει, μεταξύ των κορυφών i και j, ακμή, ή αλλιώς μονοπάτι μήκους ένα που να τις συνδέει. Σε αντίθετη περίπτωση, τοποθετούμε στην κατάλληλη θέση του πίνακα ένα μηδενικό.

Πίνακας M

	a1	p1	a2	p2	p3	p4	a3	p5	a4	p6	p7	a5
a1	1	1	0	1	0	0	0	0	0	0	0	0
p1	1	1	1	1	0	0	0	0	0	0	0	0
a2	0	1	1	0	0	0	0	0	0	0	0	0
p2	1	1	0	1	1	0	0	0	0	0	0	0
p3	0	0	0	1	1	1	0	0	0	0	0	0
p4	0	0	0	0	1	1	1	1	0	0	0	0
a3	0	0	0	0	0	1	1	1	0	0	0	0
p5	0	0	0	0	0	1	1	1	1	1	0	0
a4	0	0	0	0	0	0	0	1	1	1	1	0
p6	0	0	0	0	0	0	0	1	1	1	0	1
p7	0	0	0	0	0	0	0	0	1	0	1	0
a5	0	0	0	0	0	0	0	0	0	1	0	1

Εικόνα 5.3

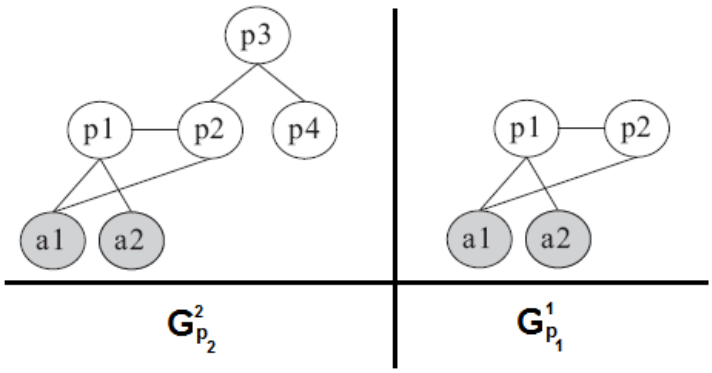
Η r-οστή δύναμη του πίνακα M είναι ο πίνακας $M^r = M \times M \times \cdots \times M = \left(\mathrm{m_{ij}^r}\right)_{|V|\times|V|}$. Κατά αναλογία με τον πίνακα M, ένα στοιχείο $\mathrm{m_{ij}^r}$ έχει την τιμή ένα (1) αν και μόνο αν υπάρχει, μονοπάτι μήκους όχι μεγαλύτερου από r που να συνδέει τις κορυφές i και j (Εικόνα 5.3). Στον πίνακα Mr ορίζεται για κάθε κορυφή u_i του γράφου, στην οποία αντιστοιχεί και η i γραμμή, το σύνολο $N_i^r = \{u_j | M_{ij}^r = 1\}$ αποτελούμενο από τις κορυφές εκείνες οι οποίες συνδέονται με μονοπάτι μήκους το πολύ r με την κορυφή i.

Πίνακας M^2

	a1	p1	a2	p2	p3	p4	a3	p5	a4	p6	p7	a5
a1	1	1	1	1	1	0	0	0	0	0	0	0
p1	1	1	1	1	1	0	0	0	0	0	0	0
a2	1	1	1	1	0	0	0	0	0	0	0	0
p2	1	1	1	1	1	1	0	0	0	0	0	0
p3	1	1	0	1	1	1	1	1	0	0	0	0
p4	0	0	0	1	1	1	1	1	1	1	0	0
a3	0	0	0	0	1	1	1	1	1	1	0	0
p5	0	0	0	0	1	1	1	1	1	1	1	1
a4	0	0	0	0	0	1	1	1	1	1	1	1
p6	0	0	0	0	0	1	1	1	1	1	1	1
p7	0	0	0	0	0	0	0	1	1	1	1	0
a5	0	0	0	0	0	0	0	1	1	1	0	1

Εικόνα 5.4

Με βάση τα στοιχεία αυτά, κατασκευάζεται ο G_i^r (ή αλλιώς $G_{u_i}^r$) ο οποίος αποτελεί υπογράφο του G ως προς την i-οστή γραμμή του M^r και αποτελείται από το σύνολο κορυφών N_i^r και τις συσχετιζόμενες ακμές. Η κατασκευή ενός γράφου G_{u_i} είναι απαραίτητο να συνοδεύεται από τον έλεγχο συνθηκών που θα εξασφαλίσουν ότι η ακτίνα του είναι όντως r. Συγκεκριμένα, ο G_i^r έχει ακτίνα r εάν για κάθε κορυφή u_k, η οποία συνδέεται με μονοπάτι μήκους το πολύ r με την κορυφή u_i, το σύνολο N_i^r δεν είναι υποσύνολο του N_k^{r-1}, $N_i^r \nsubseteq N_k^{r-1}$. Με απλά λόγια, εάν υπάρχει κάποια κορυφή, πέραν της u_i η οποία μπορεί να συνδεθεί με τις ίδιες κορυφές του γράφου με μονοπάτια το πολύ r-1 τότε την τιμή αυτή θα λαμβάνει η ακτίνα του. Για παράδειγμα, η κορυφή p_2 συνδέεται με μονοπάτι μήκους δύο με τις κορυφές $N_{p_2}^2 = \{a_1, p_1, a_2, p_2, p_3, p_4\}$ ο γράφος $G_{p_2}^2$ έχει ακτίνα 2. Εν αντιθέσει, ο $G_{a_2}^2$ έχει ακτίνα 1 καθώς $N_{p_2}^2 = \{a_1, p_1, a_2, p_2\}$ και $N_{p_1}^1 = \{a_1, p_1, a_2\}$ με αποτέλεσμα να ισχύει $G_{a_2}^2 \subseteq G_{p_1}^1$.



Εικόνα 5.5

5.3.2. Κατασκευή των γράφων Steiner ακτίνας r

Μετά την προετοιμασία της απαραίτητης υποδομής, δημιουργήθηκε η ανάγκη ύπαρξης ενός μηχανισμού που θα επιτρέπει την άμεση ανάκλησή των γράφων που κάθε φορά χρειάζονταν και στην κατεύθυνση αυτή κατασκευάστηκε ένα καινοτόμο ευρετήριο γράφων. Οι καταχωρήσεις του ευρετηρίου αντιστοιχούν στις εν δυνάμει λέξεις κλειδιά που περιέχονται στις ετικέτες των κορυφών του γράφου και καθεμία από αυτές διατηρεί το σύνολο των γράφων ακτίνας r που περιέχουν τον όρο, δηλαδή, σε κάθε όρο k_i αντιστοιχεί ένα σύνολο $I_{k_i} = \left\{G_{u_j}^r | G_{u_j}^r \, περιέχει \, τον \, k_i\right\}$. Πρέπει να διευκρινιστεί ότι ένας υπογράφος του αρχικού περιέχει μια λέξη-κλειδί όταν εκείνη συνδέεται με την ετικέτα μίας από τις κορυφές του.

Η επεξεργασία του ερωτήματος $K = \{k_1, k_2, \dots, k_m\}$ απαιτεί ως πρώτο βήμα την ανάκληση του συνόλου I_{k_i} για κάθε λέξη-κλειδί βάσει του ευρετηρίου που έχει κατασκευαστεί και τον υπολογισμό της ένωση αυτών, $\bigcup_{i=1}^{m} I_{k_i}$, έτσι ώστε να παραχθεί ένα ολοκληρωμένο σύνολο γράφων για τους οποίους είναι γνωστό ότι περιέχουν τουλάχιστον μία λέξη κλειδί. Το επόμενο βήμα επικεντρώνεται στην διατήρηση των Steiner κόμβων κάθε γράφου και την απόρριψη των υπόλοιπων ώστε να παραχθούν οι ζητούμενοι γράφοι Steiner.

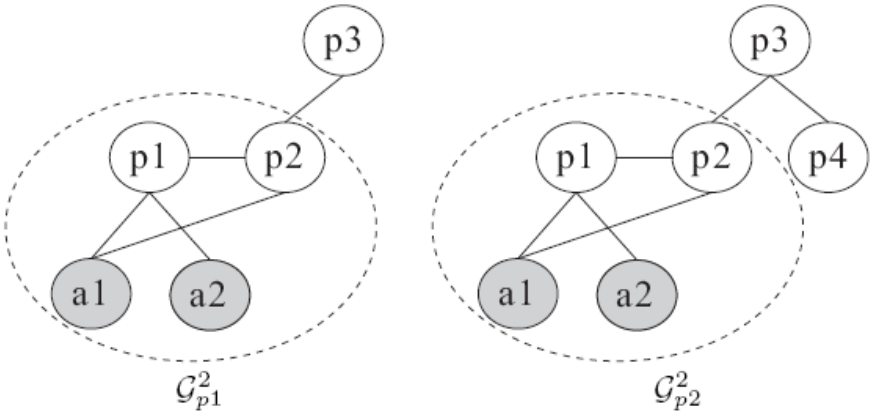
Δοθέντος ενός γράφου G^r συνοδευόμενου από το σύνολο κόμβων περιεχομένου $c_1, c_2, \dots, c_q$ που αντιστοιχεί στις κορυφές εκείνες που διαθέτουν ετικέτες που περιέχουν όρους του ερωτήματος εισόδου, αναμένεται ο εντοπισμός των κόμβων Steiner και η παραγωγή των αντίστοιχων γράφων σύμφωνα με την ακόλουθη διαδικασία:

i. Παράγεται ένα αντίγραφο του G^r από το οποίο αφαιρούνται όλοι οι κόμβοι περιεχομένου με την εξαίρεση του c_i και συγκεντρώνεται το σύνολο $P(c_i)$ που αποτελείται από τις κορυφές εκείνες για τις οποίες υπάρχει μονοπάτι που να τις συνδέει με τον c_i. Η διαδικασία αυτή επαναλαμβάνεται για κάθε μέλος του συνόλου $c_1, c_2, \dots, c_q$.

ii. Υπολογίζεται το σύνολο $P = \left\{\bigcup_{i=1}^{q} \bigcup_{j=i+1}^{q} \left(P(c_i) \cap P\left(c_j\right)\right)\right\} \cup \{c_1, c_2, \dots, c_q\}$ τα μέλη του οποίου αντιστοιχούν στους κόμβους Steiner.

iii. Κατασκευάζεται ο γράφος Steiner, υπογράφος του G^r που αποτελείται από τις κορυφές του συνόλου P και τις συσχετιζόμενες ακμές.

Εφαρμόζοντας τα παραπάνω βήματα στη βάση της εικόνας 5.2, και επιχειρώντας να απαντηθεί το ερώτημα "Shanmugasundaram Guo XRANK" αναζητούνται ως απαντήσεις γράφοι ακτίνας 2. Οι κόμβοι p_1 και p_2 αποτελούν κόμβους περιεχομένου και οδηγούν στον υπολογισμό των $G_{p_2}^2$ και $G_{p_1}^2$, όπως αυτά φαίνονται στην εικόνα 5.6. Οι αντίστοιχοι γράφοι Steiner είναι κυκλωμένοι μα διακεκομμένη γραμμή.



Εικόνα 5.6

5.4. Οι τροποποιήσεις στον EASE

Ο αλγόριθμος EASE αποτέλεσε μία καινοτόμα και προσεγμένη πρόταση που δεν επιδεχόταν μεγάλων τροποποιήσεων. Επιλέχθηκε η ενίσχυσή του με ένα στάδιο προεπεξεργασίας που συμπεριλάμβανε τον υπολογισμό των γράφων ακτίνας r και αποθήκευση αυτών σε έναν πίνακα κατακερματισμού. Ακόμα, προκειμένου να υλοποιηθεί σωστά, χρειάστηκε να ληφθούν αρκετά μέτρα κατά το στάδιο της ευρετηριοποίησης του γράφου της οντολογίας. Ειδική αναφορά γίνεται στην αλλαγή του πίνακα πρόσπτωσης.

Η δομική μονάδα του προτύπου RDF είναι η πρόταση (statement) αποτελούμενη από το υποκείμενο (subject), το αντικείμενο (object) και το κατηγόρημα (predicate). Η έννοια της σύνδεσης σε μία πρόταση, όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, μπορεί να αναπαρασταθεί πιο απτά μοντελοποιώντας το κατηγόρημα ως βέλος (arc) με φορά από το υποκείμενο προς το αντικείμενο τα οποία αντιστοιχούν σε κορυφές (nodes). Η γραφοειδής αυτή δομή του προτύπου RDF επιτρέπει την αναπαράστασή του με τη βοήθεια ενός πίνακα πρόσπτωσης ενώ δημιουργούνται ερωτήματα σχετικά με τον τρόπο υλοποίησης.

Ο πίνακας πρόσπτωσης $M = (m_{ij})_{|V|\times|V|}$ αποτελεί τρόπο αναπαράστασης ενός γράφου G(V,E) και έχει διαστάσεις $|V| \times |V|$, όπου $|V|$ ο πληθάριθμος του συνόλου των κορυφών. Τα στοιχεία m_{ii}, που βρίσκονται στη διαγώνιό του, αλλά και τα στοιχεία m_{ij}, για τα οποία υπάρχει μονοπάτι μεταξύ των κορυφών i και j, έχουν τιμή ένα· τα υπόλοιπα στοιχεία του έχουν τιμή μηδέν. Η προετοιμασία για την κατασκευή της δομής αυτής είχε ήδη γίνει από το στάδιο της προεπεξεγασίας όπου κατά την δημιουργία ακέραιων αναγνωριστικών σε κάθε πόρο της βάσης γνώσης δόθηκε προτεραιότητα σε εκείνους που αντιστοιχούσαν σε στιγμιότυπα (instances), κλάσεις (classes) και κατηγορίες (categories). Με τον τρόπο αυτό το αναγνωριστικό κάθε κόμβου του γράφου θα μπορούσε να αποτελέσει τον αύξοντα αριθμό μιας γραμμής ή μιας στήλης του πίνακα πρόσπτωσης. Ταυτόχρονα εξασφαλίστηκε η ύπαρξη συνέχειας με την έννοια ότι καμία γραμμή ή στήλη δεν υπήρχε περίπτωση να μείνει κενή και συνεπώς χωρίς κάποια χρησιμότητα, επιβαρύνοντας τον απαιτούμενο χώρο αποθήκευσης.

Βάσει του τρόπου κατασκευής του πίνακα πρόσπτωσης αλλά και δεδομένου του μοντέλου RDF, διαπιστώνεται ότι η δυαδική λογική σύμφωνα με την οποία ο άσσος σε ένα στοιχείο του πίνακα υποδηλώνει την ύπαρξη ακμής μεταξύ δύο κορυφών του γράφου δεν είναι αρκετή καθώς υπάρχει απώλεια της πληροφορίας που αφορά το κατηγόρημα. Επιλέχθηκε, λοιπόν, τα στοιχεία m_{ij} για τα οποία υπάρχει μονοπάτι μεταξύ των πόρων i και j, να συμπληρωθούν με το αναγνωριστικό του κατηγορήματος που τους συνδέει δίνοντας έτσι τη δυνατότητα για πλήρη εποπτεία του γράφου αλλά και ανακατασκευή των επιθυμητών μονοπατιών. Σε αυτό το σημείο, υπενθυμίζεται ότι με σκοπό τον εντοπισμό περισσότερων αποτελεσμάτων ο γράφος που κατασκευάστηκε βάσει των προτάσεων RDF προτιμήθηκε να μην δοθεί αρκετή έμφαση στην κατεύθυνση των ακμών. Αυτόματα ο πίνακας πρόσπτωσης γίνεται συμμετρικός καθώς για κάθε στοιχείο m_{ij} ισχύει $m_{ij} = m_{ji}$.

Κλείνοντας, πρέπει να σημειωθεί ότι ο αλγόριθμος EASE δε χρησιμοποιήθηκε κατά την εκτέλεση των πειραμάτων καθώς η μνήμη που διέθετε το σύστημα ήταν κατά πολύ ανεπαρκής συγκριτικά με τις απαιτήσεις του. Ωστόσο, η υλοποίησή του έδωσε μια άλλη διάσταση στον τρόπο αντιμετώπισης της αναζήτησης και αποτέλεσε βάση για νέους πειραματισμούς αλλά και για την ανατροφοδότηση και περαιτέρω αλλαγή των υπολοίπων αλγορίθμων.

Κεφαλαιο 6
Πειραματικά Αποτελέσματα και Μελλοντική Εργασία

6.1. Το υποσύνολο της DBpedia

Οι αλγόριθμοι που αναλύονται στα πλαίσια της εργασίας αυτής υλοποιήθηκαν με τις βιβλιοθήκες Lucene (για ανάκτηση πληροφορίας) και JgraphT (για αναπαράσταση και επεξεργασία γράφων) να κατέχουν κεντρικό ρόλο. Η απόδοση τους ελέγχθηκε με βάση τον χρόνο που απαιτούσαν για να επιστρέψουν τους γράφους-απαντήσεις σε ερωτήματα με λέξεις-κλειδιά. Το σύνολο δεδομένων στο οποίο εφαρμόστηκαν, όπως έχει επισημανθεί στο κεφάλαιο 2, έχει παραχθεί με τη βοήθεια επιλεγμένων αρχείων της DBpedia, τα οποία επέτρεπαν την κατασκευή συνεκτικών γράφων με ενδιαφέρον.

Το σύνολο της DBpedia περιγράφει 3.64 εκατομμύρια «πράγματα» και μισό εκατομμύριο «γεγονότα» με αποτέλεσμα ο γράφος της οντολογίας να αποτελείται από ανάλογο αριθμό κόμβων και πλευρών αντίστοιχα. Δεδομένου ότι κάθε κόμβος αναπαρίσταται ως ακέραιος αριθμός και κάθε πλευρά ως μια δομή αποτελούμενη από τρεις ακεραίους που αντιστοιχούν στα άκρα και στην ετικέτα της, γίνεται αντιληπτό ότι απαιτούνται μεγάλες ποσότητες μνήμης για την αποθήκευσή του. Θεωρήθηκε απαραίτητο να δημιουργηθεί ένας μικρότερος γράφος, υπογράφος του αρχικού, ο οποίος θα ήταν περισσότερο ευέλικτος και θα επέτρεπε να υπάρχει εποπτεία των αποτελεσμάτων.

Με τη βοήθεια κώδικα γραμμένου σε Perl αποσπάστηκαν τμήματα των διαθέσιμων αρχείων έτσι ώστε να παραχθεί ένα υποσύνολο των τριπλετών της υπάρχουσας οντολογίας. Η κατασκευή αυτή δεν έγινε με τυχαίο τρόπο αλλά με συνεχή έλεγχο των νέων κορυφών και ακμών που εισάγονταν στον υπογράφο έτσι ώστε να διατηρείται η συνεκτικότητά του. Συγκεκριμένα, αφού επιλέχθηκε, με τυχαίο τρόπο ένα σύνολο τριπλετών από το αρχείο mapping_based_properties_en.nt καταγράφηκαν τα URI των συμπεριλαμβανομένων πόρων. Με βάση τα URI αυτά αντλήθηκαν από τα υπόλοιπα αρχεία οι τριπλέτες που τα περιείχαν εξασφαλίζοντας τη συνοχή.

Τα πειράματα διεξήχθηκαν σε μηχάνημα με επεξεργαστή Intel Core i5 στα 2.67GHz και μνήμη 2GB, εκ των οποίων ήταν διαθέσιμα τα 1,74GB. Προσπαθώντας να γίνει πλήρης εκμετάλλευση των διαθέσιμων πόρων, κατασκευάστηκαν πολλοί διαφορετικοί υπογράφοι του αρχικού. Τελικά, ως βάση διεξαγωγής των πειραμάτων επιλέχθηκε γράφος με 6716 κόμβους και 11423 ακμές.

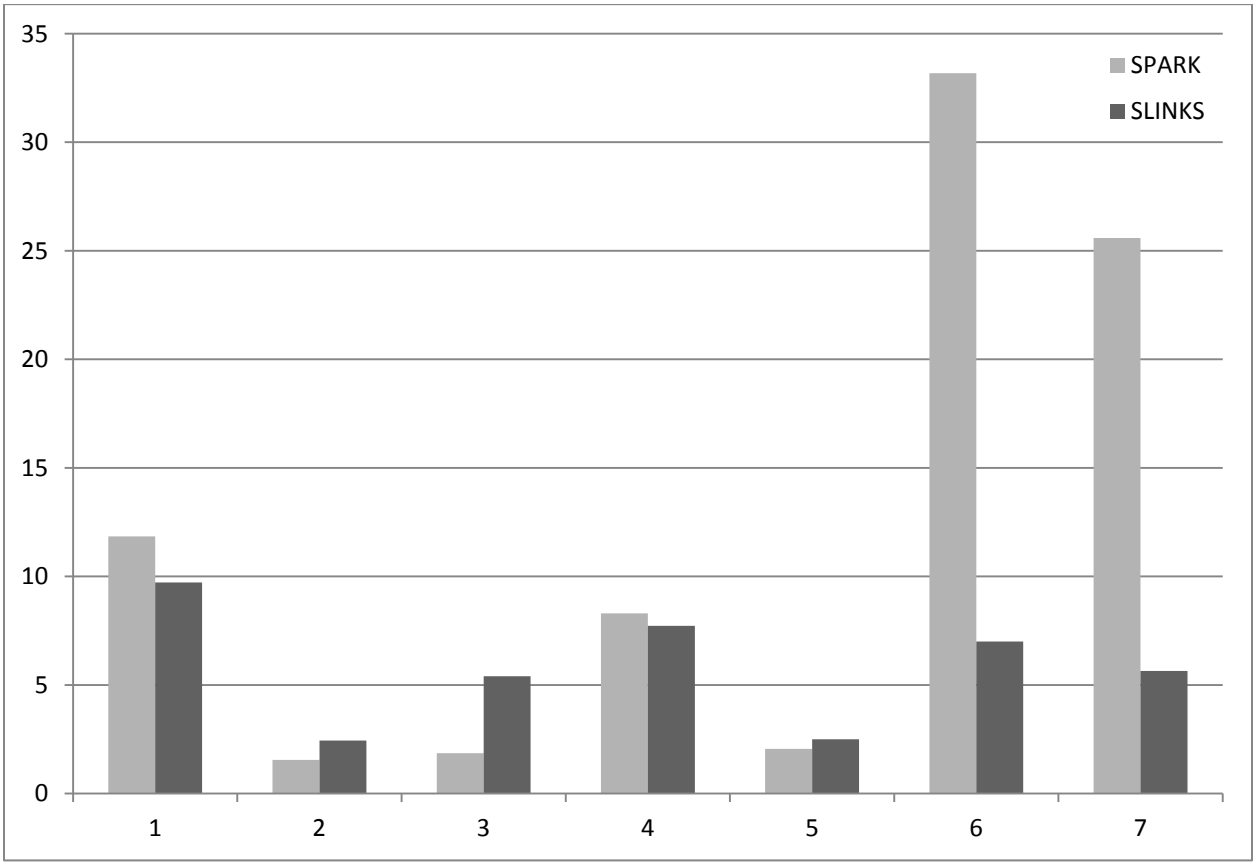
6.2. Πειράματα και Συμπεράσματα

Οι αλγόριθμοι που υλοποιήθηκαν απαιτούσαν, σε κάθε εκτέλεση, να είναι διαθέσιμες στη μνήμη δομές, είτε ήδη κατασκευασμένες είτε σταδιακά δημιουργούμενες, που απαιτούσαν μεγάλο αποθηκευτικό χώρο. Παρά τις προσπάθειες για αποδοτική τροποποίηση των δομών, όπως και τα νούμερα μαρτυρούν, η ποσότητα μνήμης συνέχισε να είναι περιοριστική με αποτέλεσμα να μην είναι δυνατό να καταδειχθούν όλες οι αξιοσημείωτες περιπτώσεις στα πειράματα που εκτελέστηκαν. Παρόλα αυτά, έγινε προσπάθεια να δοκιμαστούν περιπτώσεις τέτοιες που να καταδεικνύουν τις αδυναμίες αλλά και τη συνεισφορά κάθε αλγορίθμου.

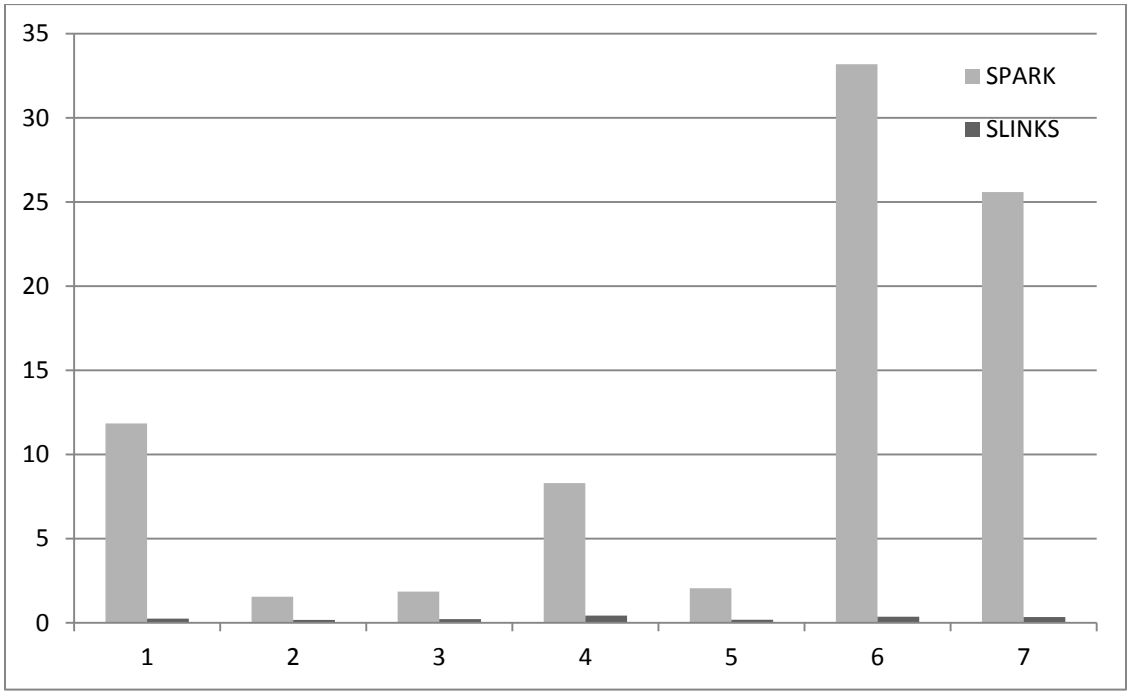
Στον πίνακα που ακολουθεί καταγράφονται τα ερωτήματα που δοκιμάστηκαν συνοδευόμενα από τον αριθμό των κόμβων του γράφου που αντιστοιχούσαν σε κάθε λέξη-κλειδί. Σημειώνεται ότι ο SPARK έχει επιλεγεί να εντοπίζει μονοπάτια μήκους όχι μεγαλύτερου από 5 ενώ ο BLINKS, επιτρέποντας μονοπάτια μήκους 4, δίνει τη δυνατότητα σε δύο κόμβους που περιέχουν λέξεις-κλειδιά να απέχουν απόσταση ίση το πολύ με 8.

	Query	**# Keyword Nodes**
1	history mathematics	(50, 11)
2	aristotle prolemy	(4 , 3)
3	logic alexander	(4 , 12)
4	ptolemy astronomer alexander	(3, 11, 12)
5	plato reason syllogism	(3 , 2 , 1)
6	mathematics america aristotle	(11, 23 , 4)
7	ptolemy astronomer religious ontologists	(3, 11,13 ,1)
	mathematics alexander aristotle reason	(11 , 12 , 4 , 2)

Ο SLINKS, όπως άλλωστε ήταν αναμενόμενο και βάσει της δομής του, είχε καλύτερα αποτελέσματα συγκριτικά με τον SPARK, όπως φαίνεται και στο γράφημα που ακολουθεί.



Ωστόσο, θα πρέπει να σημειωθεί ότι με σκοπό να λυθούν προβλήματα που αφορούσαν τις ελλείψεις σε μνήμη, ο SLINKS τροποποιήθηκε με τρόπο τέτοιο ώστε οι δομές που είχαν κατασκευαστεί κατά το στάδιο προεπεξεργασίας του να διασπαστούν και να αποθηκευτούν σε αρχεία. Η ανάγνωση από τα αρχεία αυτά κατά την εκτέλεση του αλγορίθμου στοίχησε πολύ στο χρόνο του και για να γίνει μια πιο καθοριστική σύγκριση του SLINKS με τον SPARK ως προς την απόδοση του αμιγώς αλγοριθμικού τμήματός τους παρατίθεται το ακόλουθο διάγραμμα. Παρότι τα αποτελέσματα φαίνονται να απέχουν πολύ αγγίζοντας διαφορές που δε φαίνονται πραγματικές, σημειώνεται ότι είναι ρεαλιστικά καθώς η προεπεξεργασία που έχει γίνει στον BLINKS ανάγει τον εντοπισμό κόμβων-απαντήσεων σε διάσχιση λιστών και έλεγχο καταχωρήσεων σε πίνακες κατακερματισμού για αριθμό κόμβων πολύ μικρότερο από εκείνο του συνολικού γράφου.



Τα προβλήματα που αφορούσαν την εκτέλεση του SPARK επικεντρώνονταν κυρίως στην ύπαρξη του Καρτεσιανού γινομένου το οποίο επέβαλλε τη διερεύνηση μεγάλου αριθμού συνδυασμών. Όσο περισσότεροι συνδυασμοί έπρεπε να μελετηθούν τόσο μεγαλύτερες ήταν οι απαιτήσεις σε μνήμη αλλά και σε χρόνο καθώς αυξανόταν ο αριθμός των εκτελέσεων του BFS. Επιπλέον, το βάθος της αναζήτησης που επιλεγόταν και στους δύο αλγορίθμους είχε επιπτώσεις στην απόδοσή τους καθώς έπρεπε να γίνει επίσκεψη σε περισσότερους κόμβους αλλά και να υλοποιηθούν οι απαραίτητοι έλεγχοι. Ακόμα, σημαντικό ρόλο έπαιξε και ο μέγιστος αριθμός κόμβων που επιτρεπόταν να αντιστοιχηθούν σε κάθε κόμβο.

6.3. Παρουσίαση των αποτελεσμάτων

Η έξοδος όλων των αλγορίθμων που υλοποιήθηκαν έχει προσαρμοστεί με τέτοιο τρόπο ώστε να είναι διαθέσιμη σε μορφή γράφων συνοδευόμενων από το σύνολο των κόμβων που οι οποίοι αντιστοιχούνταν σε λέξεις-κλειδιά του ερωτήματος του χρήστη. Επειδή τα αποτελέσματα σκοπός είναι να χρησιμοποιηθούν σε μετέπειτα έρευνα για μελέτη παραγόντων όπως η εξατομίκευση και η διαφοροποίησή τους επιλέχθηκε να γίνει και καταγραφή των μονοπατιών που συνέδεαν τις λέξεις-κλειδιά. Ακολουθεί ως παράδειγμα το ερώτημα "Aristotle Ptolemy" εφαρμοσμένο σε υπογράφο του πλήρους γράφου της dbPedia αποτελούμενο από 786 κόμβους και 1062 ακμές. Σημειώνονται τα μονοπάτια που προκύπτουν στους διάφορους γράφους – απαντήσεων.

http://dbpedia.org/resource/Aristotle ---> http://dbpedia.org/resource/Ptolemy (1.0)
http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Ptolemy

http://dbpedia.org/resource/Category:Aristotle ---> http://dbpedia.org/resource/Ptolemy (2.0)
http://dbpedia.org/resource/Aristotle
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Aristotle

http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Ptolemy

http://dbpedia.org/resource/Category:Aristotle ---> http://dbpedia.org/resource/Category:Ptolemy (3.0)
http://dbpedia.org/resource/Aristotle
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Aristotle

http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Ptolemy

http://dbpedia.org/resource/Ptolemy
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Ptolemy

http://dbpedia.org/resource/Category:Ptolemy -> http://dbpedia.org/resource/Category:Commentators_on_Aristotle (4.0)
http://dbpedia.org/resource/Ptolemy
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Ptolemy

http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Ptolemy

http://dbpedia.org/resource/Aristotle
http://dbpedia.org/ontology/influenced
http://dbpedia.org/resource/Avicenna

http://dbpedia.org/resource/Avicenna
http://purl.org/dc/terms/subject
http://dbpedia.org/resource/Category:Commentators_on_Aristotle

6.4. Μελλοντική Εργασία

Το μεγαλύτερο όφελος από την παρούσα διπλωματική εργασία είναι ότι ερευνήθηκε σε βάθος το θέμα της αναζήτησης σε δεδομένα με γραφοειδή δομή και ο τρόπος με τον οποίο αυτή προσαρμόζεται σε περίπτωση που υπάρχουν σημασιολογικές σχέσεις. Ερευνήθηκαν πολλοί διαφορετικοί τρόποι αντιμετώπισης και συγκεντρώθηκε πολύτιμο υλικό που θα μπορούσε να αποτελέσει έναυσμα για περαιτέρω εργασίες και να γίνει ανταλλαγή ιδεών μεταξύ των αλγορίθμων έτσι ώστε να παραχθεί ένα ενδιαφέρον αποτέλεσμα.

6.4.1. Συνδυάζοντας τον EASE και τον SPARK

Ο χρόνος εκτέλεσης τους αλγορίθμου SPARK μπορεί να μειωθεί εάν εντάξουμε τον υπολογισμό των αναζητήσεων κατά πλάτος σε στάδιο ανάλογο με την προεπεξεργασία του EASE. Προτείνεται, δηλαδή, για κάθε κόμβο (u_1, u_2, …, u_m) που αντιστοιχεί σε εν δυνάμει λέξη-κλειδί w να κατασκευαστεί ένας υπογράφος του αρχικού ο οποίος προκύπτει από εκτέλεση του αλγορίθμου BFS. Με τον τρόπο αυτό, κάθε έλεγχος για την ύπαρξη υπογράφου που να περιέχει τους κόμβους ενός από τα query sets που δημιουργούνται κατά την εκτέλεση του SPARK να αποτελεί διαδικασία που να διεξάγεται σε σταθερό χρόνο αφού ανάγεται σε έναν απλό έλεγχο της μορφής "*Αυτοί η κόμβοι ανήκουν σε αυτό το γράφο;*".

6.4.2. Βελτιώνοντας τον BLINKS

Οι δομές εκείνες που χρησιμοποιεί ο BLINKS καταλαμβάνουν μεγάλο μέρος της διαθέσιμης μνήμης και δεδομένου ότι οι πληροφορίες που περιέχουν επικαλύπτονται τίθεται το ερώτημα κατά πόσο αυτές είναι χρήσιμες. Η ενσωμάτωση όλης της χρήσιμης πληροφορίας σε έναν πιο ευέλικτο πίνακα κατακερματισμού είναι εφικτή και επιτρέπει ταχύτερες πράξεις. Ουσιαστικά μπορεί να συμβάλλει και η διαπίστωση ότι, δεδομένης της μη κατευθυνόμενης μορφής που τελικά χρησιμοποιείται, ένας κόμβος του αρχικού γράφου μπορεί να είναι "ρίζα"

ενός γράφου απάντησης μόνο αν περιέχεται στις λίστες και των τριών λέξεων κλειδιά του χρήστη.

[ΒΙΒΛΙΟΓΡΑΦΙΑ]

[1] Qi Zhou, Ching Wang, Miao Xiong, Haofen Wang, Yong Yu. *SPARK: Adapting Keyword Query to Semantic Data*, 2007

[2] Hao He, Haixun Wang, Jun Yang, Philip S. Yu. *BLINKS: Ranked Keyword Searches on Graphs*. In SIGMOD 2007 .

[3] Guoliang Li, Beng Chin Ooi, Jianhua Feng, Jianyong Wang, Lizhu Zhou. EASE: *An Effective 3-in-1 Keyword Search Method for Unstructured, Semistructured and Structured Data*. In SIGMOD 2008 .

[4] G.Bhalotia, Arvind Hulgeri, Charuta Nakhe, Soumen Chakraarti. *Keyword Searching and Browsing in databases using BANKS. In ICDE 2002* .

[5] Thanh Tran, Haofen Wang, Sebastian Rudolf, Philip Cimiano. *Top-K Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF) Data*.

[6] S.Agrawal, S. Chaudhuri, G. Das. *DBXplorer: A system for keyword-based search over relational databases*. *IN ICDE 2002* .

[7] Grigoris Antoniou, Frank van Harmelen (2009). *Εισαγωγή στο Σημασιολογικό Ιστό*, Εκδόσεις Κλειδάριθμος

[8] Liyang Yu (2011). *A Developpers Guide to the Semantic Web,* Εκδόσεις Springer

[9] Tauberer, Joshua. (n.d.). Retrieved from http://www.rdfabout.com/intro/#Why we need a new standard for the Semantic Web

[10] Thomas H.Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein. (2006). *Εισαγωγή στους Αργορίθμους, Τόμος Ι.* Πανεπιστημιακές Εκδόσεις Κρήτης.

[11] Eric Hatcher, Otis Gospodnetic, Michael McCandless, *Lucene in Action*

[12] Rogers Cadenhead, L. L. (2007). *Πλήρες Εγχειρίδιο της Java 6.* Εκδόσεις Γκιούρδασς.

[13] Sedgewick, R. (2009). *Αλγόριθμοι σε C.* Εκδόσεις Κλειδάριθμος.