Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων
και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αποδοτική αναζήτηση με χρήση λέξεων-κλειδιών σε ημιδομημένα δεδομένα

Διδακτορική Διατριβή

της

Αγγελικής Δημητρίου

Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών Ε.Μ.Π.

Αθήνα, 18 Ιουλίου 2016

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων Και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αποδοτική αναζήτηση με χρήση λέξεων-κλειδιών σε ημιδομημένα δεδομένα

Διδακτορική Διατριβή

της

Αγγελικής Δημητρίου

Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών Ε.Μ.Π.

Συμβουλευτική Επιτροπή: Ι. Βασιλείου
Τ. Σελλής
Δ. Θεοδωράτος

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 18η Ιουλίου 2016.

Ι. Βασιλείου Καθ. ΕΜΠ	Τ. Σελλής Καθ. SWIN	Δ.Θεοδωράτος Αν. Καθ. NJIT
Α. Σταφυλοπάτης Καθ. ΕΜΠ	Θ. Δαλαμάγκας Ερευνητής Β΄ Ερ. Κέντρου ΑΘΗΝΑ	Ν. Κοζύρης Καθ. ΕΜΠ
	Κ. Κοντογιάννης Καθ. ΕΜΠ	

Αθήνα, 18 Ιουλίου 2016

...

Αγγελική Δημητρίου
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

ΠΡΟΛΟΓΟΣ

Η παρούσα διατριβή εκπληρώνει τις απαιτήσεις για την απόκτηση διπλώματος στο βαθμό του Διδάκτορα στη Σχολή Ηλεκτρολόγων και Μηχανικών Υπολογιστών στο Εθνικό Μετσόβειο Πολυτεχνείο (ΕΜΠ). Η εργασία που παρουσιάζεται περιγράφει μεθόδους για αποδοτική και αποτελεσματική αποτίμηση ερωτήσεων με λέξεις-κλειδιά σε ημιδομημένα δεδομένα και πραγματοποιήθηκε στο Εργαστήριο Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ.

Είμαι ευγνώμων στον Καθ. Ιωάννη Βασιλείου, που ήταν ο επιβλέπων αυτής της δουλειάς, αλλά και στον καθ. Τιμολέοντα Σελλή που με δέχτηκαν στο Εργαστήριο Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ και μου πρόσφεραν την καθοδήγηση και τη βοήθεια που χρειάστηκα για την ολοκήρωση αυτής της διατριβής. Ιδιαίτερα, όμως, ευχαριστώ τον καθ. Δημήτρη Θεοδωράτο, ο οποίος, με τις γνώσεις, την έμπνευση και την αδιάλειπτη συνεργασία που μου πρόσφερε απλόχερα, έπαιξε αποφασιστικό ρόλο στη σχηματοποίηση των ιδεών σε μια ολοκληρωμένη δουλειά, αποτέλεσμα της οποίας είναι η παρούσα διατριβή. Οφείλω, επίσης, ένα ευχαριστώ και στον Cem Aksoy και την Ananya Dass, για την πολύ όμορφη συνεργασία που είχαμε στα πλαίσιο των κοινών ερευνητικών ενδιαφερόντων μας.

Θα ήθελα, επίσης, να αναγνωρίσω τον κρίσιμο ρόλο που έπαιξε για μένα ο Δρ. Θοδωρής Δαλαμάγκας, κρατώντας ζωντανό το ενδιαφέρον μου στη μακρά διάρκεια της ενασχόλησής μου με τα θέματα διαχείρισης δεδομένων στο Εργαστήριο Βάσεων Γνώσεων και Δεδομένων. Τέλος, σημαντική ήταν για την πορεία της δουλειάς μου η συνεργασία και οι εμπειρίες που κέρδισα από τους συναδέλφους μου στο Κέντρο Δικτύων του ΕΜΠ αλλά και από τα μέλη, παλιά και νεότερα, του Εργαστηρίου Βάσεων Γνώσεων και Δεδομένων. Και φυσικά, όπως και για κάθε άνθρωπο, καμία πρόοδος στη ζωή μου δε θα γινόταν πραγματικότητα χωρίς την οικογένειά μου. Ευχαριστώ τους γονείς μου, τον Περικλή και τον αδερφό μου που με καθόρισαν και είναι πάντα δίπλα μου.

Αγγελική Δημητρίου
Αθήνα, 18 Ιουλίου 2016

Στον πατέρα μου

ΠΕΡΙΛΗΨΗ

Η αναζήτηση με χρήση λέξεων-κλειδιών είναι ο πλέον διαδεδομένος τρόπος αναζήτησης σε ημιδομημένα δεδομένα, συχνά άγνωστης συχνά δομής. Οι σύγχρονες μηχανές αναζήτησης δίνουν πρόσβαση σε μεγάλου όγκου δεδομένα που είναι ετερογενούς μορφής και διασκορπισμένα στο διαδίκτυο. Σε αντίθεση με τις δομημένες βάσεις δεδομένων και τις δομημένες γλώσσες ερωτήσεων που τις συνοδεύουν, σ΄ αυτήν την περίπτωση α) ο χρήστης δεν έχει την ανάγκη γνώσης της δομής της πληροφορίας και β) δε χρειάζεται να κατέχει εξειδίκευση σε μια γλώσσα ερωτήσεων. Τα πλεονεκτήματα αυτά συνοδεύονται από το μειονέκτημα της ασάφειας των ερωτήσεων. Το σύστημα αποτίμησης ερωτήσεων λέξεων-κλειδιών καλείται να αντιμετωπίσει αυτό το πρόβλημα, "μαντεύοντας" το νόημα της ερώτησης του χρήστη με βάση α) τις λέξεις-κλειδιά που περιέχονται στην ερώτησή του και β) τα δεδομένα πάνω στα οποία αποτιμάται η ερώτηση. Για το λόγο αυτό, η ποιότητα των αποτελεσμάτων διαφόρων προσεγγίσεων αναζήτησης είναι χαμηλή, όπως και η επίδοσή τους.

Σε αυτό το πλαίσιο, υπάρχουν τρία βασικά προβλήματα: α) η αποφυγή απώλειας χρήσιμων αποτελεσμάτων στην απάντηση της αναζήτησης, β) η ταξινόμηση των αποτελεσμάτων με βάση κάποιο αξιόπιστο κριτήριο και γ) η αποδοτική αποτίμηση ερωτήσεων λέξεων-κλειδιών. Το σχήμα της πληροφορίας που ακολουθεί τη μορφή δένδρου ή γράφου ορίζει σχέσεις μεταξύ των οντοτήτων πληροφορίας, οι οποίες πρέπει να λαμβάνονται υπόψιν αλλά με τρόπο που δε θα μειώνει την ποιότητα των αποτελεσμάτων αναζήτησης. Η επίδοση των αλγορίθμων που αποτιμούν ερωτήσεις με λέξεις-κλειδιά και έχουν τη δυνατότητα να ταξινομούν τα αποτελέσματα που προκύπτουν από μεγάλες συλλογές δεδομένων είναι καίριο ζήτημα. Για την αντιμετώπιση αυτού του προβλήματος, οι περισσότερες γνωστές προσεγγίσεις περιορίζουν εκ των πρωτέρων το σύνολο των αποτελεσμάτων σε ένα υποσύνολο αυτών, πληρώνοντας το τίμημα της απώλειας σωστών αποτελεσμάτων από την απάντηση που επιστρέφουν.

Στην παρούσα διατριβή, εισάγονται νέες μέθοδοι αναζήτησης σε ημιδομημένα δεδομένα. Παρουσιάζεται μια νέα σημασιολογία ταξινόμησης που βασίζεται στην έννοια του μεγέθους χαμηλότερου κοινού προγόνου. Σε αναλογία με την αναζήτηση με κριτήρια εγγύτητας στο πεδίο της Ανάκτησης Πληροφορίας (IR), το μέγεθος χαμηλότερου κοινού προγόνου αντικατοπτρίζει την εγγύτητα εμφάνισης των λέξεων-κλειδιών σε ένα δένδρο δεδομένων. Αυτή η προσέγγιση δεν απορρίπτει εκ προοιμίου κανένα αποτέλεσμα, αλλά μέσω μιας κλιμακωτής ταξινόμησης επιδεικνύει βελτιωμένη αποτελεσματικότητα στην αποτίμηση ερωτήσεων με λέξεις-κλειδιά, σε σύγκριση με τις έως τώρα προσεγγίσεις.

Για την αντιμετώπιση του προβλήματος επίδοσης, σχεδιάσαμε μια οικογένεια αλγορίθμων που χρησιμοποιούν στοίβες. Οι αλγόριθμοι αξιοποιούν το δικτυωτό των διαμερίσεων των λέξεων-κλειδιών μιας ερώτησης, για να επιστρέψουν τα αποτελέσματα υπολογίζοντας ταυτόχρονα τα μεγέθη χαμηλότερων κοινών προγόνων. Το δικτυωτό

αυτό εξασφαλίζει γραμμική απόκριση των αλγορίθμων σε σχέση με το μέγεθος εισόδου, για δεδομένο αριθμό από λέξεις-κλειδιά. Ως αποτέλεσμα, οι αλγόριθμοί μας εκτελούνται αποδοτικά σε μεγάλα σύνολα δεδομένων και για ερωτήσεις με πολλές λέξεις-κλειδιά. Επεκτείνοντας τη λογική αυτή, προχωρήσαμε στο σχεδιασμό νέων top-k αλγορίθμων, που υλοποιούν μια διαφορετική λογική επιλογής κορυφαίων k αποτελεσμάτων. Η εκτεταμένη πειραματική μας ανάλυση επιβεβαιώνει τα θεωρητικώς προδοκόμενα. Σε αντίθεση με ανάλογες προσεγγίσεις οι αλγόριθμοί μας κλιμακώνονται ομαλά καθώς ο αριθμός των λέξεων-κλειδιών και των εμφανίσεών τους σε διάφορα σύνολα δεδομένων αυξάνονται.

Παρουσιάζεται, επίσης, μια πολυεπίπεδη μεθοδολογία συσταδοποίησης, που ομαδοποιεί αποτελέσματα παρόμοιας δομής και σημασιολογίας, επιτρέποντας στο χρήστη να επικεντρώνεται σε μικρό αριθμό αποτελεσμάτων. Η συσταδοποίηση αποφασίζεται από ένα σύνολο ομομορφιδμών που ορίζονται μεταξύ δενδρικών προτύπων. Οι συστάδες ορίζονται σε διαφορετικό επίπεδο λεπτεμέρειας και παρουσιάζονται εμφωλευμένες από τις γενικότερες στις ειδικότερες, καθοδηγώντας το χρήστη γρήγορα στα επιθυμητά αποτελέσματα. Για τη μέθοδο αυτή, επίσης σχεδιάστηκε ένας αποδοτικός αλγόριθμος που αποκρίνεται σε πρακτικό χρόνο, όπως επιβεβαιώνεται από την πειραματική μας μελέτη. Τα πειράματα, επίσης, κατέδειξαν ότι η μεθοδολογία συσταδοποίησης ουσιαστικά βοηθά τους χρήστες να εντοπίσουν τα επιθυμητά αποτελέσματα ξεπερνώντας σε αποτελεσματικότητα ανάλογες προσεγγίσεις.

Παρόλο που οι ερωτήσεις με λέξεις-κλειδιά είναι απλές και διευκολύνουν το χρήστη, οι υπάρχουσες σημασιολογίες όσο καλά αποτελέσματα και να επιδεικνύουν κατά περίπτωση, δεν μπορούν επ ουδενί να ¨μαντέψουν' την πρόθεση αναζήτησης του χρήστη. Το αποτέλεσμα είναι χαμηλής ποιότητας αποτελέσματα αναζήτησης. Προς αυτήν την κατεύθυνση, εισαγάγμε μια νέα γλώσσα ερωτήσεων με συνεκτικές λέξεις-κλειδιά. Διαισθητικά, μια σχέση συνεκτικότητας ανάμεσα σε λέξεις-κλειδιά υποδεικνύει πως σε ένα αποτέλεσμα οι συγκεκριμένες λέξεις-κλειδιά θα πρέπει να σχηματίζουν ένα συνεκτικό, αδιαίρετο σύνολο. Στη γλώσσα αυτή επιτρέπεται η επανάληψη λέξεων-κλειδιών και η εμφώλευση των συνεκτικών όρων. Η μορφή αυτή ερωτήσεων γεφυρώνει το χάσμα ανάμεσα στις απλές ερωτήσεις με λέξεις-κλειδιά και τις εξελιγμένες, αυτστηρές δομημένες γλώσσες. Παρόλο που είναι πιο εκφραστικές οι ερωτήσεις με συνεκτικές σχέσεις, είναι όσο απλές στη διατύπωση είναι κι οι ερωτήσεις χωρίς συνεκτικές σχέσεις, ενώ επίσης δεν απαιτούν γνώση της δομής του συνόλου δεδομένων. Για την αποτίμηση των ερωτήσεων, σχεδιάσαμε κατάλληλο αλγόριθμο στη λογική των αλγορίθμων που αξιοποιούν το δικτυωτό των διαμερίσεων των λέξεων-κλειδιών. Η πειραματική μας μελέτη αποδεικνύει την υπεροχή της μεθόδου μας σε σχέση με παλαιότερες σημασιολογίες φιλτραρίσματος και προβάλλει την επίδοση του αλγορίθμου μας, δείχνοντας ότι έχει τη δυνατότητα αποτίμησης ακόμα και ερωτήσεων με πολύ μεγάλο αριθμό λέξεων-κλειδιών.

Λέξεις-κλειδιά: αναζήτηση με λέξεις-κλειδιά, γλώσσα ερωτήσεων, χαμηλότερος κοινός πρόγονος, διαχείριση δεδομένων, ημιδομημένα δεδομένα, συσταδοποίηση, δενδρικά πρότυπα, στοίβα

ABSTRACT

Keyword search is the most popular querying technique on large semistructured datasets, often of unknown structure, in the web. Keyword queries are simple and convenient. However, as a consequence of their imprecision, there is usually a huge number of candidate results of which only very few match the user's intent. Unfortunately, the existing semantics for keyword queries are ad-hoc and they generally fail to "guess" the user intent. Therefore, the quality of their answers is poor and the existing algorithms do not scale satisfactorily.

In this context, three challenging problems are (a) to avoid missing useful results in the answer set, (b) to rank the results with respect to some relevance criterion and (c) to design algorithms that can efficiently compute the results on large datasets. A major challenge of a ranking approach is the efficiency of its algorithms as the number of keywords and the size and complexity of the data increase. To face this challenge most of the known approaches restrict their ranking to a subset of the LCAs (e.g., SLCAs, ELCAs), missing relevant results.

In this thesis, we present a novel ranking semantics for keyword queries which is based on the concept of LCA size. Similarly to metric selection in Information Retrieval, LCA size reflects the proximity of keyword matches in the data tree. This semantics does not rank a predefined subset of LCAs and through a layered presentation of results, it demonstrates improved effectiveness compared to previous relevant approaches.

To address performance challenges, we design novel stack-based algorithms, which exploit a lattice of the partitions of the keyword set to return, as an answer to a keyword query, all the results ranked on their size. This feature empowers a linear time performance on the size of the input data for a given number of query keywords. As a result, our algorithm can run efficiently on large input data for several keywords. We extend our approach to efficiently support top-k keyword query answering, with new top-k-size semantics. An extensive experimental study on various and large datasets confirms the theoretical analysis. The results show that, in contrast to other approaches, our algorithms scale smoothly when the size of the dataset and the number of keywords increase.

Furthermore, we present a multi-level clustering methodology which groups together results with similar structural and semantic features, and allows the user to focus on a small subset of the results. In order to define our cluster hierarchy, we exploit homomorphisms between label paths which allow the detection of commonalities between patterns of results. An originality of our approach is that the clusters are ranked at different levels of granularity to quickly guide the user to the relevant result patterns. We design an efficient stack-based algorithm for generating result patterns and constructing the clustering hierarchy. Our extensive experimentation

with multiple real datasets shows that our algorithm is fast and scalable. It is also shown that our clustering methodology allows the users to effectively retrieve their intended results, and outperforms a recent state-of-the-art clustering approach.

Although keyword queries are simple and convenient, the existing semantics for keyword queries are prone to failure on "guessing" the user intent. Therefore, the quality of the answers is often poor. To address this issue, we introduce the novel concept of cohesive keyword queries for tree data. Intuitively, a cohesiveness relationship on keywords indicates that they should form a cohesive whole in a query result. Cohesive keyword queries allow term nesting and keyword repetition. They bridge the gap between flat keyword queries and structured queries. Although more expressive, they are as simple as flat keyword queries and they do not require any schema knowledge. We provide formal semantics for cohesive keyword queries and rank query results on the proximity of the keyword instances. We design a stack based algorithm which efficiently evaluates cohesive keyword queries. Our experiments demonstrate that our approach outperforms in quality previous filtering semantics and our algorithm scales smoothly on queries of even 20 keywords on large datasets.

Keywords: keyword search, data management, ranking semantics, stack-based algorithm, hierarchical clustering, tree pattern, keyword query language

Κεφάλαιο 1

Εισαγωγή

Η δενδρική δομή οργάνωσης δεδομένων είναι ευρέως διαδεδομένη μέσω προτύπων όπως είναι, για παράδειγμα, τα μοντέλα δεδομένων XML και JSON. Αξιοποιείται για την αναπαράσταση κι εύκολη ανταλλαγή δεδομένων στο διαδίκτυο, που δεν ακολουθούν αυστηρή δομή και που αλλάζουν συνεχώς. Παραδείγματα περιοχών που αξιοποιούν δενδρική μορφή αναπαράστασης των δεδομένων τους είναι το πεδίο της βιοπληροφορικής (bioinformatics) [33], η εξόρυξη δεδομένων στο διαδίκτυο (web mining) [53] και η ολοκλήρωση δεδομένων (data integration) και δημοσιοποίηση στο διαδίκτυο (web publishing) [20].

Την τελευταία δεκαετία έχει απασχολήσει αρκετά την ερευνητική κοινότητα η δυνατότητα αναζήτησης σε τέτοιου είδους δεδομένα, με τρόπο εύκολο κι ανάλογο της χαλαρότητας του σχήματός τους. Στην εποχή της άνθισης των μηχανών αναζήτησης με ερωτήσεις λέξεων - κλειδιών δημιουργείται αυτονόητα η ανάγκη για αξιοποίηση αυτού του μοντέλου αναζήτησης και σε δεδομένα δενδρικής δομής, όπως έχει γίνει εκτενώς στο κλασικό πεδίο της ανάκτησης πληροφορίας (Information Retrieval). Η ευκολία που παρέχεται στο χρήστη, ο οποίος δε χρειάζεται να γνωρίζει το σχήμα της πληροφορίας που αναζητά και δεν πρέπει να διατυπώσει την ερώτησή του ακολουθώντας αυστηρή σύνταξη, έχει στον αντίποδά της τη δυσκολία που δημιουργείται στα συστήματα αποτίμησης, στο να επιστρέψουν αποδοτικά, αποτελέσματα τα οποία ικανοποιούν την πρόθεση αναζήτησης του χρήστη.

1.1 Προβλήματα και προκλήσεις

Η εγγενής ασάφεια των ερωτήσεων λέξεων - κλειδιών αντιμετωπίζεται από τη βιβλιογραφία με τον ορισμό διάφορων μοντέλων αποτίμησης, ταξινόμησης και συσταδοποίησης των απαντήσεων. Ο κοινός στόχος είναι η διευκόλυνση του χρήστη να εντοπίσει την ενδιαφέρουσα γι΄ αυτόν πληροφορία μέσα σε ένα σύνολο πολυάριθμων αποτελεσμάτων, καθώς μόνο μικρός αριθμός αυτών απαντά συνήθως ικανοποιητικά την ερώτηση του χρήστη.

Η ιεραρχική δομή οργάνωσης των δεδομένων προσδίδει σημασιολογία στην αναπαριστώμενη πληροφορία, αλλά ταυτόχρονα περιπλέκει τη διαδικασία αποτίμησης. Κάθε εμφάνιση ενός όρου μέσα στη δομή δεδομένων σχετίζεται, εν δυνάμει, με κάθε άλλο όρο του δένδρου. Κατά συνέπεια, ο αριθμός των πιθανών αποτελεσμάτων μιας ερώτησης με n λέξεις - κλειδιά είναι εκθετικός στο n. Το γεγονός αυτό δημιουργεί την ανάγκη ορισμού κατάλληλων δομών ευρετηρίου αλλά και το σχεδιασμό αποδοτικών αλγορίθμων, που θα έχουν τη δυνατότητα να επεξεργάζονται σε πρακτικό χρόνο τέτοιες

ερωτήσεις.

Την ίδια στιγμή, ένας μεγάλος αριθμός αποτελεσμάτων είναι απαγορευτικός για ένα χρήστη ο οποίος έχει υποβάλει μια ερώτηση με στόχο να λάβει μία ή έστω λίγες σχετικές απαντήσεις. Το σύστημα αποτίμησης θα πρέπει α) είτε να φιλτράρει τα αποτελέσματα επιστρέφοντας τελικά ένα υποσύνολο αυτών, είτε β) να τα ταξινομεί δίνοντας ταυτόχρονα τη δυνατότητα για γρήγορη επιστροφή των κορυφαίων k αποτελεσμάτων, είτε γ) να τα ομαδοποιεί και να τα κατηγοριοποιεί, υποβοηθώντας έτσι το χρήστη να εντοπίσει την πληροφορία που πραγματικά αναζητά.

1.2 Συνεισφορά

Εν συντομία, η συνεισφορά της διατριβής περιλαμβάνει τα παρακάτω:

- Εισαγωγή νέας σημασιολογίας μεγέθους χαμηλότερου κοινού προγόνου (lowest common ancestor size - LCA size) για ερωτήσεις λέξεων-κλειδιών σε δενδρικά δεδομένα.

- Σχεδιασμό νέου αλγορίθμου για αποτίμηση ερωτήσεων λέξεων-κλειδιών, με αποτελέσματα ταξινομημένα ως προς το μέγεθος χαμηλότερου κοινού προγόνου. Ο αλγόριθμος αξιοποιεί ένα δικτυωτό (lattice) των διαμερίσεων του συνόλου των λέξεων-κλειδιών, με αποτέλεσμα τη γραμμική συμπεριφορά της επίδοσής του σε σχέση με την είσοδο.

- Εισαγωγή νέας ποσέγγισης επιστροφής κορυφαίων k αποτελεσμάτων και σχεδιασμός αλγορίθμων που πραγματοποιούν περικοπή αποτελεσμάτων νωρίς στον υπολογισμό, επιταχύνοντας την αποτίμηση. Η προσέγγιση κορυφαίων μεγεθών αποδεικνύεται ότι υπερέχει σε σχέση με τις επικρατέστερες σημασιολογίες φιλτραρίσματος.

- Εξαγωγή εκφραστικών προτύπων από τα αποτελέσματα ερωτήσεων και ορισμός ομομορφισμών μεταξύ τους, με στόχο την ταξινόμησή τους ως προς τη σχετικότητά τους με την ερώτηση του χρήστη.

- Πολυεπίπεδη συσταδοποίηση δενδρικών προτύπων απαντήσεων μέσω της οποίας ο χρήστης καταλήγει εύστοχα και γρήγορα στα αποτελέσματα του ενδιαφέροντός του.

- Σχεδιασμό αποδοτικών αλγορίθμων για υπολογισμό αποτελεσμάτων με μεγάλη δομική και σημασιολογική λεπτομέρεια, σε πρακτικούς χρόνους.

- Εισαγωγή γλώσσας ερωτήσεων συνεκτικών λέξεων - κλειδιών (cohesive keyword queries) σε δένδρα, η οποία διατηρεί την απλότητα των ερωτήσεων με λέξεις-κλειδιά, αλλά αυξάνει με διαφορά την εκφραστικότητά τους, επιτυγχάνοντας υψηλούς βαθμούς αποτελεσματικότητας.

- Σχεδιασμό αλγορίθμου αποτίμησης ερωτήσεων συνεκτικών λέξεων - κλειδιών, ο οποίος είναι σε θέση να αποκρίνεται γρήγορα για ερωτήσεις με πολύ μεγάλο αριθμό λέξεων-κλειδιών και σε μεγάλα σύνολα δεδομένων.

1.3 Μελλοντική εργασία

Τα επόμενα βήματα, όπως αναφέρεται στο Κεφάλαιο 6, επικεντρώνονται στη λεπτομερή ταξινόμηση των αποτελεσμάτων ερωτήσεων λέξεων - κλειδιών σε δενδρικά δεδομένα, αξιοποιώντας τη στατιστική και δομική συσχέτιση των λέξεων - κλειδιών μέσα στη δομή δεδομένων προς αναζήτηση. Το μοντέλο συσχέτισης των λέξεων - κλειδιών θα επεκταθεί και σε δενδρικά πρότυπα απαντήσεων με στόχο τη διαφοροποίηση μέσω συσταδοποίησης των διαφορετικών τύπων αποτελεσμάτων. Επιπλέον, ήδη διερευνάται πώς οι μέθοδοι, που έχουν αναπτυχθεί στο πλαίσιο αυτής της διατριβής, μπορούν να βοηθήσουν σε ανάλογα προβλήματα που απαντώνται σε δεδομένα που οργάνωσης γράφου. Κάποια από τ΄ αποτελέσματα αυτά έχουν ήδη δημοσιευθεί σε διεθνή συνέδρια. Τέλος, μελετώνται ήδη τρόποι αξιοποίησης τεχνικών παράλληλης επεξεργασίας στην οικογένεια αλγορίθμων που προτάθηκαν και χρησιμοποιούν δικτυωτά των λέξεων-κλειδιών για την αποτίμηση ερωτήσεων, ώστε αποτιμούν πολύ μεγάλα σύναλα δεδομένων αποδοτικά.

1.4 Δομή της έκθεσης

Το υπόλοιπο υλικό της έκθεσης οργανώνεται ως ακολούθως: στο Κεφάλαιο 2 γίνεται μια εισαγωγή στις βασικές έννοιες και προβλήματα του πεδίου που κινείται η θεματολογία αυτής της διατριβής και παρουσιάζεται η σχετική βιβλιογραφία στο χώρο της αποτίμησης ερωτήσεων λέξεων - κλειδιών σε δενδρικά δεδομένα. Στο Κεφάλαιο 3 εισάγεται μια νέα σημασιολογία στην αποτίμηση ερωτήσεων λέξεων - κλειδιών σε δενδρικά δεδομένα και παρουσιάζονται νέοι αλγόριθμοι για την αποδοτική αποτίμηση ερωτήσεων, αξιοποιώντας τη σημασιολογία αυτή. Στο Κεφάλαιο 4 παρουσιάζεται μια νέα μέθοδος ταξινόμησης των απαντήσεων μέσω δεδνδρικών προτύπων, τα οποία συσταδοποιούνται με βάση ομομορφισμούς που ορίζονται μεταξύ τους. Στο Κεφάλαιο 5 ορίζεται μια νέα γλώσσα λέξεων - κλειδιών που ενδυναμώνει την εκφραστικότητα της μέχρι τώρα απλής διατύπωσης ερωτήσεων, χωρίς ωστόσο να δυσκολεύει το χρήστη, ενώ ταυτόχρονα επιδεικνύει αυξημένη ποιότητα αποτελεσμάτων τόσο σε δέντρα, όσο και σε γράφους. Τέλος, στο Κεφάλαιο 6 συνοψίζουμε την παρουσίαση της εργασίας και καταγράφουμε τα ερευνητικά μονοπάτια που έχουν ανοίξει από τη δουλειά αυτής της διατριβής και αποτελούν τα επόμενα βήματα της μελέτης μας.

Κεφάλαιο 2

Αναζήτηση με λέξεις-κλειδιά σε ημιδομημένα δεδομένα

Η έκρηξη ανάπτυξης του διαδικτύου τις τελευταίες δεκαετίες έφερε στο φως τερράστιες ποσότητες πληροφορίας που είναι προσβάσιμη από καθέναν. Η διαδικασία επιτυχούς εύρεσης της πληροφορίας ενδιαφέροντος από ένα χρήστη συνίσταται στη δυνατότητα που του δίνεται από τα πεοσφερόμενα συστήματα αναζήτησης, να εντοπίσει εύστοχα την ακριβή πληροφορία που αναζητά και ταυτόχρονα να το επιτύχει σε μικρό χρόνο. Αυτό είναι και το αίτημα που όλα τα συστήματα διαχείρισης βάσεων δεδομένων καλύπτουν με επιτυχία εδώ και πολλά χρόνια. Τί συμβαίνει όμως στην περίπτωση χαώδους, αδόμητης πληροφορίας σκορμπισμένης σε διαφορετικά σημεία του διαδικτύου και τελικών χρηστών που αδυνατούν αν έχουν πρόσβαση σε οργανωμένα συστήματα διαχείρισης βάσεων δεδομένων;

Η λύση στην περίπτωση αυτή προσφέρεται από τις μηχανές αναζήτησης, πολλές από τις οποίες χρησιμοποιούμε όλοι καθημερινά. Η επιτυχία τους εντοπίζεται σε δύο βασικά χαρακτηριστικά: α) δεν υπάρχει καμία απαίτηση εξειδικευμένης γνώσης από το χρήστη για τη διατύπωση μιας ερώτσης αναζήτησης και β) δεν υπάρχει καμία απαίτηση συγκεκριμένης οργάνωσης της πληροφορίας από τις πηγές που την προσφέρουν.

Συγκεκριμένα, οι χρήστες διατυπώνουν τα ερωτήματα αναζήτησης χρησιμοποιώντας μόνο λέξεις-κλειδιά. Δε χρειάζεται να έχουν γνώση ούτε κάποιας αυστηρής, δομημένης γλώσσας, όπως π.χ. SQL, XQuery, SPARQL, ούτε της οργάνωσης της πληροφορίας στις πηγές όπου απευθύνουν την αναζήτησή τους. Δε χρειάζεται να γνωρίζουν την πηγή της πληροφορίας, ενώ μπορούν ταυτόχρονα να ψάχνουν σε πολλές διαφορετικές πηγές, οι οποίες συχνά αλλάζουν σε επίπεδο περιεχομένου και δομής δυναμικά και μπορούν να ακολουθούν τελείως διαφορετικό μοντέλο αναπαράστασης πληροφορίας.

Η ελευθερία αυτή των χρηστών εισάγει, ωστόσο, δυσκολία στα συστήματα αναζήτησης να απαντήσουν με υψηλή ακρίβεια (precision) και ταυτόχρονα υψηλή πληρότητα (recall). Η απλή χρήση λέξεων-κλειδιών χωρίς άλλα συντακτικά χαρακτηριστικά μειώνει στο ελάχιστο την εκφραστικότητα των ερωτήσεων. Ο εντοπισμός των εμφανίσεων των λέξεων-κλειδιών της ερώτησης του χρήστη σε μια πηγή είναι αναγκαστικά το μόνο στοιχείο για την εύρεση της πληροφορίας ενδιαφέροντος. Ο τερράστιος αριθμός των εμφανίσεών τους, όμως, και οι αμέτρητες δυνατότητες συνδυασμού και ομαδοποίησής τους καταλήγουν σε μεγάλο αριθμό πιθανών αποτελεσμέτων, που αφενός ο χρήστης δεν μπορεί στην πράξη να ελέγξει, και αφετέρου είναι πιθανό να μην αντικατοπτρίζουν το σκοπό αναζήτησης του χρήστη. Ταυτόχρονα, ο υπολογισμός όλων αυτών των πι-

θανών αποτελεσμάτων είναι ένα απαιτητικό υπολογιστικά πρόβλημα που δεν μπορεί να λυθεί με εξαντλητικό τρόπο.

Για την αντιμετώπιση των παραπάνω προβλημάτων έχουν επικρατήσει διάφορες προσεγγίσεις, οι οποίες στοχεύουν στο να περιορίσουν την έκταση της ασάφειας των ερωτήσεων λέξεων-κλειδιών με δύο δυνατούς τρόπους: α) αναλύοντας τα στατιστικά των όρων που εμφανίζονται στις διάφορες πηγές πληροφορίας, ή β) δίνοντας επιπλέον δυνατότητες διατύπωσης ή αναλύοντας την ερώτηση του χρήστη. Για την πρώτη περίπτωση, αξιοποιούνται στατιστικά εμφάνισης των λέξεων-κλειδιών μέσα στις πηγές και συσχετίσεις μεταξύ τους. Ο σκοπός είναι, τα αποτελέσματα είτε να περιοριστούν, είτε να ομαδοποιηθούν, είτε να ταξινομηθούν, ώστε ο χρήστης να βρει στα κορυφαία αποτελέσματα, δηλ. σε πρακτικό χρονο, αυτό που τον ενδιαφέρει. Για τη δεύτερη περίπτωση, στη διατύπωση της ερώτησης αναζήτησης γίνονται δεκτά κάποια περιορισμένα συντακτικά στοιχεία, π.χ., σύμβολα, ετικέτες κτλ, που ένας χρήστης χωρίς εξειδίκευση μπορεί διαισθητικά να χρησιμοποιήσει. Προς την κατεύθυνση αυτή, πολλές φορές γίνεται επίσης ανάλυση του προφίλ του χρήστη με βάση το ιστορικό του ή λαμβάνονται υπόψη χαρακτηριστικά της ερώτησης όπως η σειρά τοποθέτησης των λέξεων-κλειδιών.

Στο πεδίο της ανάκτησης πληροφορίας (Information Retrieval, IR) όπου το πρόβλημα αναζήτησης με λέξεις-κλειδιά έχει μελετηθεί εκτενώς, η αναζήτηση διενεργείται σε συλλογές εγγράφων, καθένα από τα οποία αποτελεί πιθανό αποτέλεσμα σε μια αναζήτηση. Τα έγγραφα αυτά περιέχουν μερικώς ή συνολικά τις λέξεις της ερώτησης του χρήστη, και το σύστημα αποφασίζει αν το συγκεκριμένο έγγραφο πρέπει ή όχι να επιστραφεί στο σύνολο των αποτελεσμάτων, σε ποια σειρά θα καταχθεί κ.τ.λ.. Η εύρεση των έγκυρων εγγράφων βασίζεται στην προεπεξεργασία της συλλογής των εγγράφων που διατίθενται. Μέσω στατιστικής ανάλυσης των εμφανίσεων όλων των όρων, που περιέχονται στη συλλογή, επιλέγονται οι χαρακτηριστικοί όροι (διαδικασία *feature extraction*). Οι όροι αυτοί αποτελούν τη βάση για την αναπαράσταση κάθε εγγράφου, αλλά και των ερωτήσεων στο εκάστοτε υιοθετημένο μοντέλο αναπαράστασης. Το μοντέλο μπορεί να είναι το λογικό (boolean model), το διανυσματικό (vector model), το πιθανοτικό (probalistic model), το γλωσσικό (language model) ή παράγωγά τους. Η ομοιότητα της αναπαράστασης ενός εγγράφου με την αναπαράσταση μιας δεδομένης ερώτησης καθορίζει το σύνολο και τη σειρά παρουσίασης των αποτελεσμάτων μιας ερώτησης με λέξεις-κλειδιά.

Όταν η οργάνωση της πληροφορίας ακολουθεί ημιδομημένη μορφή, δηλ. μορφή δένδρου ή γράφου, ο προσδιορισμός των αποτελεσμάτων περιπλέκεται, εξαιτίας της μη σαφούς οριοθέτησης της μονάδας ενός αποτελέσματος, αλλά και εξαιτίας των πολλαπλών συνδέσεων των κόμβων της πληροφορίας μεταξύ τους. Στις περιπτώσεις αυτές, αντίθετα με την επιστροφή ενός ολόκληρου εγγράφου, το αποτέλεσμα ορίζεται στο πλαίσιο ενός υποδέντρου ή υπογράφου, τα όρια του οποίου πρέπει επίσης να υπολογιστούν από το σύστημα αποτίμησης. Η διαδικασία αυτή δυσχεραίνεται από το γεγονός ότι σε μια συλλογή τέτοιας μορφής, όλες οι εμφανίσεις των λέξεων-κλειδιών μιας ερώτησης δυνητικά συνδέονται πλήρως μεταξύ τους, οδηγώντας σε εκθετικό αριθμό αποτελεσμάτων σε σχέση με τον αριθμό των λέξεων-κλειδιών, και μάλιστα με επικαλύψεις. Συνεπώς, σε αυτό το μοντέλο υπεισέρχεται και ο προσδιορισμός του βαθμού λεπτομέρειας του κάθε αποτελέσματος, ώστε να μην περιορίζεται το ολοκληρωμένο νόημα μιας απάντησης για το χρήστη, αλλά ταυτόχρονα να καλύπτεται συνολικά ο σκοπός της αναζήτησής του από κάθε αποτέλεσμα.

Πληροφορία που αναπαρίσταται με βάση τα πρότυπα XML, JSON, RDF είναι εγγενώς ημιδομημένη. Υποννοείται σχήμα για τα δεδομένα, το οποίο ωστόσο δεν είναι

αυστηρό, οπότε δεν ακολουθείται με συνέπεια από τα στιγμιότυπα της πληροφορίας, ενώ πιθανώς το σχήμα δεν είναι διαθέσιμο. Ο διαμοιρασμός τέτοιου είδους πληροφορίας είναι πολύ διαδεδομένος σε πλήθος εφαρμογών στο διαδίκτυο, σε πηγές πληροφορίας, σε ανταλλαγή δεδομένων και σε απόδοση σημασιολογίας σε δεδομένα. Η παρούσα διατριβή επικεντρώνεται στην αποδοτική αναζήτηση με λέξεις-κλειδιά σε αυτόν τον τύπο οργάνωσης δεδομένων.

2.1 Σημασιολογία χαμηλότερου κοινού προγόνου (LCA)

σε δενδρικά δεδομένα

Πριν αναφερθούν οι λεπτομέρειες κι οι διαφορές των επικρατέστερων προσεγγίσεων αντιμετώπισης του προβλήματος της ασάφειας των ερωτήσεων λέξεων - κλειδιών, πρέπει να αναδειχθεί ο κοινός τους τόπος. Στην αναζήτηση με λέξεις - κλειδιά σε επίπεδο κείμενο, τμήματα κειμένου που περιέχουν όλους τους όρους της αναζήτησης αποτελούν τα υποψήφια αποτελέσματα. Στην περίπτωση δενδρικών δομών, τα αποτελέσματα ορίζονται ως υποδέντρα της αρχικής δομής της πληροφορίας. Τα υποδέντρα αυτά πρέπει να περιέχουν όλες τις λέξεις - κλειδιά της ερώτησης. Έχει επικρατήσει να θεωρούνται ως υποδέντρα αναφοράς για την απάντηση σε μια ερώτηση, τα ελάχιστα συνδετικά δέντρα (minimum connecting trees - MCT) ενός συνδυασμού εμφανίσεων των λέξεων - κλειδιών της ερώτησης στο δέντρο δεδομένων. Σε κάθε συνδυασμό, απαντάται τουλάχιστον ένα στιγμιότυπο για κάθε λέξη - κλειδί της ερώτησης. Η ρίζα αυτού του δέντρου είναι ο *χαμηλότερος κοινός πρόγονος* (LCA) των συγκεκριμένων στιγμιοτύπων των εν λόγω λέξεων - κλειδιών.

Οι χαμηλότεροι κοινοί πρόγονοι των συνδυασμών των στιγμιοτύπων των λέξεων - κλειδιών μιας ερώτησης οριοθετούν, κατά συνέπεια, τα υποδέντρα που είναι επιλέξιμα για απάντηση σε ερωτήσεις με λέξεις-κλειδιά. Έτσι, η κυρίαρχη μερίδα της βιβλιογραφίας ανάγει την εύρεση απαντήσεων στον εντοπισμό των χαμηλότερων κοινών προγόνων των στιγμιοτύπων αυτών.

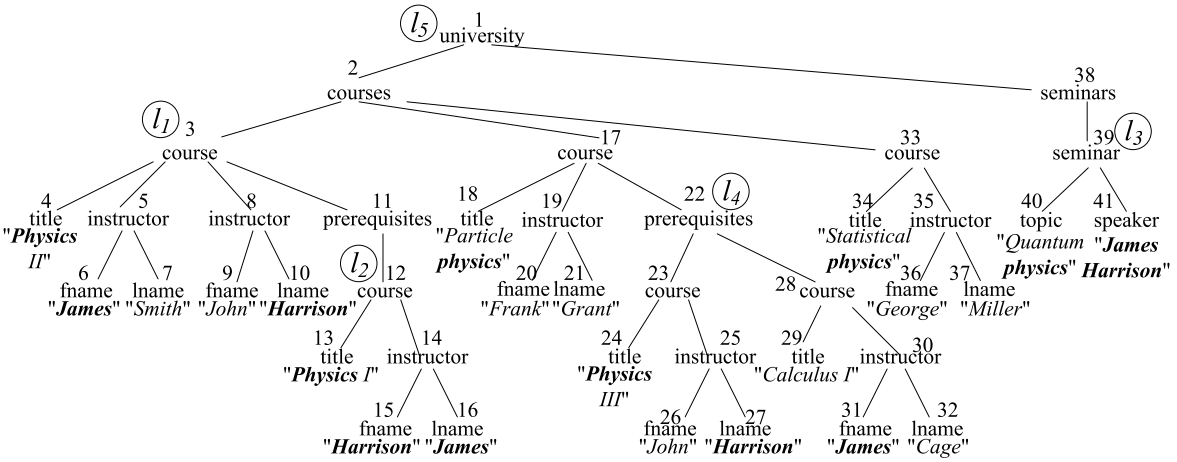
Ο εξαντλητικός υπολογισμός όλων των δυνατών συνδυασμών των στιγμιοτύπων των κλειδιών της ερώτησης, για τον εντοπισμό όλων των χαμηλότερων κοινών προγόνων, δεν είναι υπολογίσιμος σε πρακτικό χρόνο για μεγάλα σύνολα δεδομένων και λέξεις με πολλές εμφανίσεις. Τόσο για ποσοτικούς λόγους, όσο και για ποιοτικούς, χρειάζεται να περιοριστεί ο αριθμός των αποτελεσμάτων, που θα επιστραφεί τελικά στο χρήστη. Στη βιβλιογραφία, αυτό επιτυγχάνεται με την υιοθέτηση κατάλληλης σημασιολογίας από κάθε προτεινόμενη προσέγγιση. Κατατάσσουμε τις σχετικές εργασίες σε δύο βασικές κατηγορίες: α) αυτές που ακολουθούν *λογική ταξινόμησης* και β) αυτές που ακολουθούν την *τακτική φιλτραρίσματος* των αποτελεσμάτων. Στην πρώτη περίπτωση, ο στόχος είναι η αναζήτηση του χρήστη να καλυφθεί από τα κορυφαία αποτελέσματα, ώστε να μη χρειαστεί να γίνει εξέταση όλων των αποτελεσμάτων, ενώ στη δεύτερη η εξ αρχής περικοπή του συνόλου των αποτελεσμάτων με ευρηστικά κριτήρια.

Οι δύο βασικές οικογένειες των προσεγγίσεων δεν είναι σαφώς διαχωρισμένες. Συνήθως, η υιοθέτηση μιας σημασιολογίας για περικοπή του συνόλου των αποτελεσμάτων είναι η βασική αρχή. Σε δεύτερο στάδιο, από ορισμένες προσεγγίσεις, εφαρμόζεται και κάποια λογική ταξινόμησης. Παρόλο που τα κριτήρια φιλτραρίσματος για περικοπή του συνόλου αποτελεσμάτων, που έχουν κατά καιρούς προταθεί, σε πολλές περιπτώσεις μοιάζουν διαισθητικά σωστά, η στρατηγική αυτή έχει ένα εγγενές πρόβλημα: τα

μειονεκτήματα της σημασιολογίας φιλτραρίσματος κληρονομούνται αναγκαστικά στο στάδιο της ταξινόμησης. Συνεπώς, η ακρίβεια (precision) και η πληρότητα (recall) του συνόλου των αποτελεσμάτων θησιάζονται, όπως έχει δειχθεί [49].

2.1.1 Φιλτράρισμα χαμηλότερων κοινών προγόνων

Ένα μεγάλο μέρος της βιβλιογραφίας ασχολείται με τον ορισμό της σημασιολογίας των ελάχιστων κοινών προγόνων, που ορίζουν ένα ακριβέστερο και πληρέστερο σύνολο αποτελεσμάτων αναζήτησης [13, 22, 21, 26, 35, 52, 23, 28, 11, 17, 6, 29, 34, 49, 42, 2, 18, 1]. Οι βασικές σημασιολογίες φιλτραρίσματος των LCA μιας ερώτησης σε μια βάση δεδομένων είναι τέσσερις. Παραλλαγές αυτών έχουν επίσης προταθεί, οι οποίες χρησιμοποιώντας κάποια επιπλέον κριτήρια μπορούν να περικόψουν ακόμα παραπάνω από τη βασική προσέγγιση το σύνολο των αποτελεσμάτων. Οι τέσεερις κύριες σημασιολογίες διαχωρίζονται σ΄ αυτές που βασίζονται αποκλειστικά σε δομικά χαρακτηριστικά των κόμβων που περιέχουν τις λέξεις-κλειδιά και των LCA τους, δηλ. SLCA, ELCA και σ΄ αυτές που χρειάζονται και σημασιολογική πληροφορία και συγκεκριμένα τις ετικέτες των αντίστοιχων κόμβων, δηλ. VLCA, MLCA.



Σχήμα 2.1: Παράδειγμα δέντρου δεδομένων

Στο σχήμα 2.1 φαίνεται ένα παράδειγμα δενδρικής αναπαράστασης μιας βάσης δεδομένων πανεπιστημιακών μαθημάτων. Κάθε κόμβος διαθέτει μια ετικέτα, απογόνους και πιθανώς μια τιμή. Δεδομένης μιας ερώτησης, η αναζήτηση των κλειδιών της γίνεται ανάμεσα στις ετικέτες και τις τιμές των κόμβων του δέντρου. Για την ερώτηση {Physics James Harrison} , οι εμφανίσεις των λέξεων-κλειδιών σημειώνονται με έντονα γράμματα. Γι΄ αυτήν την ερώτηση, ο χρήστης πιθανότατα ενδιαφέρεται για κάποιο μάθημα ή σεμινάριο, που προσφέρεται από κάποιον James Harrison ή κάποιους James και Harrison. Οι κόμβοι 1, 2, 3, 12, 17, 22, και 39 είναι όλοι LCA των λέξεων-κλειδιών της ερώτησης. Τα πιο ακριβή, όμως, αποτελέσματα για την ερώτηση αυτή, με βάση την πιθανότερη ερμηνεία της είναι οι LCA που σημειώνονται ως l_1, l_2 και l_3. Ας δούμε τις τέσσερις βασικές σημασιολογίες φιλτραρίσματος στο συγκεκριμένο παράδειγμα.

***Ελάχιστοι χαμηλότεροι κοινοί πρόγονοι* smallest LCA - SLCA** Ως *ελάχιστοι* LCA (smallest - SLCA) ορίζονται οι LCA που δεν έχουν άλλους απόγονους LCA στα υποδέντρα τους [51, 37, 11].

Στο παράδειγμά μας, στους SLCA συμπεριλαμβάνονται οι κόμβοι l_2 και l_3. Ωστόσο, αποκλείεται από τα αποτελέσματα ο LCA l_1, καθώς υπάρχει ένας άλλος χαμηλότερος

κοινός πρόγονος, δηλ. ο LCA l_2 που είναι απόγονός του. Η απώλεια σωστών αποτελεσμάτων μειώνει την πληρότητα (recall) της προσέγγισης. Αντί του l_1, στους SLCA συμπεριλαμβάνεται το αποτέλεσμα l_4, το οποίο μάλλον δεν ενδιαφέρει το χρήστης καθώς συνδυάζει δύο διαφορετικά μαθήματα για να μπορέσει να καλύψει όλες τις λέξεις-κλειδιά. Τα μη σχετικά αποτελέσματα επηρεάζουν αρνητικά την ακρίβεια (precision) του συστήματος.

***Αποκλειστικοί* χαμηλότεροι κοινοί πρόγονοι (exclusive LCA - ELCA)** Χαλαρώνοντας τον περιορισμό πρόγονου-απόγονου LCA της σημασιολογίας smallest - SLCA, οι *αποκλειστικοί* LCA (exclusive - ELCA) ορίζονται ως εκείνοι οι LCA που είναι πιθανό να έχουν άλλους απόγονους LCA, αρκεί οι ίδιοι να αποτελούν χαμηλότερους κοινούς προγόνους στιγμιοτύπων λέξεων - κλειδιών που δε βρίσκονται στα υποδέντρα των απογόνων LCA τους [19, 52].

Έτσι, στην περίπτωση αυτή, ο LCA l_1 επιστρέφεται στο σύνολο των αποτελεσμάτων, καθώς ο l_2 ειναι μεν απόγονός του, αλλά ο l_1 είναι LCA όλων των λέξεων-κλειδιών με εμφανίσεις εκτός του υποδέντρου του l_2. Έτσι οι $ELCA$ αποτελούν μια βελτίωση σε σχέση με τους $SLCA$. Παρ΄ όλ αυτά, η σημασιολογία των ELCA αποτυγχάνει να αποκλείσει μη ενδιαφέροντα αποτελέσματα. Για παράδειγμα, επιστρέφει τον LCA l_4, όπως και η σημασιολογία SLCA. Έτσι, η σημασιολογία ELCA διευρύνοντας το σύνολο των αποτελεσμάτων σε σχέση με τη σημασιολογία SLCA, βελτιώνει σε κάποιες περιπτώσεις την πληρότητα της προσέγγισης αλλά χάνει ακόμα περισσότερο σε ακρίβεια.

***Πολύτιμοι* χαμηλότεροι κοινοί πρόγονοι (valuable LCA - VLCA)** Με κάπως διαφορετική σημασιολογία, υπάρχουν προσεγγίσεις που δε λαμβάνουν υπόψιν μόνο τη σχετική θέση μεταξύ χαμηλότερων κοινών προγόνων ή αυτών και των στιγμιοτύπων των λέξεων - κλειδιών, αλλά και τα τις ετικέτες των κόμβων του δένδρου. Οι *πολύτιμοι* (valuable -VLCA) LCA [13, 26] στοχεύουν στην ανακάληψη της πρόθεσης αναζήτησης του χρήστη εκμεταλλευόμενοι τις ετικέτες των κόμβων των μονοπατιών που συνδέουν τη ρίζα του δέντρου με τα στιγμιότυπα των λέξεων-κλειδιών. Αν σε δύο από τα μονοπάτια αυτά, εμφανίζονται κόμβοι με κοινή ετικέτα και δεν είναι και οι δυο φύλλα του δέντρου, τότε ο LCA απορρίπτεται.

Η σημασιολογία VLCA, στο παράδειγμα της Εικόνας 2.1, δεν επιστρέφει το αποτέλεσμα l_1 καθώς οι κόμβοι 5 και 8 έχουν την ίδια ετικέτα, `instructor`. Παρόλο που η σημασιολογία αυτή είναι διαισθητικά σωστή σε μερικές περιπτώσεις (π.χ. ο χρήστης ψάχνει για μάθημα του James Harrison), προβαίνει σε μονομερή ερμηνεία των ερωτήσεων, έχοντας την τάση να χάνει σωστά αποτελέσματα στη γενική περίπτωση. Επιπροσθέτως, δεν μπορεί να αποκλείσει μη σχετικά αποτελέσματα όταν η προϋπόθεση των κοινών ετικετών δεν παραβιάζεται. Στο παράδειγμά μας, αποτυγχάνει να εξαιρέσει τον LCA l_5, ο οποίος πιθανότατα δεν ενδιαφέρει το χρήστη, αλλά μπορεί να συνδυάσει τις λέξεις-κλειδιά μέσω του instructor ενός μαθήματος (3) και του speaker ενός σεμιναρίου (39).

***Σημαντικοί* χαμηλότεροι κοινοί πρόγονοι (meaningful LCA - MLCA)** Η σημασιολογία MLCA απαιτεί οι εμφανίσεις των λέξεων-κλειδιών, σε ένα έγκυρο ελάχιστο συνδετικό δέντρο ενός LCA, να είναι σημασιολογικά συσχετισμένες. Σύμφωνα με τους MLCA, δύο κόμβοι n_i και n_j στο υποδέντρο T ενός LCA είναι συσχετισμένοι αν δεν υπάρχει άλλος κόμβος n'_j με ίδια ετικέτα με τον n_j στο δέντρο δεδομένων, τέτοιος ώστε ο $LCA(n_i, n'_j)$ να είναι απόγονος του $LCA(n_i, n_j)$, δηλ. δεν υπάρχει άλλος κόμβος n'_j που να σχετίζεται πιο στενά με τον n_i απ΄ ό,τι ο n_j.

Στο παράδειγμα της Εικόνας 2.1, από τους MLCA απορρίπτεται το αποτέλεσμα l_1 καθώς οι κόμβοι (6, `fname`) και (10, `lname`) δε συσχετίζονται αρκετά στενά μεταξύ τους. Υπάρχουν οι κόμβοι (7, `lname`) και (9, `fname`), αντίστοιχα, που σχετίζονται σημασιολογικά πιο στενά με καθέναν απ΄ αυτούς. Έτσι, με μια διαφορετική θεώρηση απ΄ ό,τι οι VLCA, η σημασιολογία MLCA απορρίπτει το ίδιο αποτέλεσμα για παρόμοιο λόγο. Υποφέρει, όμως, από ανάλογα μειονεκτήματα της σημασιολογίας VLCA καθώς και σ΄ αυτήν την περίπτωση ο συνδυασμός του μαθήματος (3) και του σεμιναρίου (39) είναι εφικτός, αφού δεν υπάρχουν αντιστοιχίες και ομοιότητες ετικετών στα μονοπάτια των εμφανίσεων των λέξεων-κλειδιών τους.

Όλες αυτές οι προσεγγίσεις περικόπτουν το σύνολο των αποτελεσμάτων και, αναλόγως την περίπτωση, επιδεικνύουν χαμηλούς βαθμούς ακρίβειας ή και πληρότητας αποτελεσμάτων [49].

2.1.2 Αλγόριθμοι υπολογισμού χαμηλότερων κοινών προγόνων και των υποδέντρων τους

Ανάλογα με την προσέγγιση, τα αποτελέσματα της ερώτησης ορίζονται να είναι είτε οι LCA, είτε τα MCT ή ολόκληρα τα υποδέντρα με ρίζες τους LCA. Σε κάθε περίπτωση, η είσοδος των αλγορίθμων είναι οι ανεστραμμένες λίστες των λέξεων-κλειδιών της υποβεβλημένης ερώτησης στο δεδομένο σύνολο δεδομένων, ενώ το ζητούμενο της διαδικασίας είναι ο εντοπισμός των LCA, που είτε αυτούσιοι είτε με τα υποδέντρα τους απαντούν βέλτιστα στην ερώτηση.

Η πρόκληση για τους προτεινόμενους αλγορίθμους είναι η επίδοση. Ο μεγάλος αριθμός στιγμιοτύπων των λέξεων-κλειδιών καθιστά τον υπολογισμό απαντήσεων σε ερωτήσεις με λέξεις-κλειδιά εξαιρετικά ακριβό. Στη χειρότερη περίπτωση, ο αριθμός των αποτελεσμάτων είναι εκθετικός στον αριθμό των λέξεων-κλειδιών. Για την ακρίβεια, κάθε συνδυασμός στιγμιοτύπων των λέξεων-κλειδιών μιας ερώτρησης μπορεί να οδηγήσει σε ένα διακριτό αποτέλεσμα. Στην πράξη, βέβαια, εξαιτίας της δομής δέντρου, υπάρχουν επικαλύψεις στους LCA όλων αυτών των συνδυσαμών. Ο αριθμός των διακριτών LCA εξαρτάται από το μέγεθος του συνόλου δεδομένων. Σε ένα πλατύ δένδρο με ομοιόμορφη κατανομή στιγμιοτύπων συχνών λέξεων-κλειδιών ή σε ένα βαθύ σύνολο δεδομένων, όπου συνδυασμοί και, συνακόλουθα, LCA μπορούν να προκύψουν σε διάφορα βάθη, οι επικαλύψεις των χαμηλότερων κοινών προγόνων μειώνονται.

Η αποδοτικότητα των αλγορίθμων που προτείνονται για να παράγουν απαντήσεις σε ερωτήσεις λέξεων - κλειδιών σε δενδρικά δεδομένα εξαρτώνται και από την υιοθετούμενη σημασιολογία LCA. Όσο περιορίζεται το σύνολο των αποτελεσμάτων από την υιοθετούμενη σημασιολογία, τόσο λιγότεροι συνδυασμοί χρειάζεται να υπολογιστούν. Ο σχεδιασμός αλγορίθμων που υιοθετούν μια σημασιολογία φιλτραρίσματος, βασίζεται στην έγκαιρη απόρριψη χαμηλότερων κοινών προγόνων που δεν εμπίπτουν στο σύνολο των αποδεκτών LCA ανάλογα με την προσέγγιση. Πολλές φορές, για το σκοπό αυτό, επιστρατεύονται κατάλληλες δομές ευρετηρίου, επιπροσθέτως στις ανεστραμμένες λίστες των λέξεων-κλειδιών, οι οποίες βοηθούν στην επιτάχυνση των υπολογισμών.

Δύο διαφορετικοί αλγόριθμοι που υπολογίζουν SLCA παρουσιάστηκαν στην εργασία [51], οι οποίοι εκμεταλλεύονται τη θέση των SLCA χαμηλά στο δέντρο για να εξαιρέσουν, νωρίς, από την επεξεργασία προγόνους LCA. Σ΄ αυτήν την προσέγγιση εισάγεται επίσης και μια επέκταση του βασικού αλγορίθμου, η οποία μπορεί να επιστρέψει το πλήρες σύνολο των LCA που απαντούν σε μια ερώτηση, επαυξάνοντας το

σύνολο των SLCA. Ένας ακόμα αλγόριθμος υπολογισμού SLCA, τόσο για συζευκτικές όσο και για διαζευκτικές ερωτήσεις λέξεων - κλειδιών, αναπτύχθηκε στο [48]. Ο αλγόριθμος Indexed Stack [52] και ο αλγόριθμος Hash Count [57] βελτιώνουν την αποδοτικότητα του [19] στον υπολογισμό ELCA χρησιμοποιώντας κατάλληλες δομές ευρετηρίου.

Το μεγαλύτερο μέρος των προτεινόμενων αλγορίθμων στη βιβλιογραφία βασίζεται στη χρήση στοιβών. Η αιτία γι΄ αυτό είναι το γεγονός ότι η στοίβα προσφέρεται ως δομή για την κατά βάθος διάσχιση ενός δέντρου. Για το σύστημα XRank [19], αναπτύχθηκε για πρώτη φορά ένας αλγόριθμος με χρήση στοίβας, ο οποίος με είσοδο την ανεστραμμένη λίστα λέξεων - κλειδιών εξάγει μια ταξινομημένη λίστα ELCA. Η στοίβα που χρησιμοποιείται στους αλγορίθμους αυτής της οικογένειας, περιέχει σε κάθε χρονική στιγμή τους κόμβους που ανήκουν σε ένα μονοπάτι του δέντρου. Ενέργειες εισαγωγής (push) επαυξάνουν το μονοπάτι με νέους κόμβους και ενέργειες εξαγωγής (pop) αφαιρούν από το μονοπάτι κόμβους, γυρίζοντας την επεξεργασία πίσω προς μικρότερα βάθη του δέντρου. Η επεξεργασία νέων στογμιοτύπων λέξεων-κλειδιών εισάγει νέους κόμβους στη στοίβα. Η εξαγωγή στοιχείων οδηγεί στην αποθήκευση πληροφορίας απόγονων κόμβων στους προγόνους τους. Ανάλογα με την προσέγγιση, η πληροφορία αυτή συνδυάζεται κατάλληλα με πληροφορία που έχει ήδη αποθηκευτεί από προηγούμενα υποδέντρα του τρέχοντος κόμβου. Έτσι, όταν η επεξεργασία φτάσει στη ρίζα του δέντρου, όλα τα υποδέντρα της ρίζας έχουν εξεταστεί και αναδρομικά όλα τα υποδέντρα των απογόνων της ρίζας, που σχετίζονται μέσω σχέσεων προγόνου-απογόνου με τα στογμιότυπα των λέξεων κλειδιών. Σ΄ αυτό το σημείο προκύπτουν τα αποτελέσματα των ερωτήσεων. Αποδοτικοί αλγόριθμοι για την εύρεση SLCA και ELCA παρουσιάζονται στις εργασίες [51, 52, 57] και [54]. Οι συγγραφείς του [21] αναπτύσσουν έναν αλγόριθμο που υπολογίζει όλα τα αποτελέσματα μιας ερώτησης με λέξεις-κλειδιά σε XML δεδομένα, χρησιμοποιώντας μια συνοπτική αναπαράστασή των υποδέντρων τους. Στην εργασία [11] σχεδιάζονται αλγόριθμοι για τον αποδοτικό υπολογισμό κορυφαίνων k SLCA και ELCA. Νέες μορφές ανεστραμμένων λιστών και και αλγόριθμοι αποτίμησης με σημασιολογία SLCA και ELCA προτείνονται στις εργασίες [56] και [55].

2.2 Ταξινόμηση αποτελεσμάτων

Οι διαφορετικές σημασιολογίες χαμηλότερου κοινού προγόνου αξιοποιούνται από τις διάφορες προσεγγίσεις που τις υιοθετούν ως μέσο φιλτραρίσματος του μεγάλου αριθμού πιθανών αποτελεσμάτων αναζήτησης. Παρ΄ όλ΄ αυτά, στο χώρο του κλασικού IR (information retrieval) έχει αποδειχθεί ότι είναι πιο αποτελεσματικές οι προσεγγίσεις που επιστρέφουν το πλήρες σύνολο αποτελεσμάτων ταξινομημένων όμως με βάση κάποιο μοντέλο βαθμολόγησης και κατάταξης. Προσεγγίσεις όπως οι [5, 6, 42, 29] εφαρμόζουν μοντέλα κατάταξης αποτελεσμάτων που βασίζονται σε τεχνικές IR. Τέτοιες μέθοδοι εμπίπτουν συνήθως στο (α) διανυσματικό μοντέλο ή (β) στο πιθανοτικό μοντέλο και ενίοτε συνδυάζονται με αξιολόγηση της δομής και της συσχέτισης των κόμβων που περιέχουν τους όρους προς αναζήτηση. Τα μοντέλα αυτά απαιτούν, ωστόσο, εκτενή στατιστική ανάλυση του συνόλου των δεδομένων και πολύπλοκες διαδικασίες κατά την αναζήτηση, οι οποίες σε πολλές περιπτώσεις, καθιστούν αυτές τις λύσεις μη εφαρμόσιμες στην πράξη ή της αποκλείουν σε περιπτώσεις δεδομένων που μεταβάλλονται συχνά ή σε περιπτώσεις ροών δεδομένων.

Οι προσεγγίσεις φιλτραρίσματος μέσω της υιοθέτησης κάποιας από τις σημασιολογί-

ες χαμηλότερου κοινού προγόνου, μπορούν επίσης να συνδυαστούν με κάποιο μοντέλο κατάταξης [19, 13, 27, 11, 28, 49, 29, 42, 2]. Σ΄ αυτήν την περίπτωση, οι προτεινόμενοι αλγόριθμοι επωφελούνται από την περικοπή των αποτελεσμάτων, που απορρίπτονται εκ προοιμίου λόγω της σημασιολογίας. Αν, όμως, το φιλτράρισμα που διενεργείται επιδρά αρνητικά στην πληρότητα (recall) του συνόλου των αποτελεσμάτων, τότε η κατάταξη των αποτελεσμάτων ξεκινά εξ ορισμού από χαμηλότερη βάση, ως προς την ποιότητα της απάντησης στην αναζήτηση του χρήστη. Οι προσεγγίσεις αυτές αποδεικνύονται, μεν, πιο αποδοτικές ως προς την αποτίμηση των ερωτήσεων, αλλά υποφέρουν εξίσου από την ανάγκη στατιστικής ανάλυσης του συνόλου των δεδομένων.

Οι προσεγγίσεις ταξινόμησης αποτελεσμάτων στην αποτίμηση λέξεων-κλειδιών σε δενδρικά δεδομένα επιστρέφουν μια ταξινομημένη λίστα LCA, υποδέντρων, MCT ή άλλων δομών σύνοψης αποτελεσμάτων, με βάση την εκτιμώμενη σχετικότητα με την υποβεβλημένη ερώτηση. Το σύστημα XRank [19] ορίζει μια λογική ταξινόμησης που βασίζεται στον αλγόριθμο PageRank [8]. Το σύστημα XSearch [13] αξιοποιεί την ευρέως αποδεκτή μετρική $tf * idf$ [44] (term frequency-inverse document frequency) από το πεδίο της Ανάκτησης Πληροφορίας (Information Retrieval). Το $tf * idf$ προσδιορίζει την αξία ενός όρου ανάλογα με τη συχνότητα εμφάνισής του σε ένα έγγραφο και αντιστρόφως ανάλογα με τη συχνότητα εμφάνισής του στη συλλογή εγγράφων όπου υποβάλλεται μια ερώτηση. Για την εκμετάλλευση του $tf * ief$ σε δενδρικά δεδομένα απαιτείται η προσαρμογή του μεγέθους. Αυτό που συμβαίνει είναι ότι αντί των εγγράφων ως σημείο αναφοράς χρησιμοποιούνται τα στοιχεία ενός XML εγγράφου ή αλλιώς οι κόμβοι του δέντρου. Έτσι, προκύπτει το ανάλογο του $tf * idf$ μέγεθος $tf * ief$ (term frequency-inverse element frequency) [45].

Ένα σύνθετο σύστημα ταξινόμησης υιοθετείται από την προσέγγιση XReal [6], όπου χρησιμοποιείται μια προσαρμογή του $tf * idf$. Στο XReal μια σειρά από αρχές, δομικές και σημασιολογικές, καθορίζουν τη συνολική αξία ενός κόμβου μες στο δένδρο δεδομένων ως πιθανού αποτελέσματος σε μια ερώτηση. Η τελική συνάρτηση, που δίνει τη θέση ενός κόμβου στην ταξινομημένη απάντηση σε μια ερώτηση με λέξεις-κλειδιά, περιλαμβάνει ποσοτικοποιημένα όλα αυτά τα κριτήρια μαζί με μια έκφραση του $tf * idf$.

Κάποιες προσεγγίσεις χρησιμοποιούν ακόμα πιο εξελιγμένα μεγέθη του πεδίου IR για την αξιολόγηση αποτελεσμάτων ερωτήσεων με λέξεις-κλειδιά σε δενδρικά δεοδμένα. Στη δουλειά, που έχει γίνει στο πλαίσιο της εργασίας [49], ορίζεται ένα μοντέλο υπολογισμού της εντροπίας των κόμβων ενός δέντρου, με βάση τις δομικές συσχετίσεις των ετικετών του δέντρου και τις εμφανίσεις των λέξεων κλειδιών στις τιμές των κόμβων. Έτσι, κάποιοι κόμβοι κατατάσσονται συνολικά ως πιο αξιόλογοι για αποτελέσματα σε μια ερώτηση από άλλους και ανεξάρτητα από την υποβεβλημένη ερώτηση. Στην εργασία [42] χρησιμοποιείται η αμοιβαία πληροφορία (mutual information) για την έκφραση της συσχέτισης των στιγμιοτύπων των λέξεων κλειδιών. Με βάση το βαθμό συσχέτισης ορίζεται σημασιολογία κορυφογραμμής (skyline) για την επιλογή των επικρατέστερων αποτελεσμάτων μιας ερώτησης. Τα σύνθετα εξελιγμένα μεγέθη που αξιοποιούνται στις δουλειές αυτές απαιτούν την ανάλυση του συνόλου δεδομένων εκ των πρωτέρων. Αυτό αποκλείει την επεξεργασία δεδνδρικών δεδομένων που τροφοδοτούν το σύστημα σε μορφή ροής. Επιπλέον, οι αλγόριθμοι που έχουν σχεδιαστεί για αποτίμηση των ερωτήσεων με αυτή τη λογική καθίστανται πρακτικά μη χρησιμοποιήσιμοι από πλευράς επίδοσης χρόνου [49]. Για τη βελτίωση της επίδοσης αναγκαστικά επιστρατεύονται επιπροσθέτως και κάποιες δομές ευρετηρίου, ή γίνονται παραδοχές για την εκ των πρωτέρων περικοπή αποτελεσμάτων. Ταυτόχρονα, παρόλο που οι προσεγγίσεις βασίζονται σε πολύπλοκες μεθόδους αξιολόγησης των αποτελεσμάτων δεν

αποδεικνύονται εν γένει πιο αποτελεσματικές ως προς την ποιότητα των απαντήσεων σε σχέση με άλλες ταχύτερες μεθόδους.

Η απόσταση μεταξύ των στιγμιοτύπων λέξεων-κλειδιών αναγνωρίζεται από το πεδίο της Ανάκτησης Πληροφορίας (IR) ως ένα από τα κριτήρια εκτίμησης της ποιότητας ενός αποτελέσματος. Για το λόγο αυτό, ορίζεται η αναζήτηση εγγύτητας (proximity search), με τις διάφορες μηχανές αναζήτησης να υποστηρίζουν τελεστές όπως NEAR ή τον ακριβή ορισμό της απόστασης μευταξύ δύο λέξεων-κλειδιών, σε αριθμό λέξεων. Στα ημιδομημένα δεδομένα, η παρουσία δομής δεν αφήνει περιθώρια για μονοσήμαντο ορισμό της απόστασης μεταξύ δύο λέξεων-κλειδιών. Η κοντινότερη προσέγγιση της εγγύτητας των λέξεων-κλειδιών είναι το μέγεθος του υποδέντρου που τις περιέχει. Προς αυτήν την κατεύθυνση σχεδιάστηκε ο αλγόριθμος SA [21], όπου τα αποτελέσματα ορίζονται με βάση μια μορφή σύνοψης δέντρων, που επιδέχεται κατώφλι μεγέθους για περικοπή αποτελεσμάτων που υπερβαίνουν το δεδομένο μέγεθος. Ο αλγόριθμος SA υπολογίζει εξαντλητικά όλους τους συνδυασμούς στιγμιοτύπων λέξεων-κλειδιών και έτσι καταλήγει να μην μπορεί να χρησιμοποιηθεί πρακτικά για υπολογισμό απαντήσεων σε ερωτήσεις με περισσότερες από 5 λέξεις-κλειδιά σε πραγματικά σύνολα δεδομένων με ικανό μέγεθος. Στη σφαίρα της εγγύτητας των λέξεων-κλειδιών προτείνονται μερικές απλοποιημένες προσεγγίσεις. Στα συστήματα XSearch [13] και SAIL [27] χρησιμοποιείται η απόσταση ανα δύο των στιγμιοτύπων των λέξεων-κλειδιών, με τη συνακόλουθη απαίτηση για προεπεξεργασία του συνόλου δεδομένων. Μια ακόμα πιο απλοποιημένη προσέγγιση είναι η απόσταση κάθε στιγμιοτύπου από τον αντίστοιχο LCA. Παράδειγμα χρήσης αυτού του μέτρου είναι το σύστημα XRank [19].

2.3 Ομαδοποίηση αποτελεσμάτων

Η μέθοδος της συσταδοποίησης αποτελεσμάτων έχει εφαρμοστεί για την επίλυση του προβλήματος της ασάφειας των ερωτήσεων με λέξεις-κλειδιά στο πεδίο της Ανάκτησης Πληροφορίας (Information Retrieval) [9, 24]. Σ′ αυτό το πλαίσιο, τα αποτελέσματα μιας ερώτησης συσταδοποιούνται, οι συστάδες επισημειώνονται με τους αντιπροσωπευτικούς όρους τους και οι χρήστες επιλέγουν τελικά τα αποτελέσματα που τους ενδιαφέροντός τους, επιλέγοντας τη συστάδα με την κοντινότερη επισημείωση στην προσδοκόμενη ερμηνεία της ερώτησής τους. Εξαιτίας της ιδιαιτερότητας των δεδνδρικών δεδομένων, που διαέτουν δομή και δεν είναι σαφές εξ αρχής το επίπεδο λεπτομέρειας των επιθυμητών αποτελεμσάτων οι προσεγγίσεις συσταδοποίησης των εγγράφων δεν είναι άμεσα εφαρμόσιμες, στο πλαίσιο αυτό.

Η συσταοδαποίηση ολόκληρων XML εγγράφων έχει μελετηθεί στο παρελθόν για λόγους εξόρυξης πληροφορίας [14, 32]. Ωστόσο, οι προσεγγίσεις αυτές δε λαμβάνουν υπόψη τη δεδομένη ερώτηση και εφαρμόζονται σε επίπεδο εγγράφου και όχι υποδέντρου.Εξαίρεση αποτελούν ελάχιστες προσεγγίσεις που ομαδοποιούν αποτελέσματα σε επίπεδο υποδέντρου. Στην εργασία [34] προτείνεται ένας τρόπος ομαδοποίησης αποτελεσμάτων που βασίζεται στα δενδρικά πρότυπά τους. Η προσέγγιση [38] συσταδοποιεί τα αποτελέσματα με βάση την κατηγοριοποίηση των κόμβων που περιέχουν. Οι χρήστες μπορούν να αποφασίζουν το βαθμό λεπτομέρειας της μεθόδου συσταδοποίησης και να καθορίζουν τον αριθμό από τις συστάδες.

2.4 Άλλες προσεγγίσεις

Επιπροσθέτως, διάφορα προβλήματα στο πλαίσιο της αναζήτησης με λέξεις-κλειδιά έχουν αναδειχθεί μέσα από προηγούμενες μελέτες. Τα συστήματα XReal [5, 6] και XBridge [28] προτείνουν τρόπους να αναγνωριστεί η πρόθεση αναζήτησης του χρήστη. Το XReal ορίζει σαν τύπο κόμβου το μονοπάτι ετικετών από τη ρίζα του δέντρου μέχρι τον εν λόγω κόμβο. Για κάθε τύπο χρησιμοποιείται μια παραλλαγή του μεγέθους $tf * idf$ για να αξιολογηθεί η αξία του ως υποψήφιου τύπου αποτελέσματος. Στο ίδιο μήκος κύματος, το XBridge χρησιμοποιεί μια συνάρτηση βαθμολόγησης (scoring function) των τύπων των κόμβων, λαμβάνοντας όμως υπόψη και τη δομή κάθε επιμέρους αποτελέσματος του συγκεκριμένου τύπου.

Το σύστημα XSeek [35] διχωρίζει τους κόμβους ενός δέντρου σε κατηγορίες: οντότητα, χαρακτηριστικό ή συνδετικός κόμβος. Χρησιμοποιεί τις κατηγορίες αυτές για να διαχωρίσει ποιοι κόμβοι είναι επιλέξιμοι ως αποτελέσματα και ποιοι μπορούν να χρησιμοποιηθούν ως συνθήκες για φιλτράρισμα των αποτελεσμάτων. Ένα άλλο σύστημα, το XMean [34], εισάγει την ιδέα των εννοιολογικά συσχετισμένων κόμβων και στηρίζει τη σημασιολογία που εφαρμόζεται για την επιλογή των αποτελεσμάτων και την κατηγοριοποίησή τους στην ιδέα αυτή. Τα πρότυπα (patterns) των αποτελεσμάτων χρησιμοποιούνται για συσταδοποίησή τους. Στη συνέχεια, ορίζονται κάποιοι κανόνες γενίκευσης των προτύπων, βάσει των οποίων κατασκευάζεται μια ιεραρχία μέσα από την οποία ο χρήστης καλείται να περιηγηθεί για να εντοπίσει τα αποτελέσματα ενδιαφέροντός του. Η εργασία [38] επεκτείνοντας τις ιδέες του XSeek, αναγνωρίζουν τους υποψήφιους κόμβους για αποτελέσματα και στη συνέχεια τα ομαδοποιούν με βάση τους τύπους των στιγμιοτύπων των λέξεων-κλειδιών, δηλ. ανάλογα αν στο XML έγγραφό πρόκειται για στοιχεία ή χαρακτηριστικά.

Η αναζήτηση με λέξεις-κλειδιά σε XML με βάση τα συμφραζόμενα εξετάζεται στην εργασία [7]. Τα συμφραζόμενα, σάυτήν την περίπτωση, εκφράζονται ως μονοπάτι σ΄ ένα XML δέντρο και τα αποτελέσματα ταξινομούνται λαμβάνοντας υπόψιν την πληροφορία αυτή. Σε άλλο πλαίσιο, έχουν επίσης προταθεί υλοποιημένες όψεις (materialized views) για την υποστήριξη της αποτίμησης ερωτήσεων με λέξει-κλειδιά σε XML δεδομένα [36, 47]. Η εξειδίκευση ερωτήσεων με λέξεις-κλειδιά (keyword query refinement) και τεχνικές σύστασης λέξεων-κλειδιών (kwyword suggestion) μελετώνται στις εργασίες [43] και [40].

Τέλος, υπάρχουν προσεγγίσεις που επικεντρώνονται στην αποσαφήνιση της ερώτησης, ώστε να βελτιώσουν την ποιότητα των αποτελεσμάτων. Παραδείγματα αυτής της λογικής είναι οι εργασίες [15] και [16]. Ο στόχος σ΄ αυτές τις περιπτώσεις είναι να προκύψει μία ερώτηση με δομή επαθξάνοντας την αρχική ερώτηση με λέξει-κλειδιά. Η διαδικασία υποβοηθάται και από τον ίδιο το χρήστη.

Μια αναλυτική επισκόπηση και σύγκριση των κυρίαρχων προσεγγίσεων στην τρέχουσα βιβλιογραφία, παρουσιάζεται στη μέλέτη [39].

Κεφάλαιο 3

Σημασιολογία συμπαγούς χαμηλότερου κοινού προγόνου TLCA

3.1 Εισαγωγή

Οι δενδρικές δομές δεδομένων, π.χ. σε μορφή XML, JSON, YAML, έχουν καταστεί ευρέως χρησιμοποιούμενες για την εξαγωγή και ανταλλαγή πληροφορίας στο διαδίκτυο. Η αναζήτηση με λέξεις-κλειδιά, από την άλλη, είναι ο πιο δημοφιλής τρόπος για άντληση πληροφορίας από το διαδίκτυο, καθώς απελευθερώνει τους χρήστες α) από την απαίτηση να κατέχουν εξειδίκευση στη γνώση μιας πολύπλοκης γλώσσας ερωτήσεων (π.χ. SQL) και β) από την ανάγκη να γνωρίζουν το σχήμα της πηγής πληροφορίας στην οποία απευθύνουν τις ερωτήσεις τους.

Σε αντίθεση με την αναζήτηση σε επίπεδο κείμενο, η αναζήτηση σε δενδρικά δεδομένα δεν επιστρέφει ολόκληρα έγγραφα, αλλά κατάλληλα επιλεγμένα κομμάτια (δηλ. υποδέντρα) του συνόλου δεδομένων [46, 19]. Μια μεγάλη μερίδα της βιβλιογραφίας ασχολείται με τη μορφή [19, 35, 5, 6, 11, 39, 34] και τον εντοπισμό των σωστότερων αποτελεσμάτων [19, 13, 31, 51, 21, 48, 26, 52, 37, 39, 49, 23] ανάμεσα σ΄ αυτά που απαρτίζουν τη δενδρική βάση δεδομένων. Συνήθως τα αποτελέσματα της αναζήτησης ορίζονται ως τα ελάχιστα συνδετικά υποδέντρα που περιλαμβάνουν ένα στιγμιότυπο από κάθε λέξη-κλειδί της ερώτησης. Αυτά τα ελάχιστα συνδετικά υποδέντρα αντιπροσωπεύονται από τη ρίζα τους, που είναι ο χαμηλότερος κοινός πρόγονος (lowest common ancestor, αναφερόμενος ως LCA για συντομία εφεξής) των εν λόγω στιγμιοτύπων στο δέντρο. Προσεγγίσεις, που επιλέγουν ένα υποσύνολο των χαμηλότερων κοινών προγόνων ως απάντηση στην αναζήτηση, καλούνται προσεγγίσεις *φιλτραρίσματος*, καθώς απορρίπτουν από το σύνολο αποτελεσμάτων τους, τους LCA που θεωρούνται μη σχετικοί με την ερώτηση [13, 31, 51, 26, 52]. Παρόλο που, προσεγγίσεις φιλτραρίσματος διαισθητικά έχων λογική, είναι ευρηστικές και συχνά αποδεικνύονται ελλειματικές στην πράξη καταλήγοντας σε χαμηλά επίπεδα ακρίβειας και πληρότητας των αποτελεσμάτων [49].

Μια ανώτερη στρατηγική είναι να μην απορρίπτονται εκ των πρωτέρων κάποια αποτελέσματα της αναζήτησης, αλλά να *ταξινομούνται* οι LCA, με τους πιο σχετικούς στην κορυφή της κατάταξης. Η αποτελεσματική ταξινόμηση, για το λόγο αυτό, βελτιώνει με διαφορά την αποτελεσματικότητα του συστήματος αναζήτησης. Οι περισσότερες προσεγγίσεις που πραγματοποιούν κάποιου είδους ταξινόμηση, αντλούν έμπνευση από το

πεδίο της ανάκτησης πληροφορίας (information retrieval - IR), προσαρμόζοντας κατάλληλα στην ιεραρχική μορφή των δέντρων μετρικές και τεχνικές, που υιοθετούνται για αναζήτηση σε συλλογές εγγράφων επίπεδου κειμένου (π.χ. το μοντέλο συχνότητας εμφάνισης όρων $tf * idf$ ή την τεχνική PageRank [8]) [19, 13, 49, 3]. Αναγνωρίζοντας το γεγονός ότι οι χρήστες συνήθως ενδιαφέρονται για ένα μικρό αριθμό μόνο αποτελεσμάτων, ορισμένες εργασίες προτείνουν αλγορίθμους για την επιστροφή των κορυφαίων k αποτελεσμάτων (τοπ-κ) ερωτήσεων με λέξεις-κλειδιά σε δενδρικά δεδομένα [27, 11, 29]. Ο στόχος αυτών των αλγορίθμων είναι ο υπολογισμός των κορυφαίων αποτελεσμάτων χωρίς να χρειαστεί ο υπολογισμός του πλήρους συνόλου και η ταξινόμησή του.

3.1.1 Προκλήσεις

Παρόλη την εκτενή βιβλιογραφία, οι διάφορες προσεγγίσεις αντιμετωπίζουν μερικά ή όλα από τα παρακάτω προβλήματα:

Πρόβλημα 1: Απόδοση και κλιμάκωσή της. Ο αριθμός των LCA για μια δεδομένη ερώτηση με λέξεις-κλειδιά μπορεί να φτάσει σε μεγάλα μεγέθη. Παρόλο, που περισσότερα του ενός συνδετικά υποδέντρα μπορεί να εντιρποσωπεύονται από τον ίδιο LCA, ο συνολικός τους αριθμός στη χειρότερη περίπτωση είναι εκθετικός στο μέγεθος της ερώτησης (δηλ. τον αριθμό των λέξεων-κλειδιών). Η πολυπλοκότητα αλγορίθμων που έχουν ήδη προταθεί και οι οποίοι υπολογίζουν και πιθανώς ταξινομούν το σύνολο των LCA εξαρτάται από το γινόμενο των μεγεθών των ανεστραμμένων λιστών των λέξεων-κλειδιών [21, 49, 34]. Συνεπώς, οι αλγόριθμοι αυτοί δεν μπορούν εγγενώς να κλιμακωθούν ομαλά όσο το μέγεθος της βάσης δεδομένων και ο αριθμός των κλειδιών της ερώτησης αυξάνονται.

Πρόβλημα 2: Εξάρτηση από βοηθητικές δομές ευρετηρίου. Για να αντιμετωπιστεί το πρόβλημα της απόδοσης μια σειρά από προσεγγίσεις αξιοποιεί τη δυνατότητα κατασκευής βοηθητικών δομών ευρετηρίου, πάνω στις ανεστραμμένες λίστες των λέξεων-κλειδιών (π.χ. B+-tree [52], ranked Dewey inverted list kai B+-tree [19], data summary index [34], hash count index [57]). Επιπροσθέτως, πολλές προσεγγίσεις προχωρούν επίσης σε βοηθητικές δομές και στατιστική ανάλυση της πληροφορίας για αποδοτικότερη υλυποίηση της βασικής υιοθετούμενης σημασιολογίας ερωτήσεων[13, 12], νορμαλιζεδ τοταλ ςορρελατιον [49]). Η κατασκευή αυτών των βοηθητικών δομών απαιτεί αναγκαστικά την αποθήκευση και επεξεργασία των ανεστραμμένων λιστών όλων των λέξεων κλειδιών της βάσης δεδομένων. Αυτή η διαδικασία δεν είναι μόνο απαιτητική στη διάσταση του χρόνου, αλλά καθιστά και τέτοιες λύσεις μη εφαρμόσιμες στην πράξη σε μια σειρά από περιπτώσεις, όπως, για παράδειγμα, σάυτήν της δυναμικής ροής δεδομένων.

Πρόβλημα 3: Ποιότητα της απάντησης. Για την αποφυγή της παραγωγής μεγάλου αριθμού LCA, οι διάφορες προσεγγίσεις ταξινόμησης και επιστροφής κορυφαίων k αποτελεσμάτων [19, 13, 11, 29] παράγουν και τελικά ταξινομούν όχι το πλήρες σύνολο των LCA αλλά μόνο ένα μικρό υποσύνολο αυτών, όπως καθορίζεται από τη σημασιολογία φιλτραρίσματος που υιοθετούν (π.χ. SLCA [51, 21, 48, 37], ELCA [19, 52, 57]). Η στρατηγική αυτή, παρότι υπολογιστικά ελκυστική, σημασιολογικά είναι ανεπαρκής. Παρά την πιθανότητα ορισμού καλών κριτηρίων ταξινόμησης, 'τιμωρεί' την απάντηση μιας ερώτησης με τα μειονεκτήματα που κληρονομούνται από τη σημασιολογία φιλτραρίσματος. Για παράδειγμα, αν σχετικά αποτελέσματα παραλειφθούν από την απάντηση λόγω της σημασιολογίας φιλτραρίσματος, δεν υπάρχει τρόπος να ανακτηθούν και να

επιστραφούν στο χρήστη όσο καλής ποιότητας ταξινόμηση κι αν εφαρμοστεί στο επόμενο βήμα.

Πρόβλημα 4: Διεπαφή χρήστη κορυφαίων k προσεγγίσεων. Με στόχο να μπορέσουν οι χρήστες να διαχειριστούν ένα πιθανώς μεγάλο σύνολο αποτελεσμάτων αλλά και για λόγους επίδοσης, πολλές προσεγγίσεις ακολουθούν την τακτική της επιστροφής μόνο των κορυφαίων k αποτελεσμτάτων σε ερωτήσεις λέξεων-κλειδιών [11, 29]. Ωστόσο, η επιλογή της κατάλληλης τιμής για την παράμετρο k δεν είναι απλή υπόθεση: η επιλογή μιας πολύ μικρής τιμής μπορεί να απορρίψει χρήσιμα αποτελέσματα, ενώ η επιλογή μιας μεγάλης τιμής μπορεί να κατακλείσει το χρήστη με άσχετα αποτελέσματα και ταυτόχρονα να επιβαρύνει αναίτια το σύστημα από πλευράς απόδοσης. Η χρήση μιας καλής στρατηγικής ταξινόμησης μπορεί να ανασκευάσει το πρόβλημα του πλήθους άσχετων αποτελεσμάτων στην απάντηση, αλλά δεν μπορεί να λύσει το πρόβλημα της απόδοσης. Η επιτυχής επιλογή τιμής για την παράμετρο k απαιτεί τη γνώση του αριθμού των αποτελεσμάτων, ο οποίος με τη σειρά του εξαρτάται από το μέγεθος και τη δομή της δενδρικής βάσης δεδομένων, τον αριθμό των λέξεων-κλειδιών στην ερώτηση και τις συχνότητες εμφάνισής τους στη βάση δεδομένων. Η απαίτηση γνώσης των στατιστικών χαρακτηριστικών της βάσης δεδομένων από το χρήστη και συανκόλουθη πολύπλοκη εκτίμηση του πιθανού αριθμού αποτελεσμάτων είναι ουτοπική και ακυρώνει την απλότητα που πρεσβεύει το μοντέλο της αναζήτησης με λέξεις-κλειδιά.

3.1.2 Η προσέγγισή μας

Αντιμετωπίζουμε όλα τα παραπάνω προβλήματα εισάγοντας μια νέα σημασιολογία για την αποτίμηση ερωτήσεων με λέξεις-κλειδιά, η οποία ορίζει ένα μοντέλο ταξινόμησης με ενδιαφέροντα νέα χαρακτηριστικά. Προτείνουμε μια ιδέα που οδηγεί στο σχεδιασμό αποδοτικών αλγορίθμων με δυνατότητες αποτίμησης σε μεγάλες βάσεις δεδομένων του πλήρους συνόλου των αποτελεσμάτων μιας ερώτησης και χωρίς την ανάγκη επιπλέον δομών ευρετηρίου για την επιτάχυνση των υπολογισμών. Με βάση τη νέα σημασιολογία ορίζουμε ένα νέο μοντέλο ανάκτησης των κορυφαίων k αποτελεσμάτων σε μια ερώτηση, που είναι διασθητικά βοηθητική για το χρήστη και ταυτόχρονα βελτιώνει την επίδοση του συστήματός μας. Η μέθοδός μας, αφενός επειδή υπολογίζει το πλήρες σύνολο των αποτελεσμάτων αλλά και αφετέρου εξαιτίας του σχήματος ταξινόμησής μας επιτυγχάνει άριστη ποιότητα αποτελέσματος.

Η βάση της σημασιολογίας μας είναι η έννοια του μεγέθους χαμηλότερου κοινού προγόνου (LCA size), η οποία αντικατοπτρίζει την εγγύτητα εμφάνισης των στιγμιοτύπων των λέξεων-κλειδιών σε ένα δενδρικό σύνολο δεδομένων. Σε αναλογία με το κλασικό πεδίο του IP, η εγγύτητα των λέξεων-κλειδιών χρησιμοποιείται σαν κριτήριο ποσοτικής εκτίμησης της σχετικότητας ενός αποτελέσματος με την ερώτηση.

Η αποδοτικότητα των αλγόρθμων οφείλεται στη χρήση ενός δικτυωτού των διαμερίσεων του συνόλου των λέξεων - κλειδιών. Η μέθοδός μας ακολουθεί τα μονοπάτια του δικτυωτού για να συνδυάσει σταδιακά τα στιγμιότυπα των λέξεων - κλειδιών σε μερικούς LCA, επιτρέποντας τον αποκλεισμό συνδυασμών των ίδων λέξεων - κλειδιών που παρουσιάζονται πιο απομακρυσμένα πριν αυτά συνεισφέρουν σε ένα τελικό αποτέλεσμα. Η τεχνική αυτή αποφεύγει τον εξαντλητικό υπολογισμό όλων των δυνατών συνδυασμών για να υπολογίσει το σύνολο των LCA. Κατά συνέπεια οι αλγόριθμοί μας κλιμακώνονται ομαλά όταν το μέγεθος της βάσης δεδομένων αυξάνεται και αντιμετωπίζουν επιτυχώς το πρόβλημα 1. Το ενδιαφέρον είναι πως η αποδοση αυτή επιτυγχάνεται χωρίς να χρειάζεται να επιστρατευτούν βοηθητικές δομές ευρετηρίου, αποφεύγοντας

έτσι το βήμα της προεπεξεργασίας που άλλες προσεγγίσεις απαιτούν (πρόβλημα 2).

Οι αλγόριθμοί μας υλοποιούν σημασιολογία ταξινόμησης χωρίς φιλτράρισμα του πλήρους συνόλου των LCA. Γι΄ αυτό, δεν επιβαρύνονται από τα μειονεκτήματα (χαμηλή ακρίβεια και πληρότητα) παλαιότερων προσεγγίσεων που περιορίζονται σε ένα προκαθορισμένο, δομικά εξαρτημένο υποσύνολο LCA (πρόβλημα 3). Τα αποτελέσματα ομαδοποιούνται σε επίπεδα ταξινομημένα σύμφωνα με την εγγύτητα των εμγανίσεων των λέξεων-κλειδιών στα υποδέντρα τους. Τα επίπεδα μπορούν να πειλαμβάνουν οποιοδήποτε αριθμό από αποτελέσματα. Αυτή η κλιμακωτή ταξινόμηση απαλλάσσει το χρήστη από την ανάγκη να προσδιορίσει την τιμή του k μαζί με την ερώτηση όταν ενδιαφέρεται για τα κορυφαία μόνο αποτελέσματα (πρόβλημα 4). Τα αποτελέσματα του κορυφαίου επιπέδου καταδεικνύεται ότι ικανοποιούν απόλυτα το χρήστη στο θέμα της ακρίβειας. Ανάκτηση επόμενων επιπέδων χρειάζεται μόνο αν ενδιαφέρουν ακόμα μεγαλύτερες τιμές όσον αφορά την πληρότητα.

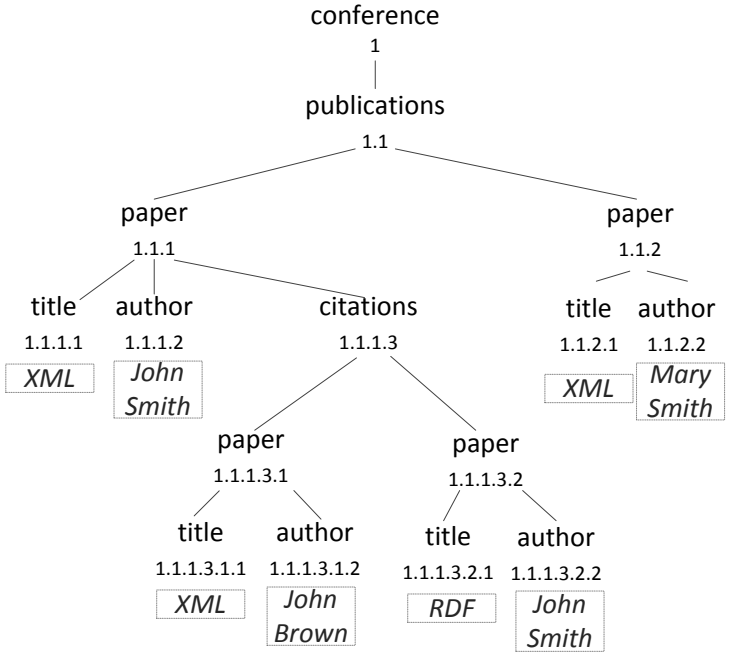
3.2 Μέγεθος χαμηλότερου κοινού προγόνου (LCA size)

3.2.1 Βασικοί ορισμοί

Μοντελοποιούμε τα δεδομένα μορφής XML δεδομένα ως διατεταγμένα δέντρα με ετικέτες. Για τη γλώσσα XML, τα στοιχεία (elements) και τα χαρακτηριστικά (attributes) αναπαριστώνται από κόμβους του δέντρου. Κάθε κόμβος έχει ένα μοναδικό αναγνωριστικό (id), μια ετικέτα (label), που αντιστοιχεί σε ένα στοιχείο ή χαρακτηριστικό, και πιθανώς μια τιμή, που αντιστοιχεί στο περιεχόμενο ενός στοιχείου ή στην τιμή ενός χαρακτηριστικού, αντίστοιχα. Για τα αναγνωριστικά των κόμβων υιοθετούμε το σχήμα κωδικοποίησης Dewey [10], το οποίο αναθέτει αναγνωριστκά σύμφωνα με μια προδιατεταγμένη διάσχιση ενός δέντρου. Το σχήμα κωδικοποίησης Dewey αντικατοπτρίζει εγγενώς τις σχέσεις πρόγονου-απόγονου και πατέρα-παιδιού ανάμεσα στους κόμβους ενός δέντρου, υποστηρίζοντας έτσι φυσικά την επεξεργασία των κόμβων με χρήση στοιβών [19].

Για παράδειγμα, στο δέντρο μια μικρής βιβλιογραφικής δενδρικής βάσης δεδομένων της εικόνας 3.1, οι λέξεις paper, author, title κ.τ.λ. αποτελούν ετικέτες του συνόλου δεδομένων, που αντιστοιχούν είτε σε στοιχεία, είτε σε χαρακτηριστικά. Κάτω από κάθε ετικέτα, σημειώνεται ο κωδικός Dewey του κάθε κόμβου, ο οποίος τον αναγνωρίζει μοναδικά. Ο κωδικός Dewey ενός κόμβου αποτελείται από τον κωδικό Dewey του πατέρα του, ο οποίος χρησιμοποιείται ως πρόθεμα, ακολουθούμενος από έναν αριθμό που αντιστοιχεί στη σχετική θέση του κόμβου αυτού ανάμεσα στα παιδιά του κόμβου-πατέρα. Για παράδειγμα, ο κόμβος author με τιμή Mary Smith έχει Dewey 1.1.2.2, που προκύπτει από τον Dewey του πατέρα, δηλ. 1.1.2 του κόμβου paper, ακολουθούμενος από τον αριθμό 2, μια και ο κόμβος αυτός είναι το 2ο παιδί του 1.1.2. Οι τιμές των στοιχείων και των χαρακτηριστικών εσωκλείονται σε παραλληλόγραμμα (π.χ. XML, John Smith, κ.τ.λ.).

Μία *ερώτηση με λέξεις-κλειδιά* Q ορίζεται ως ένα σύνολο από λέξεις-κλειδιά: $Q = \{k_1, ..., k_n\}$. Μία λέξη-κλειδί k μπορεί να εμφανίζεται σε μια ετικέτα ή σε μια τιμή ενός κόμβου n, στην οποία περίπτωση λέμε ότι ο κόμβος n αποτελεί ένα *στιγμιότυπο* της λέξης-κλειδιού k. Αφού ένας κόμβος μπορεί να περιέχει πολλούς όρους στην ετικέτα και την τιμή του, συνακόλουθα μπορεί να είναι στιγμιότυπο περισσότερων της μίας



Σχήμα 3.1: Παράδειγμα δέντρου δεδομένων

λέξεων-κλειδιών.

Το *ελάχιστο συνδετικό υποδέντρο* (MCT), M_S, ενός συνόλου S κόμβων ενός δέντρου δεδομένων D είναι το ελάχιστο υποδέντρο M_S του D που περιέχει όλους τους κόμβους του S. Η ρίζα του M_S είναι ο *χαμηλότερος κοινός πρόγονος* (lowest common ancestor - LCA) των κόμβων που ανήκουν στο S, και εκφράζεται ως $lca(S)$. Το *μέγεθος* του M_S ορίζεται ως το πλήθος των ακμών του. Έστω I ένα σύνολο από στιγμιότυπα των λέξεων-κλειδιών που περιέχονται στην ερώτηση Q. Αν το I περιέχει ένα στιγμιότυπο για κάθε λέξη-κλειδί στο, τότε λέμε ότι το I είναι ένα *στιγμίοτυπο της ερώτησης*, για το Q. Ο χαμηλότερος κοινός πρόγονος (LCA) του I καλείται επίσης καταχρηστικά χαμηλότερος κονός πρόγονος της ερώτησης Q. Ο χαμηλότερος κοινός πρόγονος ενός υποσυνόλου των λέξεων-κλειδιών της ερώτησης Q ονομάζεται *μερικός χαμηλότερος κοινός πρόγονος (partial LCA)* του Q. Εφεξής, όταν αναφερόμαστε στην έννοια του χαμηλότερου κοινού προγόνου, θα χρησιμοποιούμε το αγγλικό αρκτικόλεξο LCA.

Εισάγουμε τώρα την έννοια του μεγέθους ενός LCA. Έστω I και I' δύο διαφορετικά, αλλά όχι απαραίτητα ξένα μεταξύ τους, στιγμιότυπα μιας ερώτησης Q σε ένα δέντρο D. Προφανώς, τα ελάχιστα συνδετικά υποδέντρα τους M_I και $M_{I'}$ μπορεί να μοιράζονται κοινή ρίζα, τον LCA l.

Ορισμός 3.1. Μέγεθος χαμηλότερου κοινού προγόνου (LCA) *Δεδομένου ενός δέντρου δεδομένων, το μέγεθος του LCA l μιας ερώτησης Q ορίζεται ως το μέγεθος του μικρότερου από τα ελάχιστα συνδετικά υποδέντρα των στιγμιοτύπων της ερώτησης Q, που έχουν ρίζα τον κόμβο l.*

Για παράδειγμα, στο δέντρο της εικόνας 3.1, το μέγεθος του LCA 1.1.1.3 της ερώτησης $Q = \{XML, John, Smith\}$ είναι 4, δεδομένου ότι υπάρχουν ακριβώς δύο ελάχιστα συνδετικά υποδέντρα των στιγμιοτύπων του Q με ρίζα τον κόμβο 1.1.1.3 (το ένα με στιγμιότυπο για το `John`, τον κόμβο 1.1.1.3.1.2 ενώ το άλλο τον κόμβο

1.1.1.3.2.2) και μεγέθη 5 και 4 αντίστοιχα.

3.2.2 Σημασιολογία TLCA (συμπαγούς χαμηλότερου κοινού προγόνου)

Εισάγουμε τώρα τη σημασιολογία ταξινόμησης *συμπαγούς χαμηλότερου κοινού προγόνου* για ερωτήσεις με λέξεις-κλειδιά, που βασίζεται στην έννοια του μεγέθους του χαμηλότερου κοινού προγόνου. Στη συνέχεια, θα αναφερόμαστε στη σημασιολογία μας, ως TLCA σημασιολογία, δηλ. tight LCA. Ό όρος 'συμπαγής' χρησιμοποιείται για να εκφράσει το γεγονός ότι όσο μικρότερο είναι το μέγεθος ενός LCA τόσο πιο κοντά συνδέονται μεταξύ τους τα στιγμιότυπα των λέξεων-κλειδιών της ερώτησης, οπότε και πιο 'συμπαγής' ο συνδυασμός τους.

Σύμφωνα με την TLCA σημασιολογία, η απάντηση σε μια ερώτηση Q που τίθεται σε ένα δέντρο D είναι το σύνολο όλων των LCA της ερώτησης Q στο D , ταξινομημένο σε αύξουσα σειρά των μεγεθών των LCA. Πιο συγκεκριμένα, ένα αποτέλεσμα μιας ερώτησης Q σε ένα δέντρο D ορίζεται ως το ζεύγος (l, s) ενός LCA l του Q και του αντίστοιχου μεγέθους του s. Η απάντηση A του Q στο D ορίζεται ως η διατεταγμένη λίστα του πλήρους συνόλου των αποτελεσμάτων του Q στο D με βάση το μέγεθός τους: $A = [(l_1, s_1), (l_2, s_2), \ldots],\ s_i \leq s_j,\ i < j$. Αν δύο αποτελέσματα έχουν το ίδιο μέγεθος, η σχετική τους θέση στην απάντηση δεν έχει σημασία. Για παράδειγμα, η απάντηση της ερώτησης {`XML, John, Smith`} στο δέντρο της εικόνας 3.1 είναι $A =$ [(1.1.1, 2), (1.1.1.3, 4), (1.1, 4)].

Σύμφωνα με όσα έχουν συζητηθεί στις προηγούμενες παραγράφους, η αποτίμηση ερωτήσεων λέξεων-κλειδιών σύμφωνα με τη συμασιολογία συμπαγούς LCA απαιτεί α) την εξέταση όλων των δυνατών στιγμιοτύπων μιας ερώτησης, β) τον υπολογισμό του ελάχιστου συνδετικού υποδένδρου για καθένα απ΄ αυτά, γ) την ομαδοποίηση ανά LCA και δ) τον επιλογή του μικρότερου μεγέθους από τα μεγέθη των ελάχιστων συνδετικών υποδένδρων ανά LCA. Ας δούμε, όμως, κάποιες ιδιότητες του χαμηλότερου κοινού προγόνου και του μεγέθους LCA που μπορούν να αξιοποιηθούν για να απλοποιηθεί η διαδικασία αποτίμησης.

Ιδιότητα 3.1. *(**Προσεταιριστική ιδιότητα χαμηλότερου κοινού προγόνου**) Έστω I ένα στιγμιότυπο της ερώτησης Q, $l = lca(I)$ και $I_1, \ldots, I_n$ υποσύνολα του I τέτοια ώστε $\bigcup_1^n I_i = I$. Έστω επίσης $p_i = lca(I_i),\ i = 1..n$, μερικοί LCA του Q. Τότε, $l = lca(\{p_1, \ldots, p_n\})$.*

Δύο δέντρα T και T' καλούνται *ξένα* μεταξύ τους αν δε μοιράζονται καμία ακμή και κανέναν κόμβο εκτός ίσως από τη ρίζα τους.

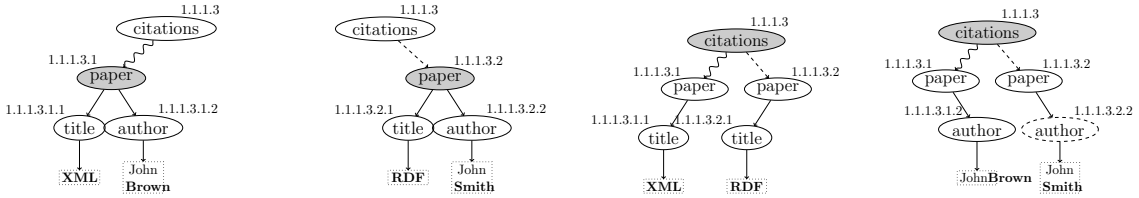
Ιδιότητα 3.2. *Δύο δέντρα T και T', με κοινή ρίζα τον κόμβο r, είναι ξένα μεταξύ τους αν και μόνο αν $E_r \cap E'_r = \emptyset$, όπου E_r (E'_r αντίστοιχα) είναι το σύνολο των προσκείμενων ακμών στον κόμβο r στο T (T' αντίστοιχα).*

Ιδιότητα 3.3. *Έστω T είναι ένα δέντρο με ρίζα r και $T_1, ..., T_n$ ξένα υποδέντρα του T με ρίζα επίσης τον κόμβο r, τα οποία όλα μαζί περιέχουν όλες τις ακμές του T. Τότε, το μέγεθος του T ισοδυναμεί με το άθροισμα των μεγεθών των $T_1, ..., T_n$, δηλ. $size(T) = \sum_1^n size(T_i)$.*

Το ελάχιστο συνδετικό υποδέντρο ενός συνόλου στιγμιοτύπων λέξεων-κλειδιών που έχει ρίζα το μερικό LCA p ενός στιγμιοτύπου ερώτησης I, μαζί με το μονοπάτι

που συνδέει το μερικό LCA p με κάποιον πρόγονό του n ορίζουν ένα *υποδέντρο μερικού LCA* του I με ρίζα τον κόμβο n. Ως συνέπεια της Ιδιότητας 3.3, το μέγεθος ενός LCA l μπορεί να υπολογιστεί ως το άθροισμα των μεγεθών ξένων υποδέντρων μερικών LCA με ρίζα τον κόμβο l.

Ας πάρουμε, για παράδειγμα την ερώτηση $Q = \{$`XML`, `Brown`, `RDF`, `Smith`$\}$ και τον LCA 1.1.1.3 του στιγμιοτύπου της $I = \{$1.1.1.3.1.1, 1.1.1.3.1.2, 1.1.1.3.2.1, 1.1.1.3.2.2$\}$ στην Εικόνα 3.1. Η Εικόνα 3.2 απεικονίζει τέσσερα υποδέντρα μερικών LCA του I με ρίζα τον κόμβο 1.1.1.3. Οι σκιασμένοι κόμβοι καταδεικνύουν τους μερικούς LCA. Με βάση την Ιδιότητα 3.1, ισχύει ότι 1.1.1.3 = lca($\{$1.1.1.3.1, 1.1.1.3.2$\}$) = λca($\{$1.1.1.3, 1.1.1.3$\}$). Παρόλο που και οι δύο συνδυασμοί μερικών LCA παράγουν τον LCA 1.1.1.3, τα υποδέντρα (γ′) και (δ′) δεν μπορούν να χρησιμοποιηθούν στον υπολογισμό του μεγέθους αυτού του LCA, καθώς δεν είναι ξένα μεταξύ τους. Αντίθετα, τα υποδέντρα (α′) και (β′) είναι ξένα μεταξύ τους, σύμφωνα με την Ιδιότητα 3.2, και κατά συνέπεια το άθροισμα των μεγεθών τους δίνει το σωστό μέγεθος για τον LCA 1.1.1.3, που είναι 6 (Ιδιότητα 3.3).



(α′) XML Brown (β′) RDF Smith (γ′) XML RDF (δ′) Brown Smith

Σχήμα 3.2: Υποδέντρα μερικών LCA του LCA 1.1.1.3 για την ερώτηση $\{$`XML,Brown,RDF,Smith`$\}$

3.3 Αποτίμηση ερωτήσεων με λέξεις-κλειδιά με χρήση δικτυωτού (lattice)

Ο υπολογισμός του πλήρους συνόλου των LCA για μια δεδομένη ερώτηση είναι εκθετικός στον αριθμό των λέξεων - κλειδιών της ερώτησης. Συγκεκριμένα, για μια ερώτηση με n λέξεις - κλειδιά, αν το μήκος της ανάστροφης λίστας μιας λέξης - κλειδιού είναι $|L|$, τότε ο υπολογισμός είναι της τάξης $O(|L|^n)$. Η αποτίμηση επιπλέον του μεγέθους των MCT κάθε πιθανού συνδυασμού στιγμιοτύπων λέξεων - κλειδιών μέσα στο δέντρο, δυσχεραίνει ακόμα τον υπολογισμό όπως έχει αποδειχθεί στην εργασία [21]. Έτσι, παρόλο που αναγνωρίζεται η αξία της συνεισφοράς του μεγέθους στην αξιολόγηση μιας απάντησης, στη βιβλιογραφία η έννοια του μεγέθους αξιοποιείται μόνο εκφυλισμένη, π.χ., ως η απόσταση μεταξύ του LCA και του στιγμιοτύπου κάθε μιας λέξης - κλειδιού [19] , ή ως η απόσταση ανά δύο των στιγμιοτύπων των λέξεων - κλειδιών [13].

Ο χαμηλότερος κοινός πρόγονος των στιγμιοτύπων λέξεων-κλειδιών ενδέχεται να αποτελεί τη ρίζα πολλών ελάχιστων συνδετικών δέντρων (MCT) στιγμιοτύπων λέξεων-κλειδιών μιας ερώτησης. Όπως αναφέρθηκε στην προηγούμενη ενότητα, ονομάζουμε *μερικό* χαμηλότερο κοινό πρόγονο (partial LCA) έναν κόμβο που αποτελεί LCA για υποσύνολο των λέξεων - κλειδιών μιας ερώτησης. Το μέγεθος μερικού LCA ορίζεται σε αναλογία με το μέγεθος LCA. Οι LCA που είναι χαμηλότεροι κοινοί πρόγονοι στιγμιοτύποων του ίδιου υποσυνόλου λέξεων - κλειδιών καλούνται *συγκρίσιμοι*. Το

μέγεθος ενός LCA είναι το άθροισμα των μεγεθών των μερικών LCA που αντιστοιχούν στα υποσύνολα ενός διαμερισμού του συνόλου των λέξεων - κλειδιών της ερώτησης. Τα μεγέθη αυτά προσαυξάνονται κατά το μέγεθος του μονοπατιού που συνδέει τον LCA με τους μερικούς LCA. Απαραίτητη προϋπόθεση, για τον υπολογισμό του μεγέθους ενός LCA κατ΄ αυτόν τον τρόπο, είναι τα υποδέντρα των μερικών LCA που αθροίζονται, να είναι ξένα μεταξύ τους. Κατά συνέπεια, ο υπολογισμός του μεγέθους ενός LCA ανάγεται στον υπολογισμό των μερικών LCA που είναι απόγονοί του. Όλοι οι δυνατοί συνδυασμοί υποσυνόλων των λέξεων - κλειδιών μιας ερώτησης δημιουργούν όλους τους δυνατούς συνδυασμούς αθροισμάτων που μπορούν να προκύψουν κατά τον υπολογισμό του μεγέθους ενός LCA. Από τα μερικά αθροίσματα που συνθέτουν το τελικό μέγεθος, λαμβάνεται υπόψιν το ελάχιστο μεταξύ των συγκρίσιμων LCA.

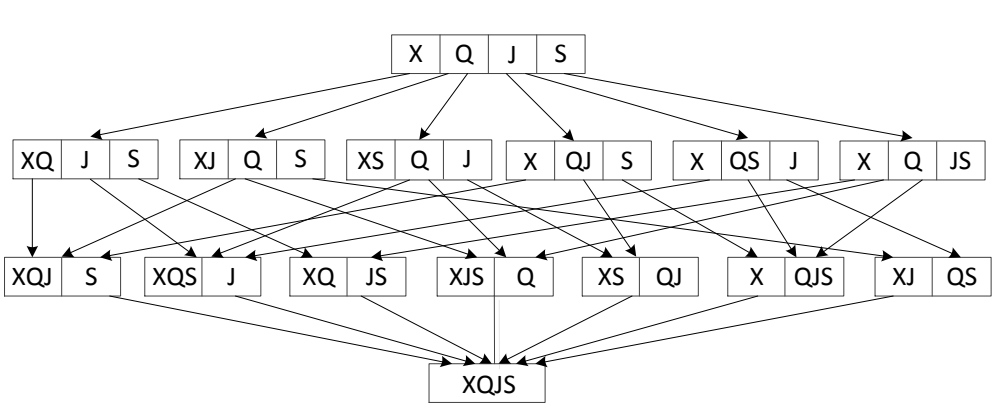
Γύρω από αυτήν την κεντρική ιδέα, σχεδιάστηκε ο αλγόριθμος LCAsz ο οποίος χρησιμοποιεί στοίβες για την επεξεργασία των δενδρικών δεδομένων. Ο αλγόριθμος, ξεκινώντας από τα στιγμιότυπα των ανάστροφων λιστών των λέξεων - κλειδιών μιας ερώτησης, συνθέτει μερικούς και τελικά ολικούς LCA υπολογίζοντας ταυτόχρονα τα μεγέθη τους. Ο LCAsz εκτελείται αξιοποιώντας το δικτυωτό της μερικής διάταξης των διαμερισμών του συνόλου των λέξεων - κλειδιών μιας ερώτησης, όπως απεικονίζεται στο Σχήμα 3.3. Η χρονική πολυπλοκότητα υπολογίζεται πως είναι, μεν, εκθετική στον αριθμό των λέξεων - κλειδιών (δηλ. εξαρτάται από το μέγεθος του δικτυωτού), αλλά είναι γραμμική σε σχέση με το μέγεθος της εισόδου. Η τελευταία ιδιότητα καθιστά τον LCAsz ταχύτατο, ακόμα και για πολύ μεγάλα σύνολα δεδομένων. Επίσης ο LCAsz επιδεικνύει πολύ καλές επιδόσεις, ακόμα και σε πολύπλοκα στη δομή και μεγάλου βάθους δενδρικά δεδομένα, όταν αντίστοιχοι αλγόριθμοι αδυνατούν να ολοκληρώσουν την εκτέλεσή τους. Οι υπολογισμοί αυτοί επιβεβαιώνονται από εκτενή πειραματική μελέτη.

3.3.1 Αλγόριθμος LCAsz

Ο αλγόριθμος LCAsz είναι ένας αλγόριθμος που χρησιμοποιεί στοίβες και επιστρέφει όλους τους LCA σε σειρά μεγέθους. Η είσοδος του αλγορίθμου είναι είναι οι ανεστραμμένες λίστες των λέξεων-κλειδιών ενός δένδρου και μια ερώτηση με λέξεις-κλειδιά. Η έξοδος είναι η απάντηση στην ερώτηση σύμφωνα με τη σημασιολογία TLCA, δηλ. η ταξινομημένη λίστα των LCA μαζί με τα μεγέθη τους: $A = [(l_1, s_1), (l_2, s_2), ..]$

Με άμεση εφαρμογή του Ορισμού 3.1, ο υπολογισμός του μεγέθους ενός LCA προϋποθέτει τον υπολογισμό όλων των ελάχιστων συνδετικών δέντρων που έχουν ρίζα αυτόν τον LCA μαζί με τα μεγέθη τους, ώστε να επιλεγεί τελικά το μοικρότερο. Ο αλγόριθμος LCAsz, ωστόσο, αποφεύγει την εξαντλητική εξέταση όλων των ελάχιστων συνδετικών υποδέντρων κάθε LCA μιας ερώτησης. Για να το πετύχει αυτό, συνδυάζει τα στιγμιότυπα των λέξεων-κλειδιών αρχικά για να υπολογίσει μερικούς και στη συνέχεια ολικούς LCA, συγκρινοντάς τους στην πορεία της διαδικασίας και αποκλείοντας μερικούς LCA με μεγάλο μέγεθος ανά κόμβο και υποσύνολο λέξεων-κλειδιών, πριν αυτοί καταλήξουν σε ολικούς LCA και τελικό μέγεθος.

Ο αλγόριθμος LCAsz εκτελεί αυτή τη διαδικασία από κάτω προς τα πάνω συνδυάζοντας βήμα, βήμα τους μερικούς LCA στιγμιοτύπων ενός υποσυνόλου S των λέξεων-κλειδιών της ερώτησης, που βρίσκονται χαμηλότερα στο δένδρο δεδομένων, σε (μερικούς ή ολικούς) LCA στιγμιοτύπων ενός υπερσυνόλου του S που βρίσκονται ψηλότερα στο δέντρο. Οι μερικοί LCA προωθούνται προς τα πάνω στους προγόνους τους. Γενικεύοντας, στη συνέχεια της περιγραφής του αλγορίθμου, όταν αναφερόμα-



Σχήμα 3.3: Δικτυωτό (lattice) της μερικής διάταξης των διαμερίσεων των λέξεων-κλειδιών της ερώτησης {`XML, query, John, Smith`}

στε στο μέγεθος ενός μερικού LCA l, θα αναφερόμαστε στο μέγεθός του υποδέντρου του κάτω από ένα συγκεκριμένο κόμβο n, δηλ. στο μέγεθός του προσαυξημένο με το μέγεθος του μονοπατιού που συνδέει τον l με τον πρόγονό του n.

Μ′ αυτήν την έννοια, στη διάρκεια της προώθησης μερικών LCA προς τους προγόνους τους, το μέγεθός τους προσαυξάνεται ανάλογα με τις ακμές του μονοπατιού που τους χωρίζει από τον υπό εξέταση κόμβο (πρόγονο) κάθε φορά. Σε κάθε κόμβο σ′ αυτό το μονοπάτι, το μέγεθος ενός μερικού LCA συγκρίνεται με τους υπόλοιπους *συγκρίσιμους* LCA (δηλ. μερικούς LCA του ίδιου συνόλου λέξεων-κλειδιών) και μόνο το μικρότερο μέγεθος καταχωρείται. Μ′ αυτήν την τεχνική, μόνο ένα μέγεθος αποθηκεύεται για κάθε ομάδα από συγκρίσιμους LCA που έχουν βρεθεί ως αυτό το σημείο εκτέλεσης του αλγορίθμου, δηλ. μέχρι το συγκεκριμένο κόμβο του δέντρου. Για παράδειγμα, στην Εικόνα 3.1, υπάρχουν 6 διαφορετικά υποδέντρα μερικών LCA με κάτω από τον κόμβο 1.1.1 για τις λέξεις-κλειδιά {`John,Smith`} με συνολικά 5 διαφορετικά μεγέθη. Μόνο το μικρότερο απ′ αυτά, όμως, θα επικρατήσει, δηλ. αυτό που αντιστοιχεί στο μερικό LCA 1.1.1.2 και είναι 1, και αυτό είναι που θα προωθηθεί τελικά στους προγόνους του 1.1.1.

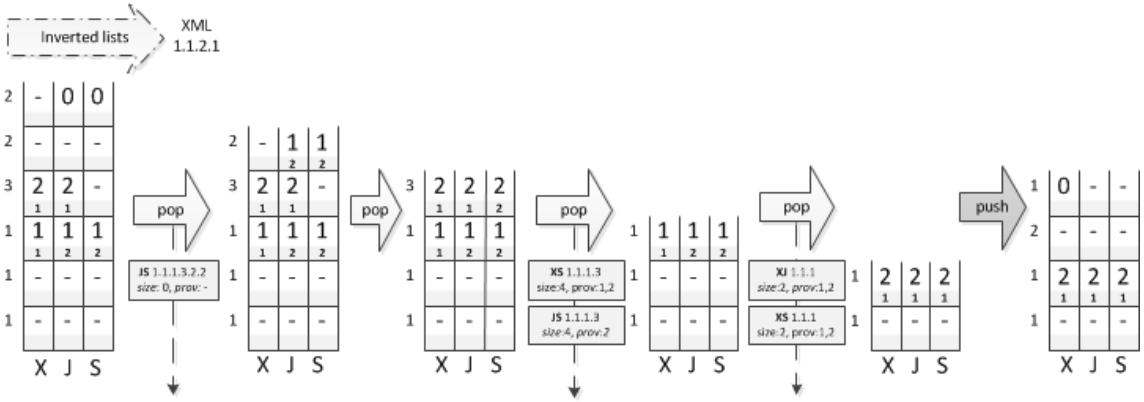
Με βάση τα παραπάνω, το πρόβλημα εντοπισμού όλων των LCA και υπολογισμού των μεγεθών τους ανάγεται στην εξέταση όλων των δυνατών συνδυασμών των λέξεων-κλειδιών κάτω από έναν LCA και αποκλεισμού αυτών με τα μεγαλύτερα μεγέθη. Ο στόχος είναι η αποφυγή του υπολογισμού ενός μερικού LCA και του μεγέθους του όσο χαμηλότερα γίνεται στο δέντρο δεδομένων, με βάση του μερικούς LCA που είναι απόγονοί του, δηλ. βρίσκονται ακόμα χαμηλότερα στο δέντρο. Μ′ αυτήν την τεχνική, ο αριθμός των ελάχιστων συνδετικών υποδέντρων που χρειάζεται να υπολογιστούν μειώνεται δραματικά.

3.3.1.1 Δικτυωτό των διαμερίσεων των λέξεων-κλειδιών

Όλοι οι δυνατοί συνδυασμοί ενός συνόλου λέξεων-κλειδιών αντιστοιχεί σε όλες τις δυνατές διαμερίσεις του. Ορίζουμε μια σχέση μερικής διάταξης $\leq$ στις διαμερίσεις ενός συνόλου: $P_1 \leq P_2$ αν και μόνο αν για κάθε σύνολο $s_1 \in P_1$ υπάρχει ένα σύνολο $s_2 \in P_2$ τέτοιο ώστε $s_1 \subseteq s_2$. Αν $P_1 \leq P_2$, τότε το σύνολο P_1 καλείται *λεπτομερέστερο* από το P_2 και το P_2 είναι *ευρύτερο* από το P_1. Η σχέση μερικής διάταξης $\leq$ σχηματίζει ένα δικτυωτό των διαμερίσεων των λέξεων-κλειδιών. Η εικόνα 3.3 απεικονίζει το διάγραμμα Hasse του δικτυωτού των διαμερίσων του συνόλου των λέξεων-κλειδιών

{XML, query, John, Smith}. Σε κάθε επίπεδο του δικτυωτού οι διαμερίσεις έχουν τον ίδιο αριθμό στοιχείων. Κάθε διαμέριση σε ένα επίπεδο παράγεται από την ένωση δύο στοιχείων μιας τουλάχιστον διαμέρισης από το προηγούμενο επίπεδο.

Για τον αλγόριθμο LCAsz μία διαμέριση αναπαριστά έναν LCA και τα στοιχεία που περιέχει αντιστοιχούν στους μερικούς LCA που τον συνθέτουν. Για κάθε διαμέριση, ο αλγόριθμος κατασκευάζει μία στοίβα. Η μοναδική στοίβα του πρώτου επιπέδου, είναι η αρχική στοίβα που αντιστοιχεί στη λεπτεμερέστερη διαμέριση, η οποία περιέχει ένα υποσύνολο για κάθε λέξη-κλειδί. Το τελευταίο επίπεδο επίσης περιέχει μία μόνο στοίβα, που αντιστοιχεί στο σύνολο όλων των λέξεων-κλειδιών. Η τελική στοίβα είναι αυτή, από την οποία προκύπτουν οι πλήρεις LCA, δηλ. τα αποτελέσματα μιας ερώτησης. Κάθε μονοπάτι από την αρχική στοίβα στην τελική, υποδεικνύει ένα μοναδικό τρόπο συνδυασμού των λέξεων-κλειδιών σε μερικούς LCA (στα εσωτερικά επίπεδα) και τελικά σε πλήρεις LCA (στο τελικό επίπεδο).



Σχήμα 3.4: Καταστάσεις της αρχικής στοίβας του δικτυωτού για την ερώτηση {XML, John, Smith} όταν γίνεται η επεξεργασία του στιγμιοτύπου 1.1.2.1 για τη λέξη-κλειδί XML

3.3.1.2 Δομή της στοίβας

Η βασική δομή που χρησιμοποιείται από τον αλγόριθμο LCAsz είναι η στοίβα. Η στοίβα χρησιμοποιείται συχνά σε αλγορίθμους επεξεργασίας δέντρων γιατί διευκολύνει την κατά βάθος διάσχιση. Στην περίπτωση του LCAsz, κάθε στιγμή μία στοίβα περιέχει στοιχεία που αντιστοιχούν σε ένα μονοπάτι του δέντρου και στο οποίο αντιστοιχεί ένας κωδικός Dewey. Συγκεκριμένα, σε κάθε θέση της στοίβας αποθηκεύεται πληροφορία για το στο στοιχείο του δέντρου με Dewey το μέρος του Dewey της στοίβας μέχρι αυτήν η θέση. Για παράδειγμα, αν το κορυφαίο στοιχείο της στοίβας αναφέρεται στον κόμβο με Dewey 1.2.3.4.5, τότε η στοίβα περιέχει πληροφορία για όλους τους κόμβους του μονοπατιού του δέντρου από τη ρίζα μέχρι τον κόμβο αυτό και ενδεικτικά, το τρίτο στοιχείο στη στοίβα αναφέρεται στον κόμβο με κωδικό 1.2.3. Εισαγωγή και εξαγωγή στοιχείου από τη στοίβα ισοδυναμούν με προσθήκη ή αφαίρεση ενός αριθμού από το τέλος του κωδικού Dewey.

Κάθε στοιχείο της στοίβας είναι ένας πίνακας. Τα στοιχεία του πίνακα αντιστοιχούν σε υποσύνολα των λέξεων-κλειδιών μιας ερώτησης, όπως υπαγορεύονται από την αντίστοιχη διαμέριση του δικτυωτού που αντιστοιχεί στη συγκεκριμένη στοίβα. Το στοιχείο του πίνακα που αντιστοιχεί στο υποσύνολο των λέξεων-κλειδιών S περιέχει το μέγεθος s ενός μερικού LCA l του S στον κόμβο που αντιστοιχεί στη θέση αυτή

της στοίβας, εφόσον ένας τέτοιος LCA έχει βρεθεί κάτω στο υποδέντρο αυτού του κόμβου. Αν ένας τέτοιος LCA έχει βρεθεί, στο στοιχείο του πίνακα περιέχει επίσης έναν η περισσότερους αριθμούς που υποδεικνύουν την προέλευση αυτού του μερικού LCA. Οι αριθμοί αυτοί αντιστοιχούν στους σχετικούς αριθμούς των παιδιών του κόμβου αυτού, που αν προστεθούν στον τρέχοντα κωδικό Dewey αναγνωρίζουν το (τα) παιδί (παιδιά) του τρέχοντα κόμβου, απ΄ όπου 'έρχεται' το υποδέντρο του συγκεκριμένου μερικού LCA. Οι αριθμοί αυτοί είναι πολλοί, αν ο τρέχοντας κόμβος είναι ο μερικός LCA. Σε αντίθετη περίπτωση, η προέλευση του μερικού LCA καταδεικνύεται από ένα μοναδικό αριθμό.

3.3.1.3 Περιγραφή του αλγορίθμου

Ο αλγόριθμος LCAsz εφαρμόζει τις ιδέες που περιγράφηκαν στις προηγούμενες παραγράφους. Το κύριο μέρος του αλγορίθμου είναι ο επαναληπτικός βρόχος των γραμμών 3-13, κατά τον οποίο ο LCAsz διατρέχει τις ανεστραμμένες λίστες των λέξεων-κλειδιών της ερώτησης και επιλέγει προς επεξεργασία το στιγμιότυπα των λέξεων-κλειδιών με τη σειρά εμφάνισής τους στο δέντρο δεδομένων. Η επεξεργασία ενός στιγμιοτύπου εμπεριέχει την εισαγωγή του σε όλες τις στοίβες που περιέχουν την αντίστοιχη λέξη-κλειδί ως σύνολο ενός στοιχείου. Η ίδια λογική ακολουθείται και για τους μερικούς LCA που θα παραχθούν αργότερα. Πρακτικά, κάθε στιγμιότυπο λέξης-κλειδιού ο ΛΆσζ το χειρίζεται ομοιόμορφα σαν ένα μερικό LCA μίας μόνο λέξης-κλειδιού (γραμμή 5).

Πατατηρώντας το δικτυωτό της Εικόνας 3.3, μπορεί κανείς να παρατηρήσει ότι οι περισσότερες διαμερίσεις διαθέτουν περισσότερες από μία εισερχόμενες ακμές. Αυτές οι ακμές, για τον αλγόριθμο, υποδεικνύουν πιθανώς πολλαπλές ενημερώσεις μιας στοίβας με στοιχεία από περισσότερες της μίας στοίβας του προηγούμενου επιπεδου για έναν κόμβο του δέντρου. Ο LCAsz, ωστόσο, αποφεύγει αυτόν τον πλεονασμό αξιποιώντας μία λίστα μερικών LCA για κάθε επίπεδο του δικτυωτού. Οι μερικοί LCA περνούν απ΄ αυτές τις λίστες πριν προωθηθούν στα επόμενα επίπεδα του δικτυωτού (γραμμή 6 του αλγορίθμου και γραμμή 16 της διαδικασίας ποπ). Η χρήση αυτή των λιστών των μερικών LCA φιλτράρει τους παραγόμενους μερικούς LCA και συγχρονίζει την προώθησή τους. Ως εκ τούτου, οι εισαγωγές σε μια στοίβα ελαχιστοποιούνται. Όταν ένας μερικός LCA προκύψει σε ένα επίπεδο του δικτυωτού, πρόστίθεται στη λίστα μερικών LCA του επόμενου επιπεδου (γραμμές 15-19). Αν η λίστα περιέχει ήδη έναν συγκρίσιμο μερικό LCA για τον ίδιο κόμβο του δέντρου (ίδιο κωδικό Dewey), μόνο το μικρότερο μέγεθος αποθηκεύεται στη λίστα (γραμμές 18-19). Για παράδειγμα, στο δέντρο της Εικόνας 3.1, υπάρχουν πολλά υποδέντρα μερικών LCA στο υποδέντρο του κόμβου 1.1.1 για το σύνολο λέξεων-κλειδιών {`XML, John, Smith`}. Παρ΄ όλ΄ αυτά, στη λίστα μερικών LCA για τον κωδικό Dewey 1.1.1 μόνο ένας μερικός LCA με μέγεθος 2 αποθηκεύεται για να προωθηθεί στα επόμενα επίπεδα του δικτυωτού. Αυτό το μέγεθος αντιστοιχεί στο ελάχιστο συνδετικό υποδέντρο των στιγμιοτύπων 1.1.1.1 (για το `XML`) και 1.1.1.2 (για τα `John` και `Smith`).

Όλοι οι μερικοί LCA αντλούνται σειριακά από τις λίστες μερικών LCA (γραμμή 7). Στη συνέχεια, εισάγονται στις στοίβες του αντίστοιχου επιπέδου του δικτυωτού, οι οποίες αντιπροσωπεύουν διαμερίσεις του συνόλου των λέξεων-κλειδιών που περιέχουν το υποσύνολο λέξεων-κλειδιών των συγκεκριμένων LCA (γραμμές 9-10). Όταν όλα τα στιγμιότυπα των λέξεων-κλειδιών έχουν εξεταστεί, οι στοίβες αδειάζουν κατά σειρά των επιπέδων του δικτυωτού (γραμμή 14): αρχικά, σε κάθε επίπεδο γίνεται επαξεργασία των εναπομείναντων LCA μέσα στη λίστα μερικών LCA του επιπέδου (γραμμές 27-31), και στη συνέχεια η διαδικασία pop καλείται για όλα τα στοιχεία των στοιβών του

Algorithm 1: LCAsz

```
1  LCAsz(k_1, ..., k_n: keyword query, invL: inverted lists)
2      buildLattice()
3      while currentNode = getNextNodeFromInvertedLists() do
4          coarsenessLevel = 1, size=0, provenance=∅
5          pLCA = newPartialLCA(currentNode.ID, currentNode.kwSubset, size,
           provenance)
6          addPartialLCA(1, pLCA)
7          while partialLCAlists contains partialLCAs for coarsenessLevel do
8              while partialLCA = partialLCAlists(coarsenessLevel).next() do
9                  for every stack of coarsenessLevel containing
                   partialLCA.kwSubset do
10                     push(stack, partialLCA.ID, partialLCA.kwSubset,
                       partialLCA.size)
11                 if coarsenessLevel < n then
12                     addPartialLCA(coarsenessLevel+1, partialLCA)
13             coarsenessLevel++
14     emptyStacks()
15 addPartialLCA(coarsenessLevel, partialLCA)
16     if (partialLCA.ID, partialLCA.kwSubset) not in
       partialLCAlists(coarsenessLevel) then
17         insert partialLCA into partialLCAlists(cL)
18     else if current (partialLCA.ID, partialLCA.kwSubset) entry size <
       partialLCA.size then
19         replace with partialLCA
20 push(stack, nodeID, kwSubset, size)
21     while stack.dewey not ancestor of nodeID do
22         pop(stack)
23     while stack.dewey ≠ nodeID do
24         addEmptyRow(stack) /* updating stack.dewey until it is
               equal to nodeID */
25     replaceSizeIfSmallerWith(stack.topRow, kwSubsetColumn, size)
26 emptyStacks()
27     foreach coarsenessLevel do
28         if partialLCAlists(coarsenessLevel) is not empty then
29             while partialLCA = partialLCAlists(coarsenessLevel).next() do
30                 for every stack of coarsenessLevel containing
                   partialLCA.kwSubset do
31                     push(stack, partialLCA.ID, partialLCA.kwSubset,
                       partialLCA.size)
32         foreach stack of coarsenessLevel do
33             repeat
34                 pop(stack)
35             until top entry contains only propagated or empty elements and
               the other entries are empty;
```

επιπέδου μέχρι αυτές να αδειάσουν (γραμμές 32-35).

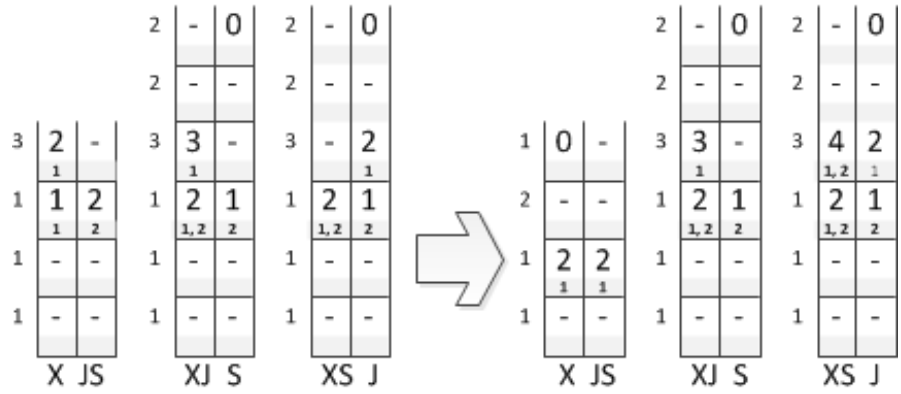
Η εισαγωγή ενός μερικού LCA σε μια στοίβα (γραμμές 20-25) μπορεί να γίνει μόνο αν αυτός ο LCA είναι παιδί ή ο ίδιος ο κόμβος που αντιστοιχεί στην κορυφή της στοίβας (δηλ. έχει ως αναγνωριστικό τον τρέχοντα τον κωδικό Dewey της στοίβας). Γι΄ αυτό το λόγο, τα στοιχεία της στοίβας που είναι εκτός του μονοπατού του LCA αυτού μέχρι τη ρίζα, εξάγονται από τη στοίβα μέχρι στην κορυφή της να μείνει κάποιος πρόγονός του.Η Εικόνα 3.4 απεικονίζει μια σειρά από ενέργειες εισαγωγής και εξαγωγής στοιχείων στην αρχική στοίβα του δικτυωτού για την ερώτηση {`XML`, `John`, `Smith`} όταν ο LCAsz πρόκειται να επεξεργαστεί το στιγμιότυπο 1.1.2.1 για τη λέξη-κλειδί `XML`. Υπό ορισμένες προϋποθέσεις, οι ενέργειες εξαγωγής στοιχείων μπορούν να προκαλέσουν τη δημιουργία νέων (μερικών) LCA. Η διαδικασία αυτή αναλύεται στην επόμενη παράγραφο. Μετά την απομάκρυνση από τη στοίβα των κόμβων που δεν είναι πρόγονοι του μερικού LCA που είναι έτοιμος να μπει στη στοίβα, ο αλγόριθμος προετοιμάζει τη στοίβα να τον δεχτεί, προσθέτοντας όσα κενά στοιχεία χρειάζεται, δηλ. ένα για κάθε πρόγονο του συγκεκριμένου LCA, ώστε το κορυφαίο στοιχείο να είναι ο κόμβος του ίδιου του LCA (γραμμές 23-24). Αυτά είναι τα στοιχεία που αντιστοιχούν στους κόμβους 1.1.2 και 1.1.2.1 στην τελευταία κατάσταση της στοίβας της Εικόνας 3.4. Τέλος, ο LCAsz αντικαθιστά το μέγεθος που αντιστοιχεί στο υποσύνολο των λέξεων-κλειδιών του LCA που είναι έτοιμος να εισαχθεί στη στοίβα, με το μέγεθος του LCA αυτού (γραμμή 25). Η αποθήκευση του νέου μεγέθους γίνεται μόνο αν δεν υπάρχει καταχωρημένο μέγεθος για το συγκεκριμένο υποσύνολο λέξεων-κλειδιών ή αν το μέγεθος που είναι καταγεγραμμένο είναι μεγαλύτερο από το καινούργιο. Η αντικατάσταση του μεγέθους επιτρέπει στον LCAsz να χειρίζεται χωρίς επιπλέον ενέργειες εισαγωγής κι εξαγωγής την εμφάνιση διαφορετικών στιγμιοτύπων των λέξεων-κλειδιών στον ίδιο κόμβο (π.χ. τα στιγμιότυπα για τα `John` και `Smith` στο κόμβο 1.1.1.3.2.2 της πρώτης κατάστασης της στοίβας της Εικόνας 3.4), όπως επίσης και την εμφάνιση λέξεων-κλειδιών σε εσωτερικούς κόμβους του δέντρου.

Η διαδικασία εξαγωγής pop, που καλείται στη γραμμή 22 του LCAsz, είναι η σημαντικότερη του αλγορίθμου καθώς σάυτήν σχηματίζονται οι μερικοί LCA και απ΄ αυτήν επιστρέφονται τα αποτελέσματα, δηλ. οι πλήρεις LCA. Το πρώτο βήμα αυτής της διαδικασίας (γραμμή 3) αφαιρεί από τη στοίβα το κορυφαίο στοιχείο, αφήνοντας τη στοίβα με τον πατέρα του ως νέο κορυφαίο στοιχείο. Αν η στοίβα αποτελείται από στοιχεία των οποίων οι πίνακες έχουν με τη σειρά τους ένα μόνο στοιχείο, τότε πρόκειται για την τελική στοίβα του τελευταίου επιπέδου του δικτυωτού, η οποία παράγει και τα αποτελέσματα της ερώτησης. Η διαδικασία $addResult()$ ταξινομεί τους παραγόμενους πλήρεις LCA και επιστρέφει την απάντηση του αλγορίθμου, όταν όλα τα αποτελέσματα έχουν υπολογιστεί. Στις περιπτώσεις των υπόλοιπων στοιβών, ο LCAsz ελέγχει για πιθανούς νέους LCA, προερχόμενους από το κορυφαίο στοιχείο που εξάγεται από τη στοίβα (γραμμές 6-15) και ενημερώνει κατάλληλα τον πίνακα με τα μεγέθη του πατέρα του, που παίρνει τη θέση του κορυφαίου στοιχείου στη στοίβα, με βάση τα μεγέθη του εξαγόμενου στοιχείου (γραμμές 16-20).

Για όλα τα ζευγάρια υποσυνόλων λέξεων-κλειδιών του εξαγόμενου στοιχείου από τη στοίβα, για τα οποία υπάρχει μέγεθος LCA και δεν έχουν κοινή προέλευση, σύμφωνα με τις Ιδιότητες 3.2 και 3.3, ο LCAsz δημιουργεί ένα νέο μερικό LCA. Στη συνέχεια, προσθέτει το νέο LCA στη λίστα μερικών LCA του επόμενου επιπέδου του δικτυωτού (γραμμές 13-15). Τα μεγέθη των μερικών LCA του εξαγόμενου στοιχείου αυξάνονται κατά 1, πριν προωθηθούν στον πατέρα του αξαγόμενου στοιχείου. Αν τα επαυξημένα μεγέθη είναι μικρότερα από τα αντίστοιχα στο στοιχείο πατέρα του εξαγόμενου στοι-

Procedure pop

```
1  pop(stack)
2      cols = stack.columns
       /* number of kwSubsets in the partition of the stack */
3      popped = stack.pop()
4      if cols = 1 then
5          addResult(stack.dewey,popped[0].size)
       /* Produce new LCAs from two partial LCAs */
6      if cols > 1 then
7          for i=0 to cols do
8              for j=i to cols do
9                  if popped[i] and popped[j] contain sizes AND
                   popped[i].provenance ∩ popped[j].provenance = ∅ then
10                     newKwSubset = popped[i].kwSubset ∪ popped[j].kwSubset
11                     newProvenance = popped[i].provenance ∪
                       popped[j].provenance
12                     newSize = popped[i].size+popped[j].size
13                     pLCA = newPartialLCA(stack.dewey, newKwSubset,
                       newSize, newProvenance)
14                     if cardinalityOf(newKwSubset) = stack.coarsenessLevel+1
                       then
15                         addPartialLCA(stack.coarsenessLevel+1, pLCA)
       /* Update ancestor (i.e., new top entry) with sizes from
          popped entry */
16     if stack is not empty and cols > 1 then
17         for i=0 to cols do
18             if popped[i].size+1 < stack.topRow[i].size then
19                 stack.topRow[i].size = popped[i].size+1
20                 stack.topRow[i].provenance = {lastStep(stack.dewey)}
21     removeLastDeweyStep(stack.dewey)
```



Σχήμα 3.5: Οι στοίβες του 2ου επιπεδου του δικτυωτού για την ερώτηση {`XML, John, Smith`} πριν και μετά την επεξεργασία του στιγμιοτύπου 1.1.2.1 για τη λέξη-κλειδί XML

χείου, τότε ο LCAsz αντικαθιστά με τα νέα μεγέθη μερικών LCA αυτά του πατέρα και ορίζει ως προέλευση των αντίστοιχων μερικών LCA στον πατέρα, το τελευταίο κομμάτι του κωδικού Dewey του παιδιού που αφαιρείται από τη στοίβα (γραμμές 18-20). Ενδεικτικά, στο παράδειγμα της Εικόνας 3.4, στην πρώτη ενέργεια εξαγωγής τα μεγέθη του κόμβου 1.1.1.3.2 για τα `John` και `Smith` αντικαθίστανται από τα αντίστοιχα του εξαγόμενου παιδιού 1.1.1.3.2.2 και οι δείκτες προέλευσης τίθενται στην τικμή 2, ως ένδειξη ότι προέρχονται από το παιδί 2 του κόμβου 1.1.1.3.2. Έτσι διασφαλίζεται ότι ένας κόμβος αποκτά τελικά τα μικρότερα μεγέθη μερικών LCA από όλα τα παιδιά του και αναδρομικά από ολόκληρο το υποδέντρο του για κάθε υποσύνολο λέξεων-κλειδιών της διαμέρισης της στοίβας όπου φιλοξενείται.

Η απαίτηση για μη κοινούς δείκτες προέλευσης (γραμμή 9) εκφράζει την Ιδιότητα 3.2, και επιτρέπει μόνο ξένα μεταξύ τους υποδέντρα μερικών LCA να παράγουν νέους (μερικούς) LCA. Ο δείκτης προέλευσης σε ένα στοιχείο μιας στοίβας για κάποιο υποσύνολο λέξεων-κλειδιών μπορεί να οριστεί με δύο τρόπους. Η πρώτη περίπτωση είναι όταν το μέγεθος ενός μερικού LCA προωθείται στον πατέρα από κάποιο παιδί, οπότε τίθεται στο τελευταίο κομμάτι του κωδικού]ενΔεωεψ του παιδιού (γραμμή 20), όπως είδαμε και στην προηγούμενη παράγρταφο. Αυτός ο αριθμός, ουσιαστικά προσδιορίζει το υποδέντρο του κόμβου που συνεισφέρει το συγκεκριμένο μέγεθος. Η δεύτερη περίπτωση είναι η δημιουργία ενός μερικού LCA σε έναν κόμβο από δύο μερικούς LCA διαφορετικών υποσυνόλων λέξεων-κλειδιών. Τότε ο δείκτης προέλευσης τίθεται να είναι η ένωση των δεικτών προέλευσης των συνδυαζόμενων μερικών LCA (γραμμή 11). Ένα παράδειγμα φαίνεται στην Εικόνα 3.4, όπου ο δείκτης προέλευσης του μερικού LCA 1.1.1.3 των {`XML, Smith`}, που παράγεται στην τρίτη ενέργεια pop, εισάγεται στη στοίβα {`XS, J`} της Εικόνας 3.13β′.

3.3.2 Ανάλυση πολυπλοκότητας LSAsz

Ο αλγόριθμος LCAsz επεξεργάζεται σειριακά τα στιγμιότυπα των λέξεων-κλειδιών από τις αντίστοιχες ανεστραμμένες λίστες. Κάθε στιγμιότυπο εισέρχεται στο δικτυωτό από την αρχική στοίβα του πρώτου επιπέδου και στη συνέχεια μερικοί και πλήρεις LCA δημιουργούνται σταδιακά ακολουθώντας τα μονοπάτια του δικτυωτού (βλ. Εικόνα 3.3). Σε κάθε στοίβα ένα στιγμιότυπο πιθανώς να δημιουργεί κάποια υποδέντρα μερικών LCA. Στη χειρότερη περίπτωση, κάθε στιγμιότυπο εισάγεται μία φορά σε κάθε στοίβα είτε αυτούσιο είτε μέσω κάποιου μερικού LCA. Το πλήθος όλων των διαμερίσων του

συνόλου των λέξεων-κλειδιών $\{w_1, ..., w_k\}$, που είναι ίσο με το πλήθος των στοιβών του δικτυωτού, δίνεται από τον αριθμό Bell B_k:

$$B_k = \sum_{i=1}^{k} S(k, i)$$

όπου $S(k, i)$ είναι ο αριθμός Stirling 2ου τύπου k στοιχείων χωρισμένων σε i υποσύνολα. Ισοδύναμα, για κάθε $i \in [1..k]$, $S(k, i)$ είναι ο αριθμός των διαμερίσεων στο επίπεδο του δικτυωτού $k - i + 1$, όπου κάθε διαμέριση αποτελείται από i υποσύνολα.

Η εισαγωγή ενός μερικού LCA σε μια στοίβα απαιτεί στη χειρότερη περίπτωση το πλήρες άδειασμά της. Άρα για ένα δέντρο δεδομένων βάθους d, απαιτούνται κατά μέγιστο d ενέργειες pop. Κάθε ενέργεια pop εμπεριέχει στη χειρότερη περίπτωση $k(k-1)/2$ συνδυασμούς μερικών LCA για παραγωγή νέων και k ενημερώσεις μεγεθών του κόμβου πατέρα εξαιτίας της εξαγωγής του παιδιού από τη στοίβα. Συνεπώς, το συνολικό κόστος μιας ενέργειας pop σε χρόνο είναι $O(k^2)$. Η εισαγωγή ενός στοιχείου σε μια στοίβα προϋποθέτει επίσης το πολύ $d - 1$ ενέργειες push, όταν το βάθος του δέντρου είναι d, δηλ. η χρονική πολυπλοκότητά της είναι $O(d)$. Με βάση τον παραπάνω συλλογισμό, αν με $|S_i|$ συμβολίζουμε το μήκος της ανεστραμμένης λίστας (δηλ. τον αριθμό στιγμιοτύπων) της λέξες-κλειδιού w_i, η πολυπλοκότητα του LCAsz για την επεξεργασία όλων των στιγμιοτύπων των λέξεων κλειδιών μιας ερώτησης με k λέξεις-κλειδιά είναι:

$$O(dk^2 B_k \sum_{i=1}^{k} |S_i|)$$

Στην παγματικότητα, η πολυπλοκότητα του αλγορίθμου είναι αρκετά χαμηλότερη, καθώς οι παραπάνω θεωρήσεις δεν μπορούν να ισχύουν όλες ταυτόχρονα. Για παράδειγμα, αν για την εισαγωγή ενός στιγμιοτύπου χρειάζεται να αδειάσει πλήρως μία στοίβα, τότε το στιγμιότυπο αυτό δε μοιράζεται κάποιο κοινό μονοπάτι με άλλο στιγμιότυπο άλλης λέξης-κλειδιού. Κατά συνέπεια, αν κάτι τέτοιο ισχύει για όλα τα στιγμιότυπα τότε δε θα παραχθούν μερικοί LCA παρά μόνο στη ρίζα του δέντρου. Αυτή η απρατήρηση, λοιπόν, δείχνει ότι ο μέγιστος αριθμός βημάτων για το άδειασμα μιας στοίβας (δηλ. ο παραγων d) και ο αριθμός των πιθανώς παραγόμενων μερικών LCA ανά στοιχείο μιας στοίβας είναι αντιστρόφως συσχετισμένοι. Εξάλλου, η εισαγωγή ενός στιγμιοτύπου αυτούσια ή μέσω κάποιου μερικού LCA σε όλες τις στοίβες του δικτυωτού υπονοεί την εμφάνιση όλων των λέξεων-κλειδιών με όλους τους δυνατούς συνδυασμούς σε όλα τα υποδέντρα όλων των LCA μιας ερώτησης σε ένα δέντρο δεδομένων, κάτι το οποίο είναι πρακτικά αδύνατο.

Παρ΄ όλ αυτά, η πολυπλοκότητα που υπολογίστηκε για τη χειρότερη περίπτωση, είναι μια παραμετροποιημένη πολύπλοκότητα εξαρτώμενη του συνολικού μήκους των ανεστραμμένων λιστών $\sum |S_i|$, του αριθμού των λέξεων-κλειδιών k και του βάθους του δέντρου d. Μια σημαντική παρατήρηση είναι ότι ο αλγόριθμος LCAsz είναι γραμμικός σε σχέση με το μέγεθος των ανεστραμμένων λιστών για δεδομένα k και d. Αντίθετως, η πολυπλοκότητα αντίστοιχων αλγορίθμων γνωστών στη βιβλιογραφία είναι ανάλογη του γινομένου των μηκών των ανεστραμμένων λιστών, δηλ. τουλάχιστον $O(|S_i|^k)$, εμποδίσοντάς τους να κλιμακώνονται ομαλά όταν η είσοδός τους μεγαλώνει. Ο αλγόριθμος LCAsz, ωστόσο, εμφανίζει μια εκθετική εξάρτηση από τον αριθμό των λέξεων-κλειδιών εξαιτίας του αριθμού B^k, αλλά αυτό συμβαίνει χωρίς την εμπλοκή του μεγέθους της εισόδου.

3.3.3 Πειραματική μελέτη επίδοσης LSAsz

Για τη μελέτη της επίδοσης του αλγορίθμου LCAsz διενεργήθηκαν διάφορα πειράματα, τόσο ανεξάρτητα όσο και σε σύγκριση με τον αλγόριθμο SAOne [21]. Ο αλγόριθμος SAOne είναι ο μόνος γνωστός αλγόριθμος, ο οποίος υπολογίζει όλα τα αποτελέσματα μιας ερώτησης με λέξεις-κλειδιά σε μια δενδρική βάση δεδομένων, ενώ ταυτόχρονα έχει εγγενώς τη δυνατότητα να επιστρέψει και το μέγεθός τους. Όλοι οι υπόλοιποι προτεινόμενοι αλγόριθμοι στη βιβλιογραφία υλοποιούν κάποια σημασιολογία φιλτραρίσματος (π.χ., SLCA, ELCA, VLCA, MLCA κ.α.). Συνεπώς, δεν προσφέρονται για ευθεία σύγκριση με τον LCAsz ως προς την ταχύτητα αποτίμησης μιας ερώτησης, αφού υπολογίζουν ένα σημαντικά μικρότερο αριθμό αποτελέσμάτων.

Τα πειράματα διενεργήθηκαν σε ένα εικονικό μηχάνημα που λειτουργεί σε ένα σύστημα με λειτουργικό Windows 7. Η μνήμη RAM του εικονικού μηχανήματος ορίστηκε, κατά τη δημιουργία του, στα 8GB. Ο κώδικας του LCAsz αναπτύχθηκε στη γλώσσα Java, ενώ η παράμετρος JVM heap space του περιβάλλοντος της Java έμεινε αμετάβλητη, δηλ. στην προκαθορισμένη τιμή του 1,5GB.

Για τα πειράματα χρησιμοποιήθηκαν πραγματικές συλλογές δεδομένων από τις πηγές DBLP [25] (βιβλιογραφικά δεδομένα) και NASA [41] (αστρονομικά δεδομένα) καθώς επίσης και η τεχνητά παραγόμενη συλλογή δεδομένων XMark [50] (δεδομένα δημοπρασιών). Στον πίνακα 3.1 φαίνονται διάφορα στατιστικά χαραξτηριστικά αυτών των συλλογών δεδομένων.

	DBLP	*XMark*	*NASA*
size	850 MB	150 MB	23 MB
maximum depth	5	11	7
average depth	1,97	6,10	5,06
avg depth per keyword	2,00	4,53	4,99
# nodes	20.966.212	1.666.315	476.646
# keywords	2.599.843	57.775	65.862
# distinct labels	35	74	61
# distinct label paths	152	514	95

Πίνακας 3.1: Στατιστικά συλλογών δεδομένων DBLP, XMark και NASA

Η συλλογή DBLP είναι με το μεγαλύτερο μέγεθος και τη μεγαλύτερη ποικιλία λέξεων-κλειδιών, αλλά ταυτόχρονα η δενδρική του δομή είναι σχετικά ρηχή. Το 99% των λέξεων-κλειδιών βρίσκονται σε κόμβους του δέντρου σε βάθος 2, ενώ οι κόμβοι αυτοί αντιστοιχούν στο 87% του συνολικού αριθμού κόμβων του δέντρου. Οι συλλογές δεδομένων XMark και NASA είναι μικρότερες συλλογές αλλά βαθύτερες και παρουσιάζουν πολυπλοκότερη δομή, αφού κοινές ετικέτες των κόμβων τους εμφανίζονται σε διαφορετικά μονοπάτια ατικετών στο δέντρο. Οι λέξεις-κλειδιά και οι κόμβοι κατανέμονται σχεδόν ομοιόμορφα στα επίεδα 2-10 στο XMark και σταο επίπεδα 2-7 στο NASA. Ωστόσο, η πηγή δεδομένων NASA, σε σύγκριση με την XMark, διαθέτει μεγαλύτερη ποικιλία σε λέξεις-κλειδιά σε σχέση με το μέγεθός της.

Οι ανεστραμμένες λίστες των λέξεων-κλειδιών που προέκυψαν από λεκτική ανάλυση των XML συλλογών δεδομένων του Πίνακα 3.1 αποθηκεύτηκαν σε μία σχεσιακή βάση δεδομένων. Για την εκτέλεση πειραμάτων με ανεστραμμένες λίστες διαφορετικού μεγέθους επιλέχθηκαν μερικές από τις συχνότερα εμφανιζόμενες λέξεις-κλειδιά, και εν συνεχεία οι ανεστραμμένες λίστες τους περικόπηκαν σε διάφορα μήκη για τα διαφορετικά πειράματα. Η επιλογή λέξεων-κλειδιών ανάμεσα στις συχνότερες εξασφαλίζει

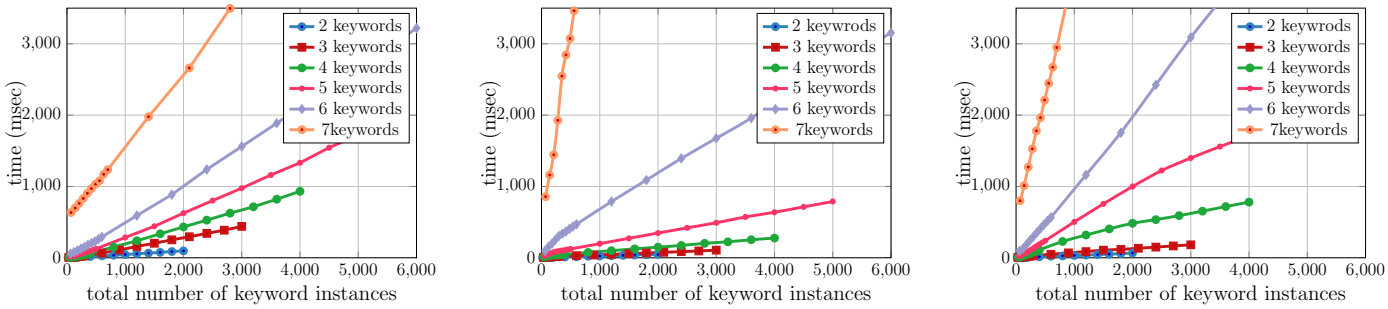
query	execution time	# keywords	total # kw instances	# results	LCA sizes
DBLP					
Q_1^D = {information systems security}	4,07 sec	3	256.086	1.027	0 - 4
Q_2^D = {online analytical processing}	0,86 sec	4	48.609	22	0 - 4
Q_3^D = {database query language}	0,96 sec	3	55.250	120	0 - 4
Q_4^D = {semantic web services automatic composition}	12,91 sec	5	101.420	10	0 - 4
Q_5^D = {spatial GIS applications}	1,11 sec	3	66.201	13	0 - 4
Q_6^D = {sensor networks power consumption}	3,63 sec	4	129.608	15	0 - 4
Q_7^D = {network computing algorithms}	2,25 sec	3	129.481	26	0 - 4
Q_8^D = {database query language}	0,96 sec	3	55.250	120	0 - 4
XMark					
Q_1^X = {province school female student}	0,80 sec	4	15.742	11	7 - 12
Q_2^X = {province school female male student}	2,20 sec	5	19.387	11	7 - 16
Q_3^X = {cash shipping Europe}	1,05 sec	3	32.929	43	3 - 9
Q_4^X = {charges shipping location United States}	35,55 sec	5	114.319	12.308	2 - 9
Q_5^X = {certainly apply leading expense offers approved}	3,19 sec	6	5.923	11	9 - 14
Q_6^X = {approved school expense offers student apply}	3,89 sec	6	8.975	11	9 - 14
Q_7^X = {cash payment delay order}	3,40 sec	4	47.928	250	2 - 10
NASA					
Q_1^N = {bright stars photometric measurements}	0,62 sec	4	10.792	27	0 - 18
Q_2^N = {estimated diameter planetary objects}	0,19 sec	4	2.814	9	0 - 10
Q_3^N = {stars position meridian circle}	0,54 sec	4	11.052	27	0 - 9
Q_4^N = {spectral photovoltaic photoconductive stars measurements}	1,11 sec	5	11.237	3	0 - 7
Q_5^N = {photometric equipment calibration spectral band}	0,63 sec	5	5.453	3	0 - 7
Q_6^N = {stellar spectral classification stars}	0,84 sec	4	12.329	91	0 - 17
Q_7^N = {stellar spectral classification stars emission}	1,99 sec	5	13.718	35	0 - 21

Πίνακας 3.2: Ερωτήσεις προς τις πηγές δεδομένων DBLP, XMark και NASA

την παραγωγή μεγάλου αριθμού αποτελεσμάτων (δηλ. LCA) κατά την εκτέλεση του αλγορίθμου. Όλα τα διαγράμματα παρουσιάζουν μέσες τιμές χρόνων εκτέλεσης για 10 διαφορετικές ερωτήσεις σε κάθε περίπτωση, που σχηματίστηκαν με τυχαία επιλογή λέξεων-κλειδιών από το σύνολο των συχνότερων. Ο χρόνος φόρτωσης των ανεστραμμένων λιστών στη μνήμη δεν παρουσιάζεται στα πειράματα, καθώς είναι ίδιος για όλες τις προσεγγίσεις, ενώ επίσης η μελέτη εστιάζει στην παρουσιάση και σύγκριση των χρόνων αποτίμησης των ερωτήσεων.

Επίδοση του LCAsz. Ο Πίνακας 3.2 δείχνει τα αποτελέσματα της αποτίμησης ενός δείγματος ερωτήσεων στις τρεις συλλογές δεδομένων. Οι ερωτήσεις αυτές περιέχουν λέξεις-κλειδιά με χιλιάδες στιγμιότυπα στις αντίστοιχες συλλογές δεδομένων και ο αλγόριθμος LCAsz, όπως φαίνεται, επιστρέφει τις απαντήσεις σε ρεαλιστικό χρόνο. Για παράδειγμα, για την ερώτηση Q_1^D ο αλγόριθμος επεξεργάζεται 256.086 στιγμιότυπα συνολικά σε περίπου 4sec επιστρέφοντας 1027 αποτελέσματα. Τα αποτελέσματα (δηλ. οι LCA) της ερώτησης κατανέμονται σε διάφορα μεγέθη, όπως φαίνεται στην τελευταία στήλη του πίνακα και επιστρέφονται ταξινομημένα. Τα μέγεθη των αποτελεσμάτων αντικατοπτρίζουν το βαθμό συσχέτισης των λέξεων-κλειδιών, με το 0 να υποδηλώνει εμφάνιση όλων των λέξεων-κλειδιών στον ίδιο κόμβο.

Κλιμάκωση αλγορίθμου LCAsz Η Εικόνα 3.6 παρουσιάζει πως ο αλγόριθμος LCAsz κλιμακώνεται χρονικά, όταν αυτξάνεται ο αριθμός των λέξεων-κλειδιών και ο αριυμός των στιγμιοτύπων τους σε καθεμιά από τις συλλογές δεδομένων. Κάθε καμπύλη στα διαγράμματα αντιστοιχεί σε μετρήσεις ερωτήσεων με τον ίδιο αριθμό λέξεων-κλειδιών, ο οποίος μεταβάλλεται από 2 έως 7. Ερωτήσεις με 8 λέξεις-κλειδιά ακολουθούν το ίδιο μοτίβο και απεικονίζονται στην Εικόνα 3.7. Δε συμπεριλαμβάνονται στα διαγράμματα της Εικόνας 3.6, καθώς οι χρόνοι εκτέλεσης τους ξεπερνούν το ανώταταο όριο των 3,5msec του κάθετου άξονα και η ενσωμάτωσή τους θα συμπίεζε την επεικόνιση των υπολοίπων. Ο αριθμός των στιγμιοτύπων ανά λέξη-κλειδί κυμαίνεται από 10 έως 1000. Αυτό σημαίνει πως για μια ερώτηση με 7 λέξεις-κλειδιά ο συνολικός



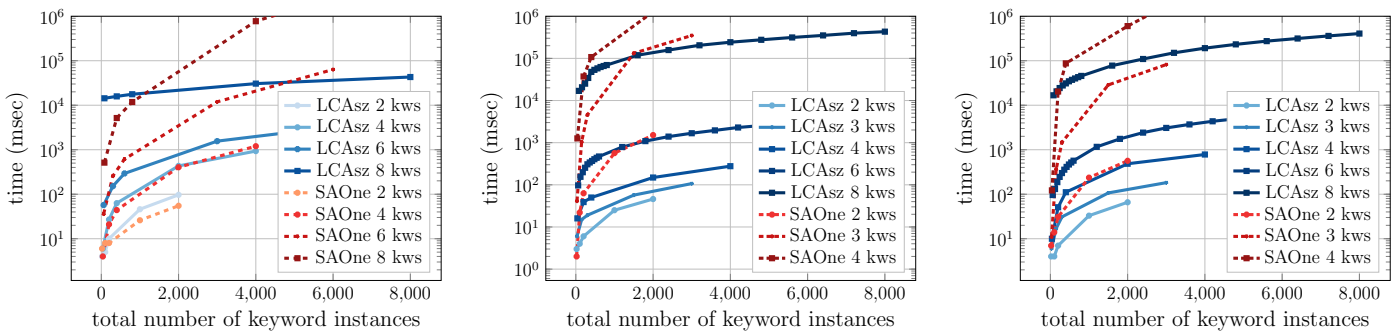
(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.6: Επίδοση του LCAsz για διαφορετικούς αριθμούς λέξεων-κλειδιών και στιγμιοτύπων

αριθμός στιγμιοτύπων μεταβάλλεται από 70 έως 7000.

Τα αποτελέσματα αυτών των διαγραμμάτων επιβεβαιώνουν καθαρά τη γραμμική συμπεριφορά του χρόνου εκτέλεσης του LCAsz σε σχέση με το μέγεθος της εισόδου για δεδομένο αριθμό λέξεων-κλειδιών, το οποίο και συσζητήθηκε στην Ενότητα 3.3.2. Η δαφορά της κλίσης μεταξύ των καμπυλών κάθε διαγράμματος δείχνει την κλιμάκωση του LCAsz σε σχέση με τον αριθμό λέξεων-κλειδιών μιας ερώτησης. Για τα δεδομένα του DBLP η μετάβαση είναι ομαλότερη σε σχέση με με τα XMark και NASA. Αυτό το φαινόμενο οφείλεται στο μικρό βάθος του DBLP που θέτει ένα όριο στο μέγιστο αριθμό από ενέργειες εισαγωγής κι εξαγωγής στις στοίβες του αλγορίθμου. Το βάθος του DBLP είναι 5, με το μέσο όρο να είναι μόνο 1,97 όπως καταγράφεται στον Πίνακα 3.1.

Σύγκριση με τον αλγόριθμο SAOne. Η Εικόνα 3.7 παρουσιάζει τη σύγκριση των χρόνων εκτέλεσης των αλγορίθμων LCAsz και SAOne που προτείνεται στην εργασία [21]. Ο κύριος αλγόριθμος SA της δουλειάς αυτής στοχεύει στον εντοπισμό των ελάχιστων συνδετικών δέντρων ενός συνόλου λέξεων κλειδιών, τα οποία επιστρέφονται με μια μια συνοπτκή μορφή ονομαζόμενη DMCT (δηλ. distance minimum connected tree), όπου εσωτερικοί κόμβοι παραλείπονται και οι εμπλεκόμενες ακμές αντικαθίστανται από μία με βάρος που αντιστοιχεί στην απόσταση των ακραίων κόμβων καθενός μονοπατιού που παραλήφθηκε. Οι συνόψεις αυτές είναι ομομορφικές με τα σύνολα λέξεων-κλειδών κάθε ερώτησης και ως τέτοιες ομαδοποιούνται ακόμα περισ-



(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.7: Σύγκριση αλγορίθμων LCAsz και SAOne για διαφορετικούς αριθμούς λέξεων-κλειδιών και στιγμιοτύπων

σότερο στη μορφή GDMCT (δηλ. grouped distance minimum connected tree). Ο SA επιστρέφει GDMCT που δεν ξεπερνούν ένα κατώφλι μεγέθους. Για την επίτευξη αυτού του στόχου, χρειάζεται να υπολογιστούν τα μεγέθη όλων των GDMCT.

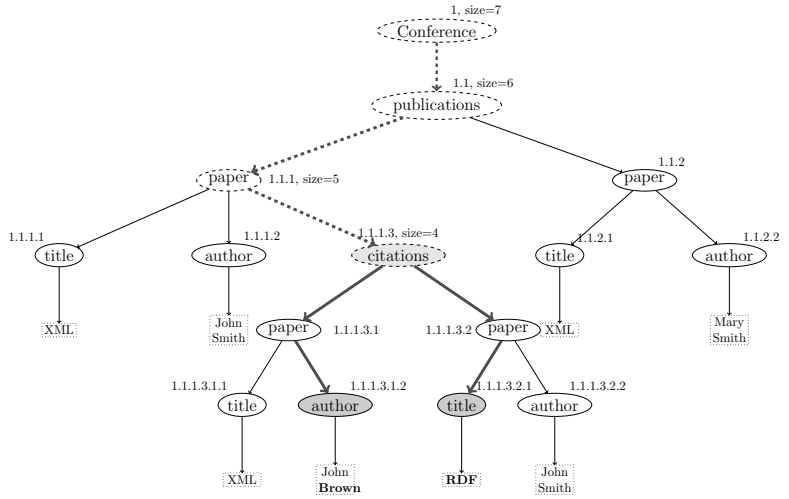
Ο αλγόριθμος SAOne είναι μια παραλλαγή του SA, που υπολογίζει μόνο LCA, οπότε αποφεύγει το βήμα της ομαδοποίησης των DMCT. Μ′ αυτήν την απλοποίηση αρκεί ο υπολογισμός ενός μόνο DMCT για την εύρεση ενός LCA ως αποτέλεσμα. Στη δική μας υλοποίηση του αλγορίθμου SAOne, σε αναλογία με τον LCAsz και τη σημασιολογία μεγέθους χαμηλότερου κοινού προγόνου, ορίζουμε το μικρότερο DMCT ενός LCA ως το κατάλληλο να αποδώσει το μέγεθος στον LCA αυτόν. Αν δεν οριστεί κατώφλι μεγέθους, ο SAOne μπορεί να επιστρέψει όλους του LCA μαζί με τα μεγέθη τους. Για λόγους δικαιοσύνσης στη σύγκριση, τροποποιήσαμε ελαφρώς τον LCAsz, ώστε να λαμβάνει υπόψιν λέξεις-κλειδιά που εμφανίζονται μαζί σε έναν κόμβο και ορίσαμε τα αναγνωριστικά των κόμβων να ακολουθούν την κωδικοποιήση Dewey. Σ′ αυτό το σημείο, αξίζει να σημειωθεί ότι ο SAOne επωφελείται από μια επιπλέον ανεστραμμένη λίστα όλων των χαμηλότερων κοινών προγόνων όλων των ζευγαριών στιγμιοτύπων των λέξεων κλειδιών. Αυτή η δομή, που χρειάζεται να προϋπολογιστεί, επιτρέπει την αποφυγή επεξεργασίας των εσωτερικών κόμβων του δέντρου που δεν είναι στιγμιότυπα τουλάχιστον μίας λέξεης-κλειδιού. Έτσι, γλιτώνει τις αντίστοιχες ενέργειες εισαγωγής κι εξαγωγής από τη στοίβα, που χρησιμοποιούν οι αλγόριθμοι SA και SAOne.

Η Εικόνα 3.7 δείχνει ότι ο LCAsz έχει καλύτερη επίδοση στις περισσότερες περιπτώσεις από τον αλγόριθμο SAOne και στις τρεις συλλογές δεδομένων. Σημειωτέον ότι ο άξονας y είναι σε λογαριθμική κλίμακα. Οι πειραματικές μετρήσεις με τον SAOne απεικονίζονται με διακεκομμένες καμπύλες. Ο SAOne ξεπερνά τον LCAsz μόνο στο σύνολο δεδομένων DBLP όταν ο αριθμός των λέξων-κλειδιών και των στιγμιοτύπων τους είναι σχετικά μικρός. Με την αύξηση, όμως, των στιγμιοτύπων οι χρόνοι εκτέλεσης του SAOne αυξάνονται σημαντικά, σε συμφωνία και με την ανάλυση της πολυπλοκότητας του SAOne που δείχνει ότι ο χρόνος εκτέλεσης του αλγορίθμου εξαρτάται από το μέγεθος της εισόδου (δηλ. τον αριθμό των στιγμιοτύπων των λέξεων-κλειδιών) υψωμένο στη δύναμη του αριθμού των λέξεων-κλειδιών στο διπλάσιο. Για παράδειγμα, παρατηρώντας τις ερωτήσεις του Πίνακα 3.2, ο SAOne χρεάζεταο 43sec αντί του 0,96sec που απαιτούνται για τον LCAsz για την ερώτηση Q_3^D σε 55.250 στιγμιότυπα λέξεων-κλειδιών του DBLP.

Με την αύξηση του βάθους της συλλογής δεδομένων, ο αριθμός των δυνατών GDMCT που παράγει ο SAOne αυξάνεται πολύ γρήγορα. Έτσι, για τις συλλογές δεδεομένων XMark και NASA, ο LCAsz ξεκάθαρα ξεπερνά τον SAOne για περισσότερα από λίγα στιγμιότυπα των λέξεων-κλειδιών και για όλα τα μήκη μιας ερώτησης. Ο SAOne αποτυγχάνει να επιστρέψει αποτελέσματα σε ρεαλιστικό χρόνο για περισσότερες από 4 λέξεις-κλειδιά στα XMark και NASA. Για το λόγο αυτό, στις Εικόνες 3.7β′ και 3.7γ′, για 6 και 8 λέξεις-κλειδιά απεικονίζεται μόνο η επίδοαη του LCAsz. Αξιοσημείωτο είναι επίδης ότι ο SAOne για 2 λέξεις-κλειδιά επιδεικνύει χειρότερη επίδοση από τον LCAsz για 4 λέξεις-κλειδιά για περισσότερα από 100 στιγμιότυπα και χειρότερα ακόμη από τον LCAsz για 6 λέξεις-κλειδιά για πάνω από 1750 στιγμιότυπα. Οι αριθμοί αυτοί στιγμιοτύπων είναι απόλυτα ρεαλιστικοί, ανα αναλογιστεί κανείς μεγάλες συλλογές δεδομένων και λέξεις-κλειδιά με μεγάλη συχνότητα εμφάνισης.

3.4 Ερωτήσεις με σπάνιες λέξεις-κλειδιά

Σ΄ αυτήν την ενότητα, παρουσιάζεται μια παραλλαγή του αλγορίθμου LCAsz, του *LCAszI*, που εκμεταλλεύεται την εξάρτηση του LCAsz από τον αριθμό των λέξεων-κλειδιών προς όφελος της επίδοσής του. Η βασική ιδέα είναι ο περιορισμός των λέξεων-κλειδιών της ερώτησης, που χρειάζεται να επεξεργαστεί ο αλγόριθμος, σ΄ αυτές που είναι συχνά εμφανιζόμενες. Έτσι, ο LCAsz εκτελείται πρακτικά μόνο για τις συχνές λέξεις-κλειδιά. Οι σπάνιες λέξεις-κλειδιά λαμβάνονται επίσης υπόψιν στον υπολογισμό αλλά ως μια διαφορετική παράμετρος του αλγορίθμου. Παρακάτω, εξηγείται η λογική πίσω από αυτήν την παραλλαγή του αλγορίθμου. Για λέξεις-κλειδιά με λίγα στιγμιότυπα (δηλ. σπάνιες), η διαδικασία αυτή εφαρμόζεται κατ΄ επανάληψη.



Σχήμα 3.8: Το υποδέντρο των σπάνιων λέξεων κλειδιών Brown και RDF

3.4.1 Ο αλγόριθμος LCAszI

Ας θεωρήσουμε το ακραίο σενάριο κατά το οποίο οι σπάνιες λέξεις-κλειδιά της υπό εξέτασης ερώτησης εμφανίζονται μόνο μία φορά στο δέντρο δεδομένων. Στο δέντρο του παραδείγματός μας, τέτοιες λέξεις-κλειδιά είναι οι `Brown` και `RDF` με LCA τον κόμβο 1.1.1.3. Μιας ερώτηση που περιέχει αυτές τις λέξεις-κλειδιά, είτε περιέχει και άλλες λέξεις είτε όχι, αναγκαστικά πρέπει να περιλαμβάνει σε κάθε αποτέλεσμα τον κόμβο 1.1.1.3 ή κάποιον πρόγονό του, εξαιτίας της ιδιότητας 3.1 (κόμβοι περιγεγραμμένοι με διακεκομμένες γραμμές στην Εικόνα 3.8). Αυτή η παρατήρηση τοποθετεί κάθε αποτέλεσμα μιας ερώτησης που περιέχει τα `Brown` και `RDF` στο μονοπάτι που ορίζεται από τη ρίζα (δηλ. τον κόμβο 1) ως τον κόμβο 1.1.1.3. Εξάλλου, το υποδέντρο του μερικού LCA 1.1.1.3 συνεισφέρει με το ίδιο πάντα μέγεθος σε οποιοδήποτε τελικό αποτέλεσμα μιας ερώτησης με τις συγκεκριμένες λέξεις-κλειδιά. Έτσι, ο αλγόριθμος LCAszI προχωρά στον υπολογισμό όπως ο LCAsz αγνοώντας όμως οποιαδήποτε συνεισφορά στο μέγεθος κάπου LCA από τις ακμές του υποδέντρου των σπάνιων λέξεων-κλειδιών. Οι ακμές αυτές παρουσιάζονται με έντονες γραμμές (διακεκομμένες ή όχι) στην Εικόνα 3.8. Για να αντιστιαθμιστεί η διαφορά στα μεγέθη των αποτελεσμάτων, πριν επιστραφούν οι πλήρεις LCA από τον LCAszI το μέγεθός τους επαυξάνεται κατά το μέγεθος του ελάχιστου συνδετικού υποδέντρου των σπάνιων λέξεων-κλειδιών. Με αυτές τις παρατηρήσεις υπόψιν, ο LCAszI σχεδιάζεται με βάση τον LCAsz εφαρμόζοντας

τις παρακάτω αλλαγές:

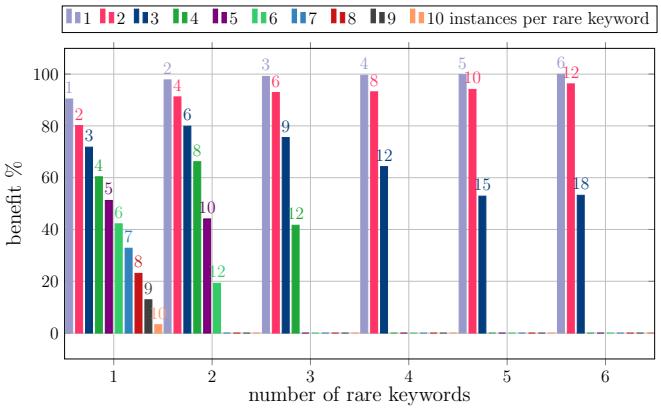
- Ο LCAszI εκτελείται LCAsz μόνο για τις συχνές λέξεις-κλειδιά μιας ερώτησης.
- Στη διάρκεια του υπολογισμού, όσο η ρίζα του υποδέντρου του LCA των συχνών λέξεων-κλειδιών δε βρίσκεται πάνω στο μονοπάτι μεταξύ της ρίζας και του LCA των σπάνιων-λέξεων κλειδιών, δεν επιστρέφεται ως αποτέλεσμα. Αντ΄ αυτού το μέγεθός τους συνεχίζει να προωθείται στους προγόνους του.
- Πριν ένας LCA l των συχνών λέξεων-κλειδιών επιστραφεί σαν αποτέλεσμα, το μέγεθός του προσαυξάνεται κατά το μέγεθος του υποδέντρου του μερικού LCA των σπάνιων λέξεων-κλειδιών κάτω από τον l. Στην Εικόνα 3.8, τα μεγέθη αυτών των υποδέντρων για τις λέξεις-κλειδιά `Brown` και `RDF` κάτω από διαφορετικούς κόμβους σημειώνονται δίπλα στους κωδικούς Dewey των κόμβων αυτών.
- Στον υπολογισμό των μεγεθών των μερικών LCA των συχνών λέξεων-κλειδιών, μόνο οι ακμές εκτός του υποδέντρου των σπάνιων λέξεων-κλειδιών λαμβάνονται υπόψιν (δηλ. οι ακμές που σημειώνονται με λεπτές γραμμές στην Εικόνα 3.8).

Η παράλειψη των ακμών του υποδέντρου των σπάνιων λέξεων-κλειδιών δεν υπονομεύει ούτε την ορθότητα του υπολογισμού των μεγεθών των μερικών LCA, ούτε τη σύγκρισή τους για την εύρεση του μικρότερου κατά την εκτέλεση του αλγορίθμου. Η περικοπή που διενεργείται από τον LCAsz των συγκρίσιμων μερικών LCA με βάση τα μεγέθη τους γίνεται σε σχέση πάντα με τον τρέχοντα κορυφαίο κόμβο της στοίβας όπου βρίσκεται η εκτέλεση του αλγορίθμου. Κάτω απ΄ ατυόν τον κόμβο το μέγεθος μερικού LCA των σπάνιων λέξεων-κλειδιών είναι σταθερό. Για παράδειγμα, όταν ο κόμβος 1.1.1 εξετάζεται από τον LCAszI, το μέγεθος του υποδέντρου του μερικού LCA των `Brown` και `RDF` είναι 5. Για την ερώτηση {`John, Brown, RDF`}, η σύγκριση των μεγεθών των LCA που αντιστοιχούν στα υποδέντρα που περιέχουν για τη λέξη-κλειδί `John` είτε τον κόμβο 1.1.1.2 είτε τον 1.1.1.3.1.2 ευνοεί το δεύτερο, είτε ως μια σύγκριση μεταξύ του 1 και του 0 (δηλ. αγνοώντας τις ακμές του υποδέντρου των `Brown, RDF`), είτε μεταξύ του 6 και του 5, αντίστοιχα.

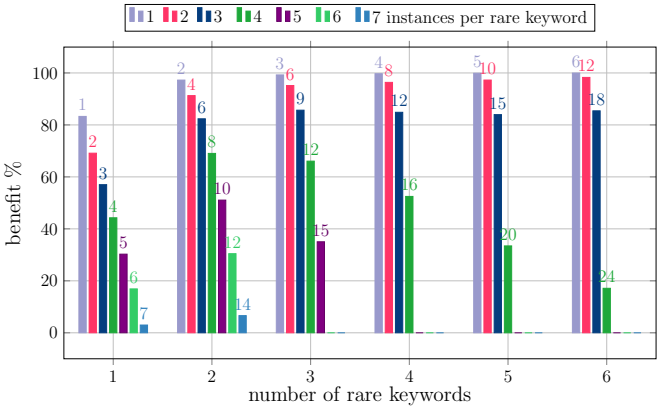
3.4.2 Βελτίωση επίδοσης LCAszI σε σχέση με τον LCAsz

Η εικόνα 3.9 παρουσιάζει τα πειραματικά αποτελέσματα επίδοσης από την εκτέλεση του LCAszI. Συγκεκριμένα, τα διαγράμματα δείχνουν το κέρδος στην επίδοση του LCAszI έναντι του LCAsz για την περίπτωση ερωτήσεων με 8 λέξεις-κλειδιά με τις σπάνιες να κυμαίνονται μεταξύ 1 και 6. Οι συνθήκες των πειραμάτων είναι οι ίδιες που περιγράφονται στην Ενότητα 3.3.3. Τα πειράματα επαναλήφθηκαν για διαφορετικούς συνδυασμούς από 8 λέξεις-κλειδιά μεγάλης συχνότητας εμφάνισης. Για κάθε λέξη-κλειδί επιλέχθηκαν 100 στιγμιότυπα με τυχαίο τρόπο. Σε κάθε περίπτωση, οι σπάνιες λέξεις-κλειδιά επιλέχθηκαν τυχαία και στη συνέχεια επίσης τυχαία περικόπηκαν οι ανεστραμμένες τους λίστες ώστε να περιοριστούν σε μήκη που κυμαίνονταν από 1 έως 10 στιγμιότυπα. Με στόχο να κατανεμηθούν οι σπάνιες λέξεις-κλειδιά ομοιόμορφα στο δέντρο, ώστε να δίνουν διαφορετικά αποτελέσματα με τις λέξεις-κλειδιά μεγάλης συχνότητας, επιλέχθηκαν τα 1 έως 10 στιγμιότυπα ομοιόμορφα κατανεμημένα από τις ανεστραμμένες λίστες των θεωρούμενων ως σπάνια εμφανιζόμενων λέξεων-κλειδιών. Για κάθε επιλογή από σπάνιες λέξεις-κλειδιά και για καθένα διαφορετικό αριθμό στιγμιοτύπων, οι αλγόριθμοι LCAsz και LCAszI εκτελέστηκαν 10 φορές.

Το όφελος από την επιλογή του LCAszI έναντι του LCAsz στις περισσότερες περιπτώσεις είναι προφανές. Οι αριθμοί στις μπάρες υποδηλώνουν τον αριθμό των



(α′) DBLP



(β′) XMark

# σπάνιων λέξεων-κλειδιών	1	2	3	4	5	6
DBLP	10	37	66	84	247	734
XMark	7	50	127	259	1028	4101

Πίνακας (3.3) Μέγιστος αριθμός στιγμιοτύπων για τον οποίο ο LCAszI είναι ταχύτερος

(γ′) Κατώφλι αριθμού σπάνιων λέξεων-κλειδιών και στιγμιοτύπων που δικαιολογούν την προτίμηση του LCAszI έναντι του LCAsz

Σχήμα 3.9: Το κέρδος στην επίδοση του LCAszI σε σχέση με τον LCAsz για 8 λέξεις-κλειδιά των 100 στιγμιοτύπων, όταν αλλάζει ο αριθμός των σπάνιων λέξεων-κλειδιών και των στιγμιοτύπων τους.

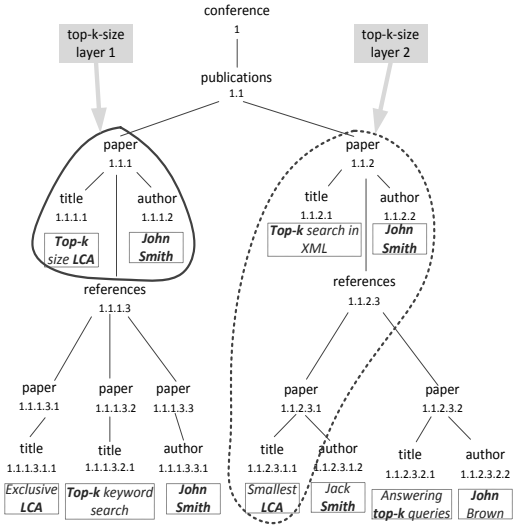
συνολικών στιγμιοτύπων όλων των σπάνιων λέξεων-κλειδιών. Αυτος ο αριθμός υποδεικνύει τον ελάχιστο αριθμό στιγμιοτύπων για τον οποίο καταγράφεται κέρδος από την επιλογή της παραλλαγής του αλγορίθμου. Για παράδειγμα, η επιλογή 3 σπάνιων λέξεων-κλειδιών από 4 στιγμιότυπα η καθεμία καταλήγει σε 12 στιγμιότυπα συνολικά αλλά σε $4^3 = 64$ διαφορετικά υποδέντρα σπάνιων λέξεων-κλειδιών, όπως συζητήθηκε στην ενότητα 3.4.1. Σε άλλη περίπτωση, π.χ. για 3 σπάνιες λέξεις-κλειδιά με 1, 1 και 64 στιγμιότυπα αντίστοιχα, τα υποδέντρα των σπάνιων λέξεων-κλειδιών είναι και πάλι 64 $(1 * 1 * 64)$, αλλά ο συνολικός αριθμός στιγμιοτύπων είναι 66. Μ′ αυτό το σκεπτικό, ο Πίνακας 3.9γ′ δείχνει το μέγιστο αριθμό στιγμιοτύπων των σπάνιων λέξεων-κλειδιών, που ευνοούν τη χρήση του LCAszI έναντι του LCAsz για δεδομένο αριθμό από σπάνιες λέξεις-κλειδιά.

Το κέρδος από την εκτέλεση του LCAszI αυξάνεται όταν αυξάνεται και ο αριθμός των σπάνιων λέξεων-κλειδιών σε σχέση με το συνολικό αριθμό λέξεων-κλειδιών μιας ερώτησης. Αυτή η συμπεριφορά αξηγείται από τη μείωση των επιπέδων του δικτυωτού του αλγορίθμου και του συνακόλουθου αριθμού στοιβών, αφού μόνο οι στοίβες των συχνών λέξεων-κλειδιών χρειάζεται να δημιουργηθούν. Για παράδειγμα, για μια ερώτησης με 8 λέξεις-κλειδιά όπου οι 6 είναι σπάνιες, κατά τον υπολογισμό σχηματίζεται ένα δικτυωτό από στοίβες για 2 λέξεις-κλειδιά μόνο. Η συλλογή XMark επωφελείται περισσότερο από τον LCAszI απ′ ό,τι η συλλογή DBLP, όπως καταγράφεται στον Πίνακα 3.9γ′. Από μια διαφορετική οπτική, αυτό είναι αναμενόμενο από την ομαλότερη αύξηση της κλίσης των καμπυλών στα διαγράμματα κλιμάκωσης του LCAsz στην Εικόνα 3.6 για το DBLP σε σχέση με το XMark. Τα αποτελέσματα για το NASA είναι ανάλογα με αυτά του XMark.

3.5 Κλιμακωτή ταξινόμηση αποτελεσμάτων

Η κλιμακωτή ταξινόμηση των αποτελεσμάτων με βάση την έννοια του χαμηλότερου κοινού προγόνου, δημιουργεί τις προδιαγραφές για αναζήτηση των κορυφαίων k αποτελεσμάτων. Σχεδιάστηκαν τρεις διαφορετικές εκδόσεις αλγορίθμων για το σκοπό αυτό. Στην απλούστερη έκδοση, ορίζεται από το χρήστη ένα κατώφλι μεγέθους το οποίο δεν πρέπει να ξεπερνούν τα αποτελέσματα αναζήτησης. Στη δεύτερη έκδοση, ορίζεται από το χρήστη ο αριθμός των κορυφαίων αποτελεσμάτων που είναι επιθυμητό να επιστραφούν. Η πιο ενδιαφέρουσα είναι η τρίτη έκδοση του αλγορίθμου, στην οποία ο χρήστης ορίζει τον αριθμό των κορυφαίων μεγεθών LCA που επιθυμεί να επιστραφούν. Σε όλους τους αλγορίθμους, χρησιμοποιούνται τα όρια που ορίζονται στην είσοδο από το χρήστη, για να απορρίψουν όσο το δυνατόν νωρίτερα στον υπολογισμό, αποτελέσματα που ξεπερνούν το επιτρεπτό ανά περίπτωση μέγεθος. Έτσι οι top-k αλγόριθμοι επιδεικνύουν πολύ καλύτερη επίδοση σε σχέση με το βασικό LCAsz.

Η κλιμακωτή ταξινόμηση με βάση το μέγεθος LCA δεν είναι λεπτομερής. Ομαδοποιεί, όμως, αποτελέσματα με κοινά χαρακτηριστικά. Για παράδειγμα, για την ερώτηση {`XML, query, John, Smith`}, ένα αποτέλεσμα με LCA έναν κόμβο paper όπου τα {`John, Smith`} εντοπίζονται σ΄ ένα παιδί του με ετικέτα author και τα {`XML, query`} σ΄ ένα άλλο παιδί με ετικέτα title, τότε το μέγεθός του είναι 2. Το ίδιο μέγεθος θα έχει οποιοδήποτε αποτέλεσμα αντιστοιχεί τις λέξεις - κλειδιά με τον ίδιο τρόπο σε κόμβους του δένδρου δεδομένων. Κατά συνέπεια, ανεξάρτητα από τον αριθμό των επιμέρους αποτελεσμάτων, ο χρήστης ορίζοντας ότι επιθυμεί τα top-1-size αποτελέσματα, θα λάβει όλα εκείνα με το ελάχιστο δυνατό μέγεθος. Έτσι η κλιμακωτή ταξινόμηση δίνει την ευκαιρία στο χρήστη να ορίζει το όριο k χωρίς να χρειάζεται να έχει εκτίμηση του ακριβούς αριθμού κορυφαίων αποτελεσμάτων που χρειάζεται να εξετάσει, δεδομένου πως δεν έχει στοιχεία για το μέγεθος του συνόλου δεδομένων ή για τη συχνότητα εμφάνισης των λέξεων - κλειδιών της αναζήτησής του σ΄ αυτό.

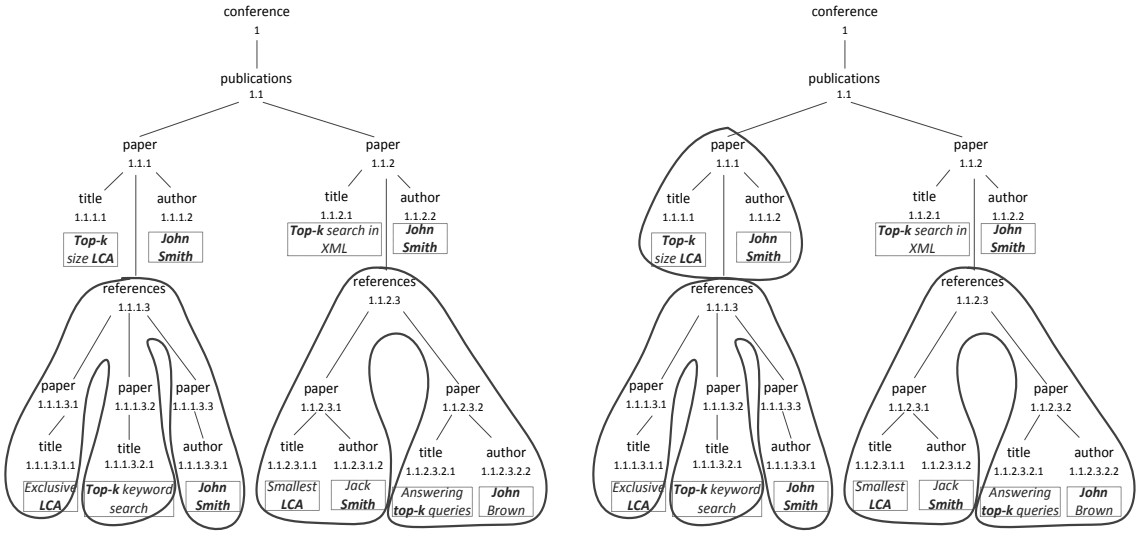


Σχήμα 3.10: top-1-size TLCA και top-2-size TLCA σημασιολογία για την ερώτηση {`top-k, LCA, John, Smith`}

Σε αντίθεση με τις σημασιολογίες SLCA και ELCA που ευνοούν τους LCA, οι οποίοι βρίσκονται χαμηλότερα στο δέντρο, εισάγουμε τη σημασιολογία tight LCA - TLCA, που λαμβάνει υπόψιν την εγγύτητα των κλεξεων - κλειδιών (keyword proximity) σε α-

ναλογία με το πεδίο του κλασικού IR. Συνεπώς, προτείνουμε ένα μοντέλο όπου το μέγεθος του ελάχιστου συνδετικού δέντρου των στιγμιοτύπων των λέξεων-κλειδιών είναι το κριτήριο ταξινόμησης των απαντήσεων σε μια ερώτηση λέξεων-κλειδιών. Η απάντηση σε μια ερώτηση λέξεων-κλειδιών με υιοθέτηση της TLCA σημασιολογίας είναι μία κλιμακωτά ταξινομημένη λίστα του πλήρους συνόλου των πιθανών απαντήσεων. Έτσι, σε αντίθεση με τις προτεινόμενες σημασιολογίες η πληρότητα της απάντησης είναι 100%.

Η σημασιολογία συμπαγούς LCA είναι εμπνευσμένη από την έννοια της εγγύτητας των λέξεων κλειδιών, η οποία έχει αξιοποιηθεί στο πεδίο της ανάκτησης πληροφορίας (IR) [4] και βασίζεται στη διαισθητική παρατήρηση ότι η ποιότητα ενός εποτελέσματος εξαρτάται από την εγγύτητα των στογμιοτύπων των λέξεων-κλειδιών. Η ταξινόμηση σύμφωνα με τη συμασιολογία TLCA, εκτός από την ταξινόμηση ομαδοποιεί επίσης αποτελέσματα με κοινό μέγεθος σε μία κοινή βαθμίδα. Κατά συνέπεια, εφόσον το μέγεθος χρησιμοποιείται ως κριτήριο σχετικότητας με την ερώτηση, αποτελέσματα ίσου βαθμού σχετικότητας ομαδοποιούνται μαζί. Έτς, αποτελέσματα ίδιας σχετικότητας με την ερώτηση ομαδοποιούνται στο ίδιο επίπεδο ταξινόμησης. Τα αποτελέσματα κορυφαίου επιπέδου (top-1-size) είναι τα πιο σχετικά. Στις περισσότερες περιπτώσεις, δε, τα αποτελέσματα των δύο κορυφαίων επιπέδων (top-2-size) μαζί φαίνεται πως είναι αρκετά για την ανάκτηση του συνόλου των σχετικών αποτελεσμάτων.



(α′) SLCAs (β′) ELCAs

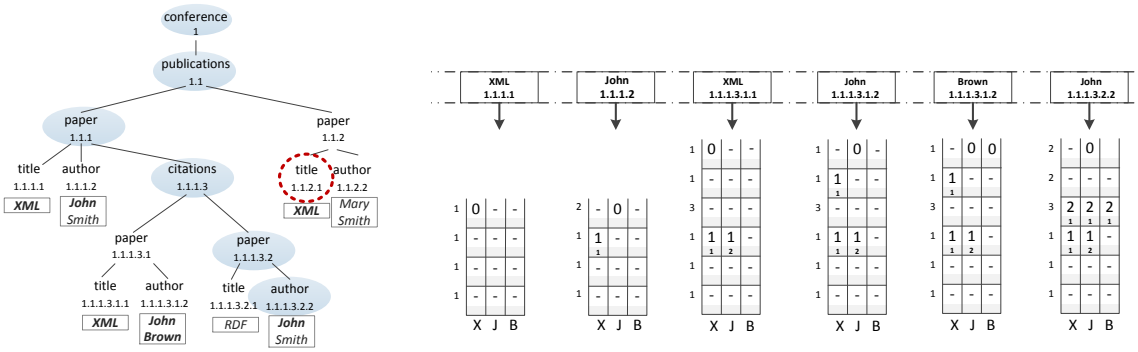
Σχήμα 3.11: SLCA και ELCA σημασιολογία για την ερώτηση {top-k, LCA, John, Smith}

Στις Εικόνες 3.10-3.11 παρουσιάζεται με ένα παράδειγμα σε ένα δέντρο δεδομένων πώς σχετίζεται η σημασιολογία TLCA με τις σημασιολογίες SLCA και ELCA. Αντίθετα με τις σημασιολογίες VLCA και MLCA, οι SLCA και ELCA δε λαμβάνουν υπόψιν τις ετικέτες των κόμβων του δέντρου αλλά μόνο τις δομικές συσχετίσεις μεταξύ των στιγμιοτύπων των λέξεων-κλειδιών. Ας πάρουμε τη βιβλιογραφική βάση δεδομένων που εμφανίζεται στις Εικόνες 3.10-3.11 και την ερώτηση top-k, LCA, John, Smith}. Είναι λογικό να θεωρήσει κανείς πως ο χρήστης ενδιαφέρεται για εργασίες με συγγραφέα κάποιον John Smith με θέμα που σχετίζεται με top-k και LCA. Το πιο σχετικό αποτέλεσμα με την ερώτηση αυτή είναι ο κόμβος paper (1.1.1), ο οποίος είναι πράγμα-

τι ένα paper με θέματα εντοπ-κ και LCA και συγγραφέα John Smith. Ένα λιγότερο σχετικό αποτέλεσμα είναι ο κόμβος paper (1.1.2) με θέμα 'top-k σεαρχ¨, συγγραφέα John Smith και αναφορά σε μια δημοσίευση για smallest LCA. Η σημασιολογία ELCA απορρίπτει το αποτέλεσμα paper (1.1.2), εξαιτίας της παρουσίας του LCA references (1.1.2.3), ο οποίος είναι λιγότερο σχετικός αφού αντιπροσωπεύει μια συλλογή από διαφορετικούς κόμβους. Εκτός αυτού, επιστρέφει και το 2ο κόμβο references, που είναι επίσης μη σχετικός. Η σημασιολογία SLCA επιστρέφει τα δύο άσχετα αποτελέσματα και χάνει και τα δύο σωστά. Αντίθετα, η TLCA σημασιολογία δε χάνει κανένα σωστό αποτέλεσμα. Το πιο σχετικό αποτέλεσμα paper (1.1.1) περλαμβάνεται στα πιο σχετικά αποτελέσματα του κορυφαίου επιπέδου (layer-1 Εικόνα 3.11) και ο σχετικός σε μικρότερο βαθμό κόμβος paper (1.1.2) στο επόμενο επίπεδο (layer-2 Εικόνα 3.11). Το παράδειγμα αυτό δείχνει πώς η σημασιολογία TLCA έχει τη δυνατότητα να επιστρέφει σωστά αποτελέσματα που βρίσκονται ψηλότερα στο δέντρο αναζήτησης. Αυτή η ιδιότητα είναι ιδιαίτερα χρήσιμη δε βαθειά και αναδρομικά δεδνδρικά σύνολα δεδομένων στα οποία οι σημασιολογίες ELCA και SLCA είναι πιθανό να αποτυγχάνουν.

3.5.1 Αλγόριθμοι top-k για υπολογισμό κορυφαίων χαμηλότερων κοινών προγόνων με σημασιολογία μεγέθους

Σ΄ αυτήν την ενότητα, παρουσιάζονται αλγόριθμοι που υλοποιούν τη σημασιολογία TLCA αλλά επιστρέφουν μόνο τα κορυφαία αποτελέσματα. Οι αλγόριθμοι αυτοί βασίζονται στην ιδέα του αλγορίθμου LCAsz, που περιγράφηκε στην Ενότητα 3.3.1. Περιλαμβάνουν, όμως, τις απαραίτητες προσαρμογές ώστε να επιστρέψουν τα επιθυμητά αποτελέσματα αποφεύγοντας τον υπολογισμό του πλήρους συνόλου των LCA μιας ερώτησης. Αρχικά περιγράφεται ο αλγόριθμος $T\text{-}LCAsz$, ο οποίος για δεδομένη ερώτησης Q και ένα κατώφλι μεγέθους T, επιστρέφει τους LCA ενός δέντρου με μέγεθος το πολύ T. Ο αλγόριθμος με κατώφλι αποτελεί τη βάση για τους αλγορίθμους κορυφαίων αποτελεσμάτων που θα παρουσιάστούν στη συνέχεια.



(α΄) Το δέντρο δεδομένων της Εικόνας 3.1

(β΄) Καταστάσεις της αρχικής στοίβας

Σχήμα 3.12: Καταστάσεις της αρχικής στοίβας του δικτυωτού για την ερώτηση {XML, John, Brown} όταν γίνεται η επεξεργασία του στιγμιοτύπου 1.1.2.1 για τη λέξη-κλειδί XML

3.5.1.1 Αλγόριθμος με κατώφλι μεγέθους (T-LCAsz)

Ο αλγόριθμος με κατώφλι T-LCAsz είναι ένας αλγόριθμος που χρησιμοποιεί στοίβες για την αποτίμηση των ερωτήσεων λέξεων-κλειδιών σε δενδρικά δεδομένα, επιστρέφοντας όλους τους LCA μέχρι ένα κατώφλι μεγέθους T, ταξινομημένους κατά σειρά μεγέθους. Η είσοδος του αλγορίθμου είναι οι ανεστραμμένες λίστες των λέξεων-κλειδιών σε ένα δέντρο δεδομένων, μία ερώτηση Q και ένα κατώφλι μεγέθους T. Η έξοδος του T-LCAsz είναι η απάντηση $A = [(l_1, s_1), (l_2, s_2), \ldots]$ του Q στο δέντρο δεδομένων, όπου $s_i \leq T$.

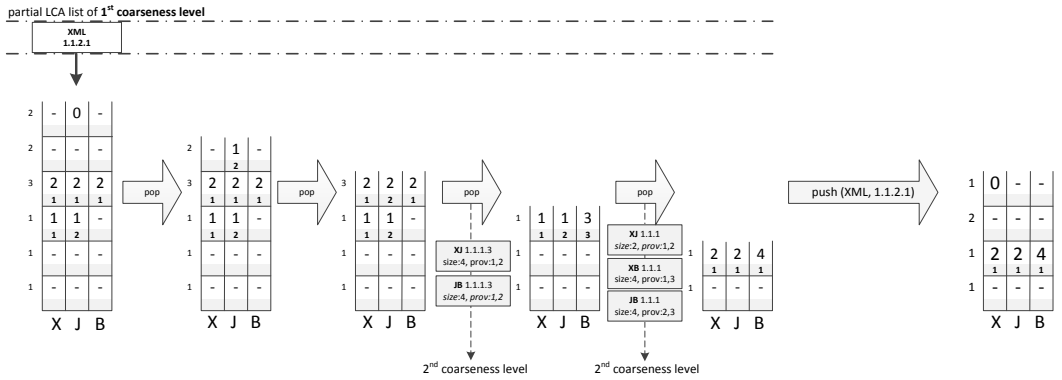
Algorithm 2: T-LCAsz

```
1  LCAsz(k_1,...,k_n: keyword query, invL: inverted lists, T: size threshold)
2      kwSubsets = {{k_1},{k_2}...{k_n}}
3      buildLattice() /* constructs empty stacks of the lattice and updates
           kwSubsets */
4      while currentNode = getNextNodeFromInvertedLists() do
5          coarsenessLevel = 1, size=0, provenance=∅
6          pLCA = newPartialLCA(currentNode.ID, currentNode.kwSubset, size,
           provenance)
7          addPartialLCA(1, pLCA)
8          while partialLCAlists contains partialLCAs for coarsenessLevel do
9              while partialLCA = partialLCAlists(coarsenessLevel).next() do
10                 for every stack of coarsenessLevel containing partialLCA.kwSubset do
11                     if partialLCA.size ≤ T   /* only for top-k mode */
12                     then
13                         push(stack, partialLCA.ID, partialLCA.kwSubset,
                           partialLCA.size)
14                 if coarsenessLevel < n then
15                     if partialLCA.size ≤ T   /* only for top-k mode */
16                     then
17                         addPartialLCA(coarsenessLevel+1, partialLCA)
18             coarsenessLevel++
19     emptyStacks()
```

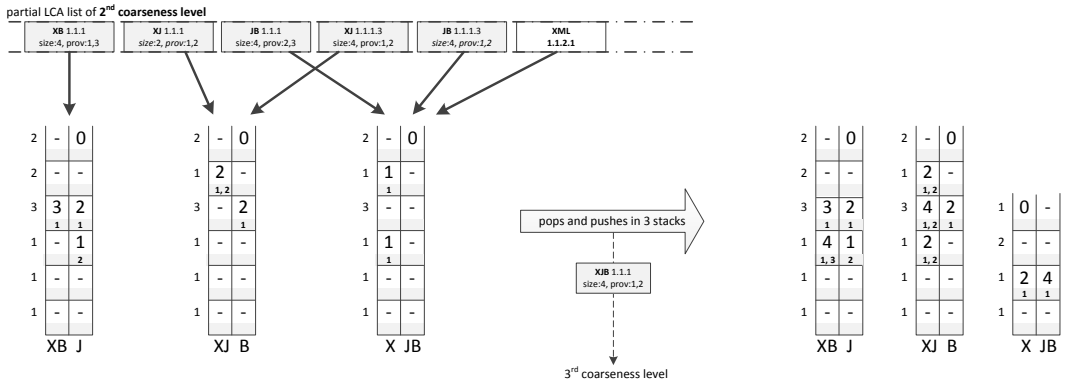
Ο αλγόριθμος T-LCAsz λειτουργεί όπως και ο αλγόριθμος LCAsz σχηματίζοντας μερικούς LCA και στη συνέχεια πλήρεις LCA ακλουθώντας τα μονοπάτια του δικτυωτού που σχηματίζουν οι διαμερίσεις του συνόλου των λέξεων-κλειδιών. Σε κάθε βήμα του αλγορίθμου κάποιος μερικός LCA εισάγεται σε κάποια στοίβα του δικτυωτού. Αυτό έχει ως αποτέλεσμα την εξαγωγή από τη στοίβα των κόμβων που δεν είναι πρόγονοι του συγκεκριμένου LCA. Κάθε τέτοια ενέργεια, δηλ. ενέργεια pop, προκαλεί το συνδυασμό μερικών LCA των εξαγόμενων στοιχείων της στοίβας προς δημιουργία νέων που θα προωθηθούν στα ακόλουθα επίπεδα του δικτυωτού. Τα μέγεθη των μερικών LCA που φιλοξενούνται σε εξαγόμενα στοιχεία μιας στοίβας προωθούνται επαυξημένα κατά 1 στον στο προηγούμενο στοιχείο της στοίβας που αντιστοιχεί στον κόμβο πατέρα τους στο δέντρο. Στην Εικόνα 3.12β΄ παρουσιάζονται οι καταστάσεις της αρχικής στοίβας του δικτυωτού για την ερώτηση Q={`XML, John, Brown`} που υποβάλλεται στο δέντρο της Εικόνας 3.12α΄ με κατώφλι T=5.

Η διαφορά είναι, στην περίπτωση του T-LCAsz, ότι σε διάφορα σημεία του αλγορίθμου χρειάζεται να ελέγχεται ότι τα μεγέθη των α) παραγόμενων (μερικών) LCA

και β) των προωθούμενων προς τους προγόνους μερικών LCA, δεν ξεπερνούν το δοσμένο κατώφλι T. Έτσι, στον ψευδοκώδικα 1, όπου παρουσιάζεται το κύριο σώμα του αλγορίθμου T-LCAsz, έχουν προστεθεί οι έλεγχοι μεγέθους μερικών LCA πριν αυτοί εισαχθούν σε μια στοίβα ή πριν προωθηθούν στα επόμενα επίπεδα του δικτυωτού (γραμμές 11 και 15). Τέτοιοι έλεγχοι έχουν προστεθεί στη διαδικασία emptyStacks() αλλά και στην pop() που παρουσιάζεται στον ψευδοκώδικα 1. Το αποτέλεσμα των ελέγχων αυτών στις στοίβες όλων των επιπεδων του δικτυωτού παρουσιάζεται στην Εικόνα 3.13.



(α′) Καταστάσεις της στοίβας του 1ου επιπέδου του δικτυωτού στη διάρκεια της επεξεργασίας του στιγμιοτύπου του `XML, 1.1.2.1`



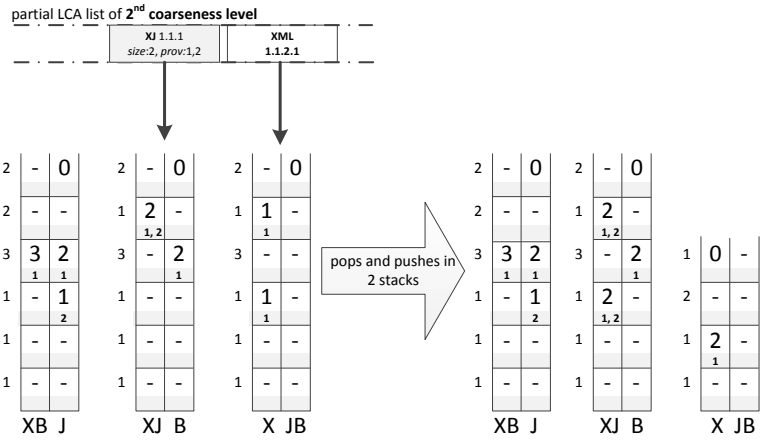
(β′) Καταστάσεις της στοίβας του 2ου επιπέδου του δικτυωτού πριν και μετά την επεξεργασία μερικών LCA και του στιγμιοτύπου του `XML, 1.1.2.1`

Σχήμα 3.13: Οι στοίβες για την ερώτηση Q={`XML, John, Brown`} με κατώφλι T=5 όταν ο T-LCAsz επεξεργάζεται το στιγμιότυπο 1.1.2.1 για το `XML`

Ανάλυση αλγορίθμου Ο αλγόριθμος T-LCAsz παράγει LCAs για ερωτήσεις με λέξεις-κλειδιά ακολουθώντας τα μονοπάτια του δικτυωτού, όπως και ο LCAsz. Το μέγεθος του δικτυωτού δίνεται από τον αριθμό Bell ενός συνόλου με στοιχεία όσες είναι και οι λέξεις-κλειδιά της ερώτησης και ο οποίος για αριθμό λέξεων-κλειδιών k ορίζεται ως εξής:

$$B_{n+1} = \sum_{i=0}^{n} \binom{n}{i} B_i, \; B_0 = B_1 = 1$$

Για κάθε μερικό LCA που παράγεται στη διάρκεια εκτέλεσης του T-LCAsz και σε κάθε στοίβα του δικτυωτού που αυτός εισάγεται μπορεί να χρειαστούν στη χειρότερη



Σχήμα 3.14: Οι στοίβες του 2ου επιπεδου του δικτυωτού για την ερώτηση Q={`XML, John, Brown`} πριν και μετά την επεξεργασία του στιγμιοτύπου 1.1.2.1 για το `XML` από τον topKsz-LCAsz με k=1

περίπτωση d ενέργειες push και d pop, όπου d είναι το βάθος του δένδρου στο οποίο υποβάλλεται η ερώτηση. Ωστόσο, ο αριθμός αυτός των push και pop περιορίζεται και από το κατώφλι T. Κάθε ενέργεα pop συνοδεύεται από $k(k-1)/2$ συνδυασμούς στοιχείων το πολύ, οι οποίοι καταλήγουν σε νέους μερικούς LCA και σε ίσο αριθμό από συγκρίσεις με το κατώφλι T (δηλ. $O(k^2)$). Για κάθε pop , λαμβάνουν χώρα k ενημερώσεις μεγεθών στο προηγούμενο στοιχείο της στοίβας και k συγκρίσεις με το κατώφλι μεγέθους T ($O(k)$). Κατά συνέπεια, αν το μήκος μιας ανεστραμμένης λίστας είναι $|S_i|$, η πολυπλοκότητα του T-LCAsz είναι:

$$O(tk^2 B_k \sum_{i=1}^{k} |S_i|), \quad t = min(d, T)$$

Συνεπώς, ο T-LCAsz είναι γραμμικός σε σχέση με το μήκος της εισόδου όπως και ο LCAsz. Το κατώφλι T περιορίζει το πλάτος και το βάθος των ελάχιστων συνδετικών δέντρων των στιγμιοτύπων των λέξεων-κλειδιών. Αυτή η παρατήρηση οδηγεί σε περιορισμό των (μερικών) LCA και των συνακόλουθα απαιτούμενων ενεργειών στις στοίβες. Χαμηλότερες τιμές κατωφλίου επηρεάζουν τον T-LCAsz απορρίπτοντας μερικούς LCA νωρίς στον υπολογισμό, σε αρχικά επίπεδα του δικτυωτού, μειώνοντας ακόμα πιο πολύ το χρόνο αποτίμησης.

3.5.1.2 Αλγόριθμοι κορυφαίων μεγεθών και κορυφαίων αποτελεσμάτων topKsz-LCAsz

Όταν ένα κατώφλι μεγέθους LCA δεν είναι γνωστό εκ των πρωτέρων, ο αλγόριθμος T-LCAsz δεν μπορεί να χρησιμοποιηθεί. Σ′ αυτήν την ενότητα παρουσιάζουμε τους αλγορίθμους *topKsz-LCAsz* και *topK-LCAsz* οι οποίοι περιορίζουν την απάντηση σε μια ερώτηση, είτε στο σύνολο των LCA με μεγέθη ανάμεσα στα μικρότερα k (top-k-size), είτε στο σύνολο των k μικρότερων LCA (top-k), αντίστοιχα.

Σύμφωνα με τη σημασιολογία TLCA, αποτελέσματα ίσου μεγέθους κατατάσσονται στην ίδια θέση στην ταξινόμηση και γι′ αυτό αξιολογούνται ως ίσης σχετικότητας με την ερώτηση. Οι αλγόριθμοι topKsz-LCAsz και topK-LCAsz παραλλάσσουν τον T-LCAsz, ώστε να μπορεί να λειτουργήσει με ένα αυτόματα προσαρμόζομενο κατώφλι

μεγέθους. Αρχικά το κατώφλι μεγέθους ορίζεται στο άπειρο και για τους δυο αλγορίθμους. Στη συνέχεια ακολουθεί η φάση μάθησης, κατά την οποία το κατώφλι συγκλίνει στο k μικρότερο μέγεθος LCA ή στο μέγεθος του k μικρότερου LCA, αντίστοιχα. Όσο νωρίτερα στον υπολογισμό ο αλγόριθμος καταλήγει στο τελικό κατώφλι T, τόσο αποδοτικότερα γίνεται η περικοπή παραγωγής άσχετων ενδιάμεσων αποτελεσμάτων.

Η συνάρτηση addResult() του T-LCAsz προσθέτει έναν LCA στο σύνολο αποτελεσμάτων (γραμμές 1-6). Στην περίπτωση των top-k αλγορίθμων, η συνάρτηση addResult() προσαρμόζει το κατώφλι μεγέθους T του T-LCAsz (συνάρτηση adjustResultSet(), γραμμή 7) στη διάρκεια της φάσης μάθησης. Το σύνολο αποτελεσμάτων αποτελείται από δοχεία αποτελεσμάτων με το ίδιο μέγεθος. Κάθε νέος LCA προστίθεται στο δοχείο μεγέθους του (συνάρτηση addNode(), γραμμή 6). Αν αυτό το δοχείο δεν υπάρχει, δημιουργείται (συνάρτηση createNewSize(), γραμμή 5).

Μετά την προσθήκη ενός νέου αποτελέσματατος κατά την εκτέλεση του topKsz-LCAsz, αν το σύνολο αποτελεσμάτων αποτελείται από ακριβώς k διαφορετικά δοχεία, το κατώφλι μεγέθους T ορίζεται στο μέγιστο μέγεθος των δοχείων (διαδικασία getLastSize(), γραμμή 12). Διαφορετικά, αν ο νέος LCA προκαλεί τη δημιουργία ενός νέου δοχείου, αυξάνοντας τον αριθμό των δοχείων σε $k+1$, το δοχείο μέγιστου μεγέθους καταστρέφεται, και το κατώφλι T προσαρμόζεται στο νέο μέγιστο μέγεθος.

Procedure pop

```
1  pop(stack)
2      cols = stack.columns
       /* number of kwSubsets in the partition of the stack */
3      popped = stack.pop()
4      if cols = 1 then
           /* addResult() updates T only in top-k mode */
5          T = addResult(stack.dewey, popped[0].size)
       /* Produce new LCAs from two partial LCAs */
6      if cols > 1 then
7          for i=0 to cols do
8              for j=i to cols do
9                  if popped[i] and popped[j] contain sizes and popped[i].provenance ∩
                   popped[j].provenance = ∅ and popped[i].size + popped[j].size ≤ T then
10                     newKwSubset = popped[i].kwSubset ∪ popped[j].kwSubset
11                     newProvenance = popped[i].provenance ∪ popped[j].provenance
12                     newSize = popped[i].size+popped[j].size
13                     createNewStack(stack, stack.coarsenessLevel+1, i, j, newKwSubset)
                           /* if it does not exist */
14                     pLCA = newPartialLCA(stack.dewey, newKwSubset, newSize,
                       newProvenance)
15                     if cardinalityOf(newKwSubset) = stack.coarsenessLevel+1 then
16                         addPartialLCA(stack.coarsenessLevel+1, pLCA)
       /* Update ancestor (i.e., new top entry) with sizes from popped entry */
17     if stack is not empty and cols > 1 then
18         for i=0 to cols do
19             if popped[i].size+1 < stack.topRow[i].size and popped[i].size+1 ≤ T then
20                 stack.topRow[i].size = popped[i].size+1
21                 stack.topRow[i].provenance = {lastStep(stack.dewey)}
22     removeLastDeweyStep(stack.dewey)
```

Function addResult

```
1  addResult(lcaDewey, lcaSize)
2      if results contain lcaSize then
3          sizeBucket = getSize(results, lcaSize)
4      else
5          sizeBucket = createNewSize(results, lcaSize)
6      addNode(sizeBucket, lcaDewey)
7      return adjustResultSet()
8  adjustResultSet() /* topKsz-LCAsz */
9      if countSizes(results) > K then
10         removeMaxSize(results)/* remove max size bucket */
       /* Check if K threshold is reached */
11     if countSizes(results) = K then
12         return getLastSize(results)
13     else
14         return ∞
15 adjustResultSet() /* topK-LCAsz */
16     if count(results) > K then
17         removeLast(results)/* remove an arbitrary dewey from the bucket with
               max size */
       /* Check if K threshold is reached */
18     if count(results) = K then
19         return getLastSize(results)
20     else
21         return ∞
```

Στην περίπτωση του topK-LCAsz, η σύγκλιση του κατωφλίου T ξεκινά όταν έχουν υπολογιστεί τα πτώτα k αποτελέσματα, οπότε το κατώφλι T τίθεται στο μέγιστο μέγεθος των δοχείων (διαδικασία getLastSize(), γραμμή 19). Αν ένας νέος LCA αυξήσει το συνολικό αριθμό αποτελεσμάτων σε $k+1$, ένας τυχαίος LCA αφαιρείται από το δοχείο μέγιστου μεγέθους. Αν το συγκεκριμένο δοχείο απομείνει άδειο, καταστρέφεται.

Το κύριο σώμα του T-LCAsz επίσης προσαρμόζεται για να ενσωματώσει της λειτουργικότητα των topKsz-LCAsz και topK-LCAsz. Δύο επιπλέον έλεγχοι εγγυώνται ότι το κατώφλι δε μειώνεται περαιτέρω ανάμεσα στη δημιουργία ενός LCA και την επιεξεργασία του από τις στοίβες των ακόλουθων επιπέδων του δικτυωτού.

Η Εικόνα 3.14 απεικονίζει πώς ο αλγόριθμος topKsz-LCAsz με k=1 διχειρίζεται την προώθηση μερικών LCA στις στοίβες του δεύτερου επιέδου του δικτυωτού, μετά την επεξεργασία του στιγμιοτύπου 1.1.2.1 για τη λέξη-κλειδί XML. Αυτό δείχνει τη διαφορά με τον T-LCAsz με κατώφλι T=5 της Εικόνας 3.13β′. Η παραγωγή του LCA 1.1.1.3.1 για την ερώτηση {`XML, John, Brown`} με μέγεθος 2, οδηγεί στην προσαρμογή του T στην τιμή 2. Γι΄ αυτό, η παραγωγή του LCA 1.1.1 με μέγεθος 4 αποφεύγεται, και όλοι οι μερικοί LCA με μεγέθγ μεγαλύτερα του 2 απορρίπτονται.

3.5.2 Επίδοση αλγορίθμων φιλτραρίσματος με σημασιολογία TLCA

Η πειραματική μελέτη που ακολουθεί καταδεικνύει την αποδοτικότητα των αλγορίθμων T-LCAsz, topKsz-LCAsz και topK-LCAsz και την αποτελεσματικότητα της σημασιο-

λογίας κορυφαίων k TLCA τόσο σαν προσέγγιση ταξινόμησης όσο και φιλτραρίσματος. Για τα πειράματα, χρησιμοποιήθηκαν, όπως και στην Ενότητα 3.3.3, οι πραγματικές συλλογές δεδομένων DBLP[1] και NASA[2] και η τεχνητά παραγόμενη συλλογή δεδομένων XMark[3], αλλά διαφορετικές εκδόσεις αυτών της προηγούμενης ενότητας. Ο Πίνακας 3.4 παρουσιάζει τα στατιστικά χαρακτηριστικά των δενδρικών αυτών συλλογών δεδομένων. Τόσο τα στοιχεία όσο και τα χαρακτηριστικά των XML εγγράφων των συλλογών αντιμετωπίστηκαν χωρίς διαφοροποίηση ως ξεχωριστοί κόμβοι. Οι ανεστραμμένες λίστες αποθηκεύτηκαν σε μια βάση δεδομένων MySQL. Τα πειράματα διενεργήθηκαν σε ένα σύστημα με λειτουργικό Mac OS Lion με επεξεργαστή 1.8GHz dual core Intel Core i5 και ο κώδικας των αλγορίθμων αναπτύχθηκε σε Java.

	DBLP	*XMark*	*NASA*
size	1,15 GB	116,5 MB	25,1 MB
maximum depth	5	11	7
# nodes	34.141.216	2.048.193	530.528
# keywords	3.403.570	140.425	69.481
# distinct labels	44	77	68
# distinct label paths	196	548	110

Πίνακας 3.4: Στατιστικά συλλογών δεδομένων DBLP, XMark και NASA

queries	Q_1^D			Q_2^D			Q_3^D			Q_4^D			Q_5^D			Q_6^D		
		size	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**
keywords	keyword	0	442	XML	0	92	XML	2	18	XML	2	5	proceedings	2	15.296	proceedings	3	13.803
	search	2	9	keyword	2	2	keyword	4	1	keyword	4	1	publisher	3	4	publisher	4	3
		3	1	search	4	1	search			search			editor			editor		
		4	1				Chen			Yi						ISBN		
										Chen								
# keywords	2			3			4			5			3			4		
total # kw instances	31.987			41.624			119.526			139.600			150.529			180.309		
# results	453			95			19			6			15.300			13.806		

Πίνακας 3.5: Ερωτήσεις στη συλλογή δεδομένων DBLP

queries	Q_1^X			Q_2^X			Q_3^X			Q_4^X			Q_5^X		
		size	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**
keywords	order	0	10,805	order	2	10,787	order	2	8,010	order	3	8,010	order	3	5,236
	payment	1	3	payment	3	8	payment	3	7	payment	4	7	payment	4	2,777
		2	11	shipping	4	154	shipping	4	135	shipping	5	135	shipping	5	139
		3	320		5	135	charges	5	121	charges	6	121	charges	6	67
		4	414		6	122		6	104	location	7	104	location	7	153
		5	354		7	21		7	23		8	23	description	8	28
		6	52		8	76		8	62		9	62		9	62
# keywords	2	7	192	3	9	3	4	9	1	5	10	1	6	10	1
total # kw instances	35.365	8	19	68.134	10	2	90.627	10	3	112.377	11	2	168.558	11	2
# results	12.172	9	2	11.308			8.468	11	2	8.465			8.465		

Πίνακας 3.6: Ερωτήσεις στη συλλογή δεδομένων XMark

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, οι τρεις συλλογές δεδομένων, που χρησιμοποιήθηκαν στα πειράματα, καλύπτουν μια ευρεία γκάμα χαρακτηριστικών ώστε να μελετηθούν οι αλγόριθμοι σε ποικίλες συνθήκες. Το DBLP είναι το μεγαλύτερο σύνολο δεδομένων αλλά τα XMark και το NASA έχουν πολυπλοκότερη δομή, με το NASA να έχει και το μέγιστο αριθμό λέξεων-κλειδιών ανά κόμβο, καθώς περιέχει αρκετό κείμενο.

[1]http://www.informatik.uni-trier.de/ ley/db/

[2]http://www.cs.washington.edu/research/xmldatasets/www/repository.html

[3]http://www.xml-benchmark.org

queries	Q_1^N			Q_2^N			Q_3^N			Q_4^N			Q_5^N		
		size	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**		**size**	**# LCAs**
keywords	stars	0	80	stars	0	2	stars	4	144	stars	4	14	stars	6	2
	name	2	953	name	2	644	name	5	13	name	5	4	name	8	4
		3	134	title	3	125	title	6	186	title	6	37	title	9	23
		4	1423		4	59	description	7	287	description	7	35	description	10	120
		5	701		5	632		8	234	number	8	309	number	11	29
		6	1489		6	876		9	151		9	181	initial	12	32
		7	225		7	491		10	29		10	179		13	232
		8	39		8	250		11	4		11	78		14	163
		10	30		9	268		13	8		12	59		15	155
# keywords	2			3			4			5	13	4	6	16	68
total # kw instances	86.547			100.168			108.894			125.994	14	3	140.711	17	54
# results	5.074			3.347			1.056			907	15	4	883	18	1

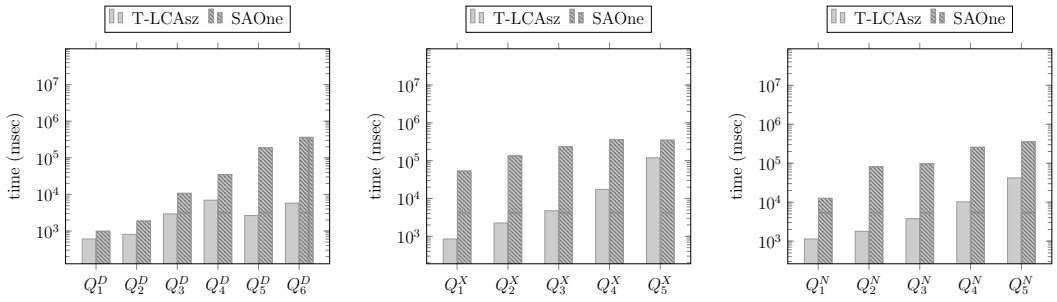
Πίνακας 3.7: Ερωτήσεις στη συλλογή δεδομένων NASA

3.5.2.1 Αποδοτικότητα των αλγορίθμων

Οι Πίνακες 3.5, 3.6 και 3.7 περιέχουν το σύνολο των ερωτήσεων που εκτελέστηκαν για τα πειράματα επίδοσης των αλγορίθμων. Γα κάθε ερώτηση, οι πίνακες εμφανίζουν τις λέξεις-κλειδιά, το συνολικό αριθμό από στιγμιότυπα και το συνολικό αριθμό αποτελεσμάτων για καθεμιά, θεωρώντας ότι δεν περιορίζονται από κάποιο κατώφλι μεγέθους. Για κάθε ερώτηση, φαίνεται επίσης η κατανομή των αποτελεσμάτων ανά μέγεθος LCA. Ο Πίνακας 3.8 δείχνει τον αριθμό από λέξεις-κλειδιά και τον τύπο τους,

DBLP			
keyword	label instances	value instances	total
XML	0	9,637	9,637
keyword	0	1,137	1,137
search	0	30,850	30,850
Yi	0	20,074	20,074
Chen	0	77,902	77,902
proceedings	19,453	52,970	72,423
publisher	28,855	86	28,941
editor	45,823	3,342	49,165
ISBN	27,911	1,869	29,780
XMark			
keyword	label instances	value instances	total
order	0	12,705	12,705
payment	21,750	910	22,660
shipping	21,750	11,019	32,769
charges	0	22,493	22,493
location	21,750	0	21,750
description	44,500	11,681	56,181
NASA			
keyword	label instances	value instances	total
stars	0	11,052	11,052
name	71,688	3,807	75,495
title	13,605	16	13,621
description	8,199	527	8,726
number	0	17,100	17,100
initial	14,512	205	14,717

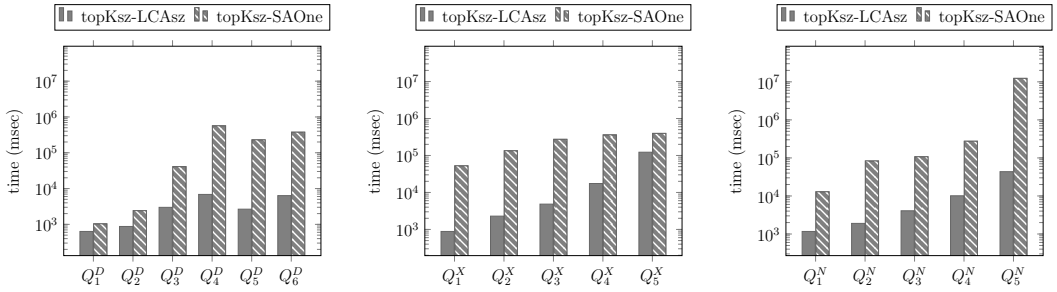
Πίνακας 3.8: Αριθμός και τύπος στιγμιοτύπων των λέξεων-κλειδιών



(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.15: Σύγκριση επίδοσης T-LCAsz και SAOne στον υπολογισμό αποτελεσμάτων όταν το κατώφλι είναι το ελάχιστο μέγεθος

δηλ. αν τα στιγμιότυπά τους εμφανίζονται στην ετικέτα ενός κόμβου ή στο περιεχόμενό του, στο αντίστοιχο δέντρο δεδομένων. Παρουσιάζονται για καλύτερη κατανόηση της δομής των συλλογών δεδομένων, παρόλο που οι αλγόριθμοι που μελετώνται δεν κάνουν κάποιο διαχωρισμό. Διαφορετικές ερωτήσεις με κατώφλι μεγέθους ή κορυφαίων αποτελεσμάτων με τον ίδιο αριθμό λέξεων-κλειδιών επιστρέφουν απαντήσεις με διαφορετικούς αριθμούς αποτελεσμάτων ακόμα κι αν ο συνολικός αριθμός των στιγμιοτύπων των λέξεων-κλειδιών τους είναι παρόμοιος. Για παράδειγμα, οι ερωτήσεις Q_5^X και Q_5^N αποτελούνται από 5 λέξεις-κλειδιά και παρόμοιο συνολικό αριθμό από στιγμιότυπα, αλλά η κατανομή των αποτελεσμάτων τους στα μικρότερα μεγέθη LCA είναι τελείως διαφορετικός.

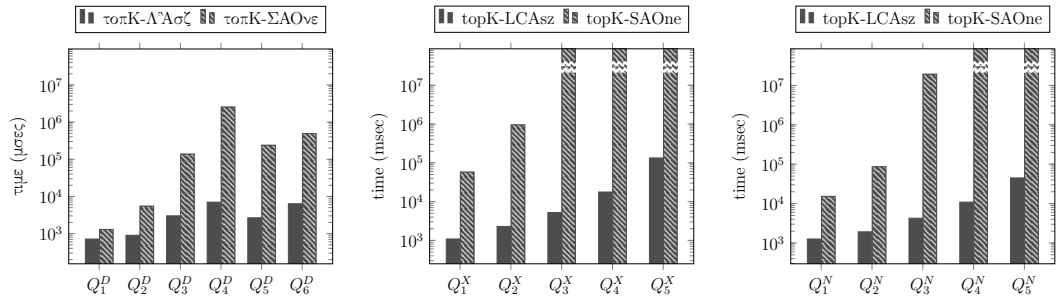


(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.16: Σύγκριση επίδοσης topKsz-LCAsz και topKsz-SAOne στον υπολογισμό αποτελεσμάτων κορυφαίου μεγέθους

3.5.2.2 Σύγκριση επίδοσης

Η επίδοση των τριών αλγορίθμων συγκρίνεται με τον αλγόριθμο SA [21] και τις παραλλαγές του. Όπως έχει ήδη συζητηθεί, ο αλγόριθμος αυτός είναι ο μόνος στη βιβλιογραφία που έχει εγγενώς τη δυνατότητα να υπολογίσει μεγέθη ελάχιστων συνδετικών δέντρων σε δεδομένα XML. Εντοπίζει ελάχιστα συνδετικά δέντρα των στιγμιοτύπων των λέξεων-κλειδιών και τα συνοψίζει παραλείποντας εσωτερικούς κόμβους και θέτοντας βάρη αποστάσεων στις ακμές αντ΄ αυτών, κατασκευάζοντας DMCT, ενώ στη συνέχεια τα ομαδοποιεί σε GDMCT. Ο αλγόριθμος SA δέχεται, επίσης, ως είσοδο κατώφλι μεγέθους, με στόχο να επιστρέψει μόνο GDMCT που δεν το ξεπερνούν. Για τη σύγκριση, χρησιμοποιήθηκε η παραλλαγή SAOne που δε χρειάζεται να πραγματοποιήσει συγχώνευση των DMCT καθώς επιστρέφει μόνο LCA.



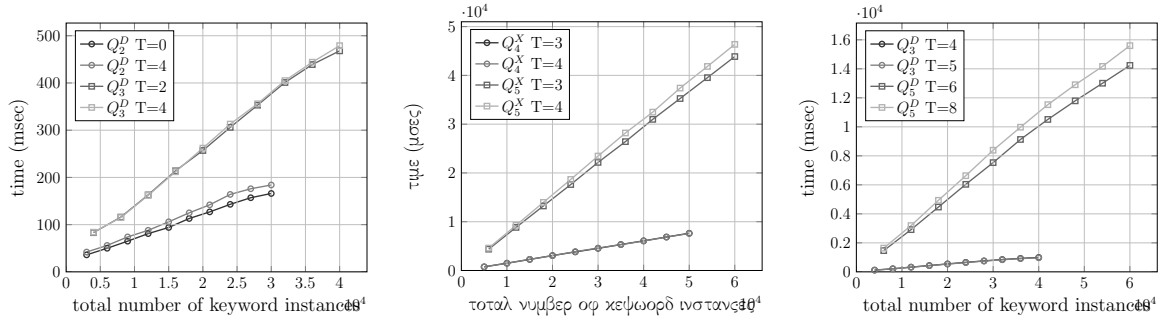
(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.17: Σύγκριση επίδοσης topK-LCAsz και topK-SAOne στον υπολογισμό top-k αποτελεσμάτων με ελάχιστο μέγεθος

Αφήνοντας το μικρότερο συνδετικό δέντρο να συνεισφέρει το μέγεθος του (κατ' αναλογία με τη σημασιολογία μεγέθους), ο αλγόριθμος SAOne προσφέρεται για ευθεία και δίκαιη σύγκριση με τους αλγορίθμους μας, ενώ έχει αποδειχθεί ότι ξεπερνά σε επίδοση την εξαντλητική προσέγγιση που υπολογίζει όλα τα δυνατά ελάχιστα συνδετικά δέντρα των λέξεων-κλειδιών για δεδομένη ερώτηση σε ένα σύνολο δεδομένων [21]. Για τη σύγκριση με τους αλγορίθμους topKsz-LCAsz και topK-LCAsz, επεκτείναμε τον SAOne ώστε να δέχεται την παράμετρο k ως είσοδο και α) να υπολογίζει αποτελέσματα με μέγεθος ανάμεσα στα k κορυφαία (*topKsz-SAOne*) και β) να υπολογίζει τα κορυφαία k αποτελέσματα με βάση το μέγεθος LCA (*topK-SAOne*). Και στις δύο προσεγγίσεις, φροντίσαμε να γίνεται περικοπή DMCT με μεγάλα μεγέθη νωρίς, ώστε να αποφεύγεται ο υπολογισμός όλων των αποτελεσμάτων του βασικού SAOne.

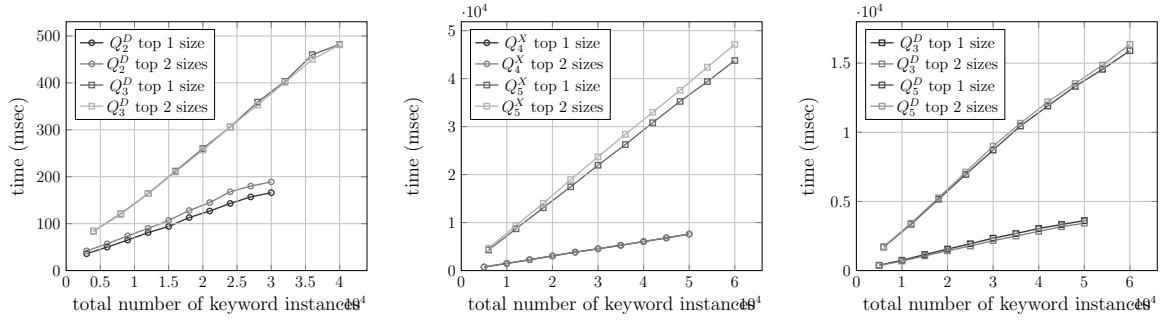
Οι Εικόνες 3.15-3.17 δείχνουν τα αποτελέσματα επίδοσης τριών ζευγαριών αλγορίθμων (δηλ. T-LCAsz έναντι SAOne, topKsz-LCAsz έναντι topKsz-SAOne και topK-LCAsz έναντι topK-SAOne). Ο άξονας y εκφράζει τον καθαρό χρόνο επεξεργασίας των ανεστραμμένων λιστών των λέξεων-κλειδιών αφού έχουν φορτωθεί στη μνήμη. Ο χρόνος φόρτωσης δε λαμβάνεται υπόψιν, καθώς επιβαρύνει ισόποσα όλους τους αλγορίθμους. Η κλίμακα του άξονα y είναι λογαριθμική. Όλοι οι αλγόριθμοι ρυθμίζονται να επιστρέφουν τα αποτελέσματα του μικρότερου μεγέθους σε κάθε ερώτηση των Πινάκων 3.5, 3.6 και 3.7. Για τους αλγορίθμους με κατώφλι, η παράμετρος T τίθεται στο μέγεθος που αντιστοιχεί στους LCA με το μικρότερο μέγεθος (π.χ. T=0 για το Q_1^X, T=2 για το Q_2^X), για τους αλγορίθμους κορυφαίων μεγεθών η παράμετρος $k = 1$ και για τους αλγορίθμους κορυφαίων αποτελεσμάτων η παράμετρος k τίθεται στον αριθμό των αποτελεσμάτων στο μικρότερο μέγεθος LCA ανά ερώτηση (π.χ., k=442 για το Q_1^X, k=92 για το Q_2^X). Όπως φαίνεται στα διαγράμματα, οι αλγόριθμοι T-LCAsz, topKsz-LCAsz και topK-LCAsz ξεπερνούν καθαρά σε επίδοση στον αλγόριθμο SAOne και τις παραλλαγές του σε όλες τις περιπτώσεις. Η διαφορά δεν είναι τόσο μεγάλη στο DBLP για ερωτήσεις με λίγες λέξεις-κλειδιά και μικρό αριθμό από στιγμιότυπα. Ωστόσο, όσο ο αριθμός των λέξεων-κλειδιών και των στιγμιοτύπων τους αυξάνεται η διαφορά στην επίδοση μεταξύ των δύο οικογενειών αλγορίθμων διευρύνεται σημαντικά. Οι αλγόριθμοι T-LCAsz, topKsz-LCAsz και topK-LCAsz δεν επηρεάζονται από το μέγεθος της εισόδου (δηλ. το μήκος των ανεστραμμένων λιστών), όπως αποδεικνύεται και από τους χρόνους εκτέλεσης που αποδεικνύονται κάτω από 10sec στις περισσότερες περιπτώσεις.

Οι αλγόριθμοι κορυφαίων k αποτελεσμάτων (topK-LCAsz και topK-SAOne) εν γένει χρειάζονται περισσότερο χρόνο για την επιστροφή των αποτελεσμάτων, καθώς



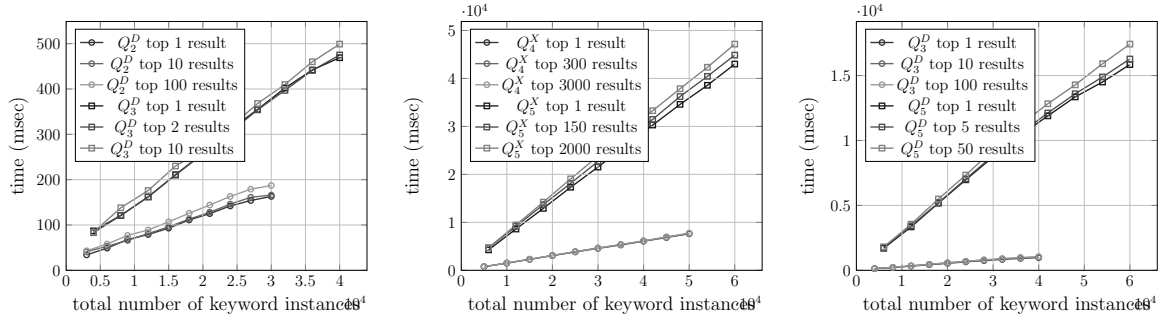
(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.18: Επίδοση T-LCAsz για μεταβλητό αριθμό στιγμιοτύπων (επιστρέφοντας αποτελέσματα με μέγεθος μικρότερο ή ίσο του T)



(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.19: Επίδοση topKsz-LCAsz για μεταβλητό αριθμό στιγμιοτύπων (επιστρέφοντας αποτελέσματα που ανήκουν στα top-k μεγέθη)



(α′) DBLP (β′) XMark (γ′) NASA

Σχήμα 3.20: Επίδοση topK-LCAsz για μεταβλητό αριθμό στιγμιοτύπων (επιστρέφοντας top-k αποτελέσματα)

το κατώφλι μεγέθους μπορεί να μην επιτύχει σύγκλιση προς την τελική του τιμή αρκετά νωρίς στην επεξεργασία, παράγοντας στο μεταξύ αρκετά αποτελέσματα που ξεπερνούν αυτό το κατώφλι. Για τα σύνολα δεδομένων XMark και NASA, ο topK-SAOne αποτυγχάνει να επιστρέψει στην πράξη κορυφαία k αποτελέσματα για ερωτήσεις με περισσότερες από 3 και 4 λέξεις-κλειδιά, αντίστοιχα. Αντίθετα, ο topK-LCAsz διαχειρίζεται αποδοτικά τη σύγκλιση του μεγέθους LCA και επιδεικνύει πολύ καλή επίδοση.

3.5.2.3 Κλιμάκωση

Για τα πειράματα μελέτης της κλιμάκωσης των αλγορίθμων σε σχέση με το μέγεθος της εισόδου, εκτέλεστηκαν οι τρεις αλγόριθμοι με είσοδο τις ανεστραμμένες λίστες των λέξεων-κλειδιών των ερωτήσεων των Πινάκων 3.5-3.7. Οι ανεστραμμένες λίστες περικόπηκαν σε διάφορα μήκη για την επίτευξη της μεταβολής της εισόδου. Η δυνατότητα αυτή προσφέρεται από λέξεις-κλειδιά μεγάλης συχνότητας, οι οποίες επιλέχθηκαν στις περιπτώσεις αυτές, ώστε, μεταξύ άλλων, να δοκιμαστεί η αντοχή των αλγορίθμων από την παραγωγή μεγάλου αριθμού μερικών LCA κατά την αποτίμηση. Κάθε καμπύλη σχηματίζεται από τις μέσες τιμές χρόνων ακτέλεσης δέκα επαναλήψεων ανά ερώτηση. Ο χρόνος φόρτωσης των ανεστραμμένων λιστών στη μνήμη δε λαμβάνεται υπόψιν.

Οι Εικόνες 3.18, 3.19 και 3.20 επιδεικνύουν πώς κλιμακώνοναται οι αλγόριθμοι T-LCAsz, topKsz-LCAsz και topK-LCAsz καθώς αλλάζουν τα μήκη των ανεστραμμένων λιστών, δηλ. το μέγεθος της εισόδου, σε κάθε περίπτωση. Οι λίστες περικόπηκαν ώστε να περιέχουν από 1000 έως 10.000 στιγμιότυπα ανά λέξη-κλειδί. Για λόγους καθαρότητας των διαγραμμάτων, δύο ερωτήσεις απεικονίζονται ανά αλγόριθμο και σύνολο δεδομένων, παρόλο που και για τις υπόλοιπες ερωτήσεις τα αποτελέσματα είναι ανάλογα.

Για κάθε ερώτηση και αλγόριθμο, παρουσιάζονται δύο ή τρεις καμπύλες που αντιστοιχούν σε διαφορετικές τιμές των παραμέτρων T και k. Για τον T-LCAsz το κατώφλι μεγέθους T ορίζεται στις δύο τιμές που αντιστοιχούν στα μικρότερα μεγέθη LCA της ερώτησης (βλ. Πίνακα 3.5-3.7). Ανάλογα ορίζεται η παράμετρος k για τους topKsz-LCAsz και topK-LCAsz. Στην πρώτη περίπτωση, ορίζεται στις τιμές 1 και 2 για επιστρεφή των αποτελεσμάτων των κορυφαίων 1 ή 2 μεγεθών LCA. Στη δεύτερη περίπτωση, ο αριθμός των αποτελεσμάτων εξαρτάται από το μήκος των ανεστραμμένων λιστών. Έτσι, σε κάθε περίπτωση ερώτησης για τον topK-LCAsz, οι τιμές που επιλέχθηκαν για την παράμετρο k ήταν 1 (για το κορυφαίο αποτέλεσμα) και δύο ακόμα τιμές που κατά προσέγγιση αντιστοιχούν στον αριθμό των αποτελεσμάτων των κορυφαίων 1 ή 2 μεγεθών LCA.

Οι ερωτήσεις των διαγραμμάτων κλιμάκωσης καλύπτουν ένα ευρύ φάσμα χαρακτηριστικών: τα Q_2^D και Q_3^D δίνουν σαν αποτέλεσμα λίγους LCA σε σχέση με τον αριθμό στογμιοτύπων των λέξεων-κλειδιών τους και σχεδόν όλοι ανήκουν στο μικρότερο μέγεθος. Αντιθέτως, τα Q_4^X και Q_5^X επιστρέφουν πολλούς LCA σε σχέση με τα μήκη των ανεστραμμένων λιστών τους. Τα περισσότερα απ΄ αυτά τα αποτελέσματα κατανέμονται στο κορυφαίο (Q_4^X) ή στα κορυφαία δύο (Q_5^X) μεγέθη. Τέλος, οι λέξεις-κλειδιά στις ερωτήσεις του συνόλου δεδομένων NASA δε είναι τόσο συσχετισμένες καθώς επιστρέφουν λίγα αποτελέσματα σε σχέση με τις εμφανίσεις τους στο δέντρο δεδομένων.

Για δεδομένο αριθμό από λέξεις-κλειδιά, η κλιμάκωση που καταδεικνύεται από τις καμπύλες μαρτυρά τη γραμμική συμπεριφορά των T-LCAsz, topKsz-LCAsz και topK-LCAsz σε συνάρτηση με το μέγεθος της εισόδου. Αυτή η παρατήρηση συμφωνεί και με την ανάλυση της πολυπλοκότητάς τους που παρουσιάζεται στην Ενότητα 3.5.1.1. Οι αλγόριθμοι κλιμακώνονται ομαλά σε σχέση με το μήκος των ανεστραμμένων λιστών, ανεξαρτήτως του συνόλου δεδομένων στο υποβάλλεται μια ερώτηση και του αριθμού των λέξεων-κλειδιών.

3.5.2.4 Σύγκριση των τριών αλγορίθμων φιλτραρίσματος

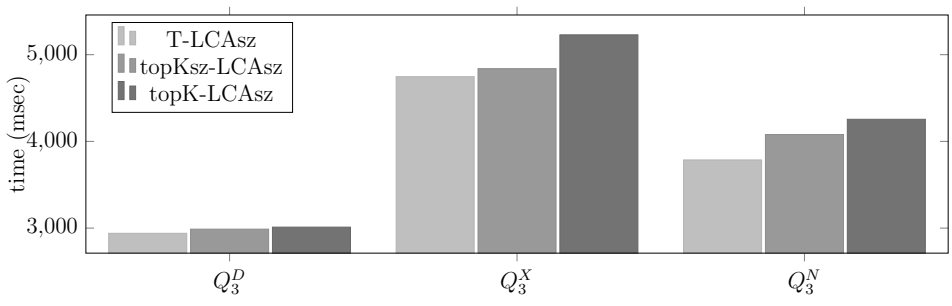
Η Εικόνα 3.21 απεικονίζει τη σχετική επίδοση των T-LCAsz, topKsz-LCAsz και topK-LCAsz, όταν υπολογίζουν τα αποτελέσματα μικρότερου μεγέθους μιας ερώτησης. Για

παράδειγμα, στην περίπτωση του Q_3^D, T=2 για τον T-LCAsz, k=1 για τον topKsz-LCAsz και k=12 για τον topK-LCAsz. Οι ερωτήσεις Q_3^D, Q_3^X και Q_3^N επιλέχθηκαν δειγματοληπτικά από κάθε σύνολο δεδομένων για τη σύγκριση.

Μια γενική παρατήρηση είναι ότι ο T-LCAsz είναι ταχύτερος από τους άλλους δύο αλγορίθμους. Αυτό είναι αναμενόμενο καθώς στην περίπτωση του T-LCAsz, το κατώφλι μεγέθους T είναι γνωστό εκ των πρωτέρων και δε χρειάζεται ο αλγόριθμος να διατρέξει τη φάση εκμάθησης. Οι άλλοι δύο αλγόριθμοι διαφέρουν λιγότερο ή περισσότερο ανάλογα με την χαρακτηριστικά του συνόλου δεδομένων και της συγκεκριμένης ερώτησης. Στις περισσότερες περιπτώσεις, ο topKsz-LCAsz συγκλίνει στο τελικό κατώφλι μεγέθους πολύ γρηγορότερα από τον topK-LCAsz. Αυτό διακιολογείται από το γεγονός ότι για τον topKsz-LCAsz αρκεί μόνο ένα αποτέλεσμα μικρότερου μεγέθους να παραχθεί, ώστε να να προσδιορισθεί το μικρότερο μέγεθος, τη στιγμή που ο topK-LCAsz χρειάζεται k αποτελέσματα για να επιτύχει την ίδια σύγκλιση. Για παράδειγμα, στην εκτέλεση της ερώτησης Q_3^X ο topKsz-LCAsz (με k=1) προσδιορίζει το μικρότερο μέγεθος 2 αφού έχει επεξεργαστεί τη 49ή λέξη-κλειδί από τις 90.627, ενώ ο topK-LCAsz (με k=8.010) συμπληρώνει τις θέσεις των 8.010 αποτελεσμάτων μικρότερου μεγέθους αφού έχει επεξεργαστεί το σύνολο των στιγμιοτύπων (δηλ. μετά το 90.627ο στιγμιότυπο). Όπως φαίνεται, λοιπόν, στην Εικόνα 3.21, σάυτήν την περίπτωση ο topKsz-LCAsz απαντά στην ερώτηση σχεδόν όσο γρήγορα και ο T-LCAsz ενώ ο topK-LCAsz υστερεί σημαντικά.

3.5.3 Αποτελεσματικότητα σημασιολογίας κλιμακωτής ταξινόμησης και φιλτραρίσματος TLCA

Σ′ αυτήν την ενότητα παρουσιάζεται η μελέτη αποτελεσματικότητας τόσο της κλιμακωτής ταξινόμησης όσο και του φιλτραρίσματιος με βαση τη σημασιολογία TLCA. Στα πειράματα αυτής της μελέτης, χρησιμοποιήθηκαν οι συλλογές δεδομένων DBLP και NASA που είναι πραγματικά σύνολα δεδομένων αλλά ταυτόχρονα έχουν διαφορετικά χαρακτηριστικά (βλ. Πίνακα 3.1). Ο Πίνακας 3.9 περιλαμβάνει τις ερωτήσεις που υποβλήθηκαν σε καθένα από τα δύο σύνολα δεδομένων. Ο αλγόριθμος topKsz-LCAsz ρυθμίστηκε να επιστρέφει τους top-2-size LCA, δηλ. τα αποτελέσματα των μικρότερων δύο μεγεθών. Οι LCA επιστρέφονται ομαδοποιημένοι σε δύο κλίμακες: μικρότερου μεγέθους (top-1-size) LCA και 2ου μικρότερου μεγέθους. Η σχετικότητα των αποτελεσμάτων με την ερώτηση, τόσο με βαθμολογία όσο και δυαδική (δηλ. σωστό λάθος) προσφέρθηκε από ένα σύνολο εξιδικευμένων χρηστών. Για τη βαθμολογία, χρησιμοποιήθηκε μια κλιμακα 4 τιμών, με το 0 να υποδηλώνει πως ένα αποτέλεσμα δεν έχει καμία σχέση με την ερώτηση. Για την υποβοήθηση των ειδικών στη διαδικασία, μέσω



Σχήμα 3.21: Σύγκριση επίδοσης των τριών top-k μεθόδων στον υπολογισμό των top-1-size αποτελεσμάτων

της αποφυγής αξιολόγησης καθενός LCA ξεχωριστά, το οποίο είναι πρακτικά αδύνατο για πολύ μεγάλο αριθμό αποτελεσμάτων, δόθηκαν στους ειδικούς τα δενδρικά πρότυπα των απαντήσεων. Μέσω αυτών οι μεμονωμένοι LCA ομαδοποιήθηκαν και ταυτόχρονα οι χρήστες μπορούσαν να αποκτήσουν διαίσθηση για τη φύση του αποτελέσματος. Τα δενδρικά πρότυπα απαντήσεων (βλ. Κεφάλαιο 4) καταδεικνύουν πώς συνδέονται τα στιγμιότυπα των λέξεων-κλειδιών στο υποδέντρο ενός LCA για να σχηματίσουν το ελάχιστον συνδετικό δέντρο τους, αλλά και πώς ο LCA συνδέεται με τη ρίζα του δέντρου δεδομένων. Ως βαθμολογία για έναν LCA λήφθηκε η μέγιστη βαθμολογία των δεδνδρικών προτύπων τα οποία απαντήθηκαν στο υποδέντρο του.

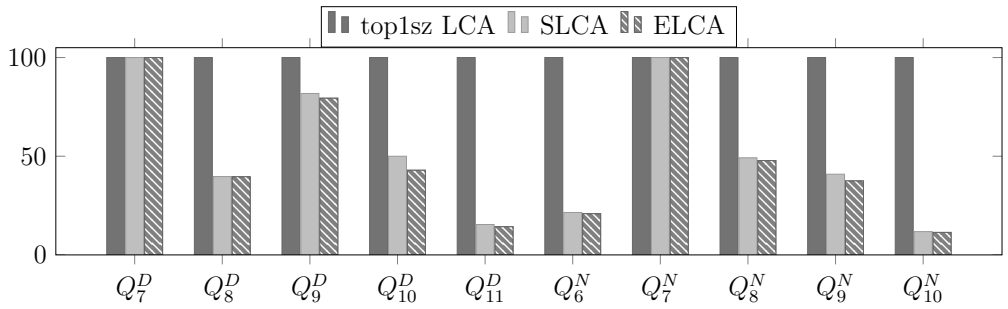
3.5.3.1 Προσέγγιση κλιμακωτού φιλτραρίσματος TLCA σημασιολογίας

Για να αξιολογηθεί η αποτελεσματικότητα του φιλτραρίσματος με βάση τη σημασιολογία TLCA χρησιμοποιήθηκε η λογική επιλογής κορυφαίου μεγέθους LCA για μεγιστοποίηση της ακρίβειας της απάντησης μιας ερώτησης και η λογική επιλογής των κορυφαίων δύο μεγεθών για μεγιστοποίηση της πληρότητας. Η σημασιολογία TLCA σθγκρίνεται με αυτές των SLCA και ELCA. Οι σημασιολογίες αυτές, όπως και η σημασιολογία TLCA χρησιμοποιούν μόνο δομικές πληροφορία για να επιλέξουν τα σχετικά με την ερώτηση αποτελέσματα. Οι άλλες προσεγγίσεις χρησιμοποιούν επιπλέον σημασιολογική πληροφορία (π.χ. τις ετικέτες των κόμβων, πληροφορία για το σχήμα της συλλογής δεδομένων κ.α.) και πολλές φορές απαιτούν και τη χρήση επιπλέον δομών ευρετηρίου που χρειάζονται προεπεξεργασία των ανεστραμμένων λιστών [13, 26, 31]. Η σύγκριση βασίζεται στις ευρέως διαδομένες μετρικές της ακρίβειας (*precision (P)*), της πληρότητας (*ρεςαλλ (P)*) και του $\mathcal{F}$-measure= $\frac{2P\times R}{P+R}$ [4]. Τα αποτελέσματα της σύγκρισης φαίνονται στην Εικόνα 3.22.

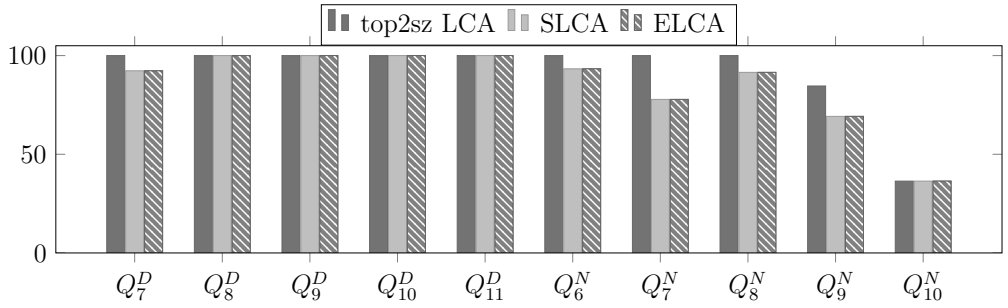
Τα top-1-size TLCA αποτελέσματα επιδεικνύουν τέλεις ακρίβεια για όλες τις ερωτήσεις σε όλα τα σύνολα δεδομένων. Η ακρίβεια των υπόλοιπων προσεγγίσεων μεταβάλλεται και στις περισσότερες περιπτώσεις μένει κάτω από το 50%. Όλες οι προσεγγίσεις αγγίζουν υψηλές τιμές πληρότητας στη συλλογή δεδομένων DBLP, καθώς η ρηχότητά του δεν αφήνει πολλά περιθώρια για απόρριψη σωστών αποτελεσμάτων. Παρ΄ όλ΄ αυτά, στο NASA η προσέγγιση top-1-size TLCA επιτυγχάνει μεγαλύτερη πληρότητα σε όλες τις ερωτήσεις. Ο Πίνακας 3.22γ΄ δείχνει τις μέσες τιμές $\mathcal{F}$-measure για όλες τις ερωτήσεις ανά σύνολο δεδομένων. Το $\mathcal{F}$-measure συνδυάζει ακρίβεια και πληρότητα σε ένα μέγεθος, και, όπως είναι φανερό, οι προσεγγίσεις top-1-size TLCA

DBLP	Q_7^D	neural networks algorithms
	Q_8^D	author Yi Chen
	Q_9^D	Yi Chen XML
	Q_{10}^D	editor Vijay year 2012
	Q_{11}^D	best paper award year 2012
NASA	Q_6^N	observed stars classification
	Q_7^N	sun astrolabe santiago
	Q_8^N	photometric measurements
	Q_9^N	ccd photometric system magnitudes
	Q_{10}^N	galaxies clusters optical observations

Πίνακας 3.9: Ερωτήσεις στα DBLP και NASA για την πειραματική μελέτη της αποτελεσματικότητας της σημασιολογίας TLCA



(α′) Precision %



(β′) Recall %

	DBLP	NASA
top1sz	86,15	79,13
top2sz	72,52	87,33
SLCA	67,23	51,15
ELCA	65,25	50,10

(γ′) $\mathcal{F}$-measure %

Σχήμα 3.22: Ακρίβεια, πληρότητα και $\mathcal{F}$-measure των σημασιολογιών φιλτραρίσματος TLCA (top-1-size και top-2-size), SLCA και ELCA

και top-2-size TLCA επιτυγχάνουν καλύτερες τιμές απ΄ ό,τι οι SLCA και ELCA.

3.5.3.2 Προσέγγιση κλιμακωτής ταξινόμησης TLCA σημασιολογίας

Η αξιολόγηση της σημασιολογίας ταξινόμησης TLCA έγινε με χρήση του μέγέθους Mean Average Precision (MAP) [4] και του Normalized Discounted Cumulative Gain (NDCG) [4] για τις ερωτήσεις του Πίνακα 3.9. Το μέγεθος MAP είναι η μέση τιμή των επιμέρους τιμών ακρίβειας ενός ταξινομημένου συνόλου αποτελεσμάτων κάθε φο-

DBLP			
MAP (%)		NDCG (%)	
top1sz	top2sz	top1sz	top2sz
91	100	100	100

NASA			
MAP (%)		NDCG (%)	
top1sz	top2sz	top1sz	top2sz
89,8	96,8	100	100

Πίνακας 3.10: Αποτελεσματικότητα ταξινόμησης TLCA σημασιολογίας

ρά που ένα σωστό αποτέλεσμα προστίθεται στην απάντηση μιας ερώτησης. Αν ένα σωστό αποτέλεσμα δεν επιστραφεί, η συνεισφορά του ισούται με 0. Το μέγεθος MAP τιμωρεί επιβαρύνει την αξιολόγηση ενός αλγορίθμου όταν σωστά αποτελέσματα δεν επιστρέφονται καθόλου ή όταν λάθος αποτελέσματα επιστρέφονται ψηλά στην ταξινόμηση. Δεδομένης μιας θέσης στην ταξινόμηση, το μέγεθος Discounted Cumulative Gain (DCG) ορίζεται ως το άθροισμα των βαθμολογιών των αποτελεσμάτων μιας απόάντηση έως αυτήν τη θέση, διαιρεμένο με το λογάριθμο της θέσης αυτής. Το διάνυσμα DCG μιας απάντησης είναι το διάνυσμα των επιμέρους DCG τιμών των αποτελεσμάτων, με την κάθε τιμή να καταλαμβάνει στο διάνυσμα τη θέση του στην κατάταξη. Στη συνέχεια, το διάνυσμα NDCG παράγεται από την κανονικοποίηση του διανύσματος DCG με το διάνυσμα της ιδεατής, τέλειας κατάταξης (δηλ. αυτή που ακολουθεί την ταξινόμηση που ορίζουν οι βαθμολογίες των ειδικών). Το NDCG επιβαρύνει έναν αλγόριθμο όταν ευνοεί αποτελέσματα με χαμηλές βαθμολογίες έναντι αυτών με υψηλές στην ταξινόμηση.

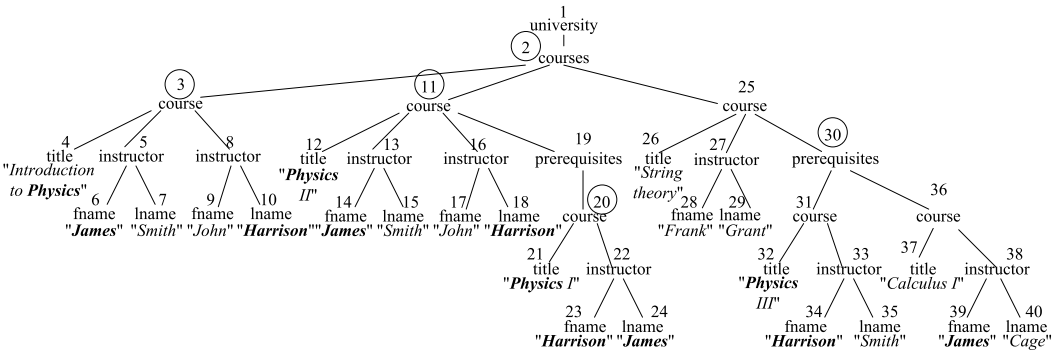
Ο Πίνακας 3.10 δείχνει τις τιμές των μεγεθών MAP και NDCG των top-1-size και top-2-size προσεγγίσεων για τις ερωτήσεις του Πίνακα 3.9. Επειδή για τη σημασιολογία TLCA η σειρά των αποτελεσμάτων ίδιου μεγέθους δεν έχει σημασία, η ταξινόμηση αποτελεσμάτων έγινε με δύο τρόπους: α) σύμφωνα με τη βαθμολογία των ειδικών και β) αντίθετα από τη βαθμολογία των ειδικών χρηστών. Τόσο για την καλύτερη όσο και για τη χειρότερη ταξινόμηση με βάση τη βαθμολογία των ειδικών, υπολογίστηκαν και τα δύο μεγέθη. Κάθε τιμή που εμφανίζεται στον Πίνακα 3.10 είναι η μέση τιμή της βέλτιστης και της χειρότερης τιμής για το αντίστοιχο μέγεθος.

Το τέλειο NDCG και στα δύο σύνολα δεδομένων επιβεβαιώνει τη διαισθητική εκτίμηση για τη σημασιολογία TLCA, καθώς υποδηλώνει πως τα αποτελέσματα ελάχιστου μεγέθους βαθμολογούνται πράγματι ψηλά και από τους ειδικούς. Η τέλεια τιμή επίσης του NDCG για τα αποτελέσματα των μικρότερων δύο μεγεθών δείχνει ότι η ταξινόμηση κατά αύξηση σειρά μεγέθους LCA συμφωνεί με την τέλεια ταξινόμηση των αποτελεσμάτων. Οι περισσέτερες τιμές για το μέγεθος MAP είναι ελαφρώς κατάωτερες του 100%. Αυτό συμβαίνει διότι ένας μικρός αριθμός σωστών LCA απορρίπτεται από το top-1-size ή το top-2-size φιλτράρισμα. Η υψηλή τιμή NDCG, ωστόσο, αποδεικνύει ότι τα αποτελέσματα που απορρίπτονται παρότι σωστά, δεν τους έχει αποδοθεί υψηλή βαθμολογία από τους ειδικούς.

Κεφάλαιο 4

Συλλογισμός σε δενδρικά πρότυπα απαντήσεων

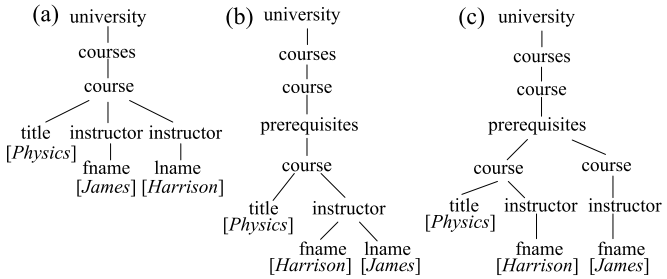
Προχωρώντας τη διερεύνηση στο πρόβλημα της αναζήτησης βέλτιστων αποτελεσμάτων για ερωτήσεις λέξεων - κλειδιών σε δενδρικά δεδομένα, εστιάσαμε στα δενδρικά πρότυπα των απαντήσεων. Κάθε αποτέλεσμα χαρακτηρίζεται από το χαμηλότερο κοινό πρόγονο κάποιων (LCA) στιγμιοτύπων των λέξεων - κλειδιών μιας ερώτησης. Ο κόμβος LCA είναι η ρίζα ενός τουλάχιστον συνδετικού δέντρου των στιγμιοτύπων. Η δομή αυτού του δέντρου δίνει πληροφορίες για τον τύπο του αποτελέσματος. Πολλά ελάχιστα συνδετικά δέντρα μπορεί να έχουν κοινή ρίζα, αλλά και πολλοί χαμηλότεροι κοινοί πρόγονοι είναι δυνατόν να αποτελούν ρίζες ελάχιστων συνδετικών δέντρων που έχουν όμοια δομή.



Σχήμα 4.1: Παράδειγμα δέντρου βάσης δεδομένων πανεπιστημιακών μαθημάτων

Έστω το δέντρο δεδομένων του Σχήματος 4.1 και η ερώτηση {Physics, James, Harrison}. Οι κόμβοι course (3), course (11), course (20) και prerequisites (30) αποτελούν χαμηλότερους κοινούς προγόνους στιγμιοτύπων των λέξεων - κλειδιών. Τα ελάχιστα συνδετικά υποδέντρα των κόμβων course (3) και course (11) έχουν ακριβώς την ίδια δομή. Επιπλέον, την ίδια δομή έχει και το μονοπάτι που συνδέει τη ρίζα τους με τη ρίζα ολόκληρου του δέντρου. Ακόμα ένα στοιχείο είναι πως τα στιγμιότυπα των ίδιων λέξεων - κλειδιών εντοπίζονται στις ίδιες θέσεις και στα δύο αυτά υποδέντρα. Λαμβάνοντας αυτές τις παραμέτρους υπόψιν, μπορούμε να καταλήξουμε ότι τα δύο υποδέντρα είναι ισοδύναμα.

Με βάση τα παραπάνω, ορίζουμε ως πρότυπο ενός υποδέντρου της βάσης δεδομένων, ένα δέντρο ίδιας δομής με το εν λόγω υποδέντρο και επιπλέον με τις επισημειώσεις των λέξεων κλειδιών στους κόμβους όπου εντοπίζονται. Στο Σχήμα 4.2 φαίνονται τρία



Σχήμα 4.2: Δενδρικά πρότυπα των λέξεων-κλειδιών *Physics, James, Harrison* στο δέντρο του σχήματος 4.1.

δενδρικά πρότυπα της ερώτησης {Physics, James, Harrison} στο δέντρο του Σχήματος 4.1.

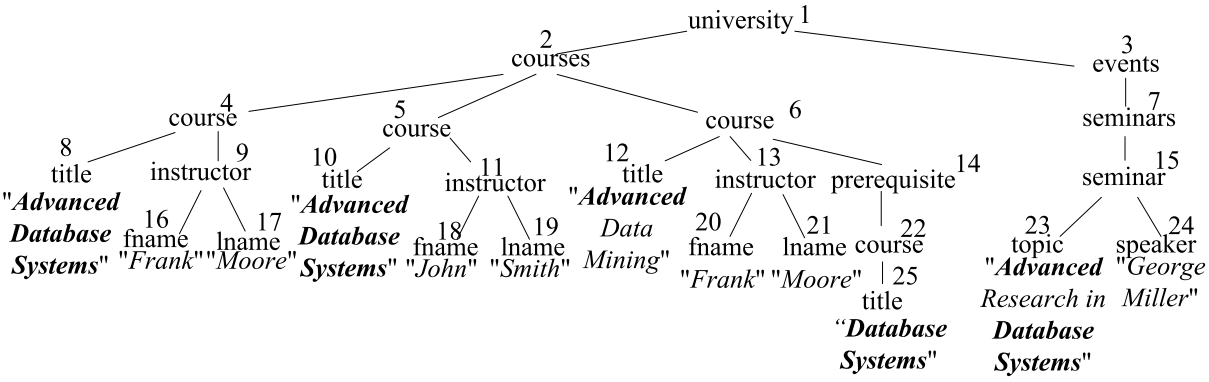
Στόχος του συλλογισμού σε δενδρικά πρότυπα, είναι να οριστεί ο τρόπος που αυτά μπορούν να ταξινομηθούν με βαθμό σχετικότητας στη δεδομένη ερώτηση. Μια τέτοια ταξινόμηση θα οδηγήσει και σε αντίστοιχη κατάταξη τα ίδια τα αποτελέσματα. Έτσι, στο Σχήμα 4.2 μπορεί κανείς εύκολα να συμπεράνει ότι το πρότυπο c είναι χειρότερο από τα a και b μια και ενοποιεί τις λέξεις-κλειδιά από δύο διαφορετικά μαθήματα αντί για ένα. Συνεπώς, τα αποτελέσματα course (3), course (11) και course (20) πρέπει να βαθμολογηθούν υψηλότερα από το prerequisites (30).

Το σύστημά μας προχωρά σε τέτοιου είδους συμπεράσματα αξιοποιώντας δύο είδη ομομορφισμών ανάμεσα σε δενδρικά πρότυπα και τις σχέσεις που οι ομορφισμοί αυτοί ορίζουν. Οι σχέσεις αυτές ορίζουν μια αυστηρή μερική διάταξη των δενδρικών προτύπων. Χρησιμοποιώντας αυτή τη διάταξη κατασκευάζουμε ένα γράφο των δενδρικών προτύπων. Ο γράφος αυτός αποτελεί τη βάση της κατάταξης των δενδρικών προτύπων και κατ΄ επέκταση των αποτελεσμάτων που αντιστοιχούν σ΄ αυτά.

Η προσέγγισή μας είναι μια πολυεπίπεδη μεθοδολογία συσταδοποίησης των αποτελεσμάτων μιας ερώτησης με λέξεις-κλειδιά σε δενδρικά δεδομένα. Τα αποτελέσματα συσταδοποιούνται σε τρία διαφορετικά επίπεδα. Οι συστάδες σε κάθε επίπεδο είναι εμφωλευμένες στις συστάδες του ανώτερου επιπέδου, σχηματίζοντας έτσι μια ιεραρχία από συστάδες. Για κάθε συστάδα, επιλέγεται ο κατάλληλος αντιπρόσωπος που προσφέρεται για εξέταση από το χρήστη κατά την περιήγηση στην ιεραρχία και ο οποίος περιγράφει την στίστοιχη συστάδα επαρκώς. Στο λεπτομερέστερο επίπεδο, οι συστάδες αντιστοιχούν σε πρότυπα των αποτελεσμάτων. Οι συστάδες του τρίτου επιπέδου, αλλά και οι συστάδες του δευτέρου που βρσίκονται εμφωλευμένες σε μια κοινή συστάδα του τρίτου, οργανώνονται σε γράφους. Οι ακμές ορίζουν μια μερική διάταξη των συστάδων. Η μερική διάταξη των συστάδων σε συνδυασμό με την εμφώλευσή τους αξιοποιούνται για την ταξινόμηση των συστάδων. Ο χρήστης περιηγείται στο γράφο των συστάδων, ξεκινώντας από τις γενικότερες, και τελικά κατεβαίνοντας στα τρία επίπεδα καταλήγει στις απαντήσεις της ερώτησης.

4.1 Θεωρητικό υπόβαθρο

Αναπαράσταση δεδομένων. Η αναπαράσταση των δεδομένων XML γίνεται με δέντρα όπου οι κόμβοι τους φέρουν ετικέτες και είναι δυνατόν να περιλαμβάνουν περιεχόμενο σε μορφή κειμένου. Οι κόμβοι αντιστοιχούν σε στοιχεία ή χαρακτηριστικά των στοιχείων και οι ακμές υποδηλώνουν σχέσεις πατέρα-παιδιού μεταξύ στοιχείων ή συνδέχουν ένα στοιχείο με τα χαρακτηριστικά του. Οι λέξεις-κλειδιά μιας ερώτησης



Σχήμα 4.3: Ένα δέντρο δεδομένων T.

μπορούν να εντοπιστούν είτε στις ετικέτες των κόμβων είτε στο περιεχόμενό τους. Ένας κόμβος n, λοιπόν, μπορεί να *περιέχει* μία λέξη κλειδί k αν ο κόμβος n περιέχει τη λέξη-κλειδί k στην ετικέτα του ή στο περιεχόμενό του. Στην περίπτωση αυτή ο κόμβος n λέγεται επίσης και στιγμιότυπου του k.

Σημασιολγία ερωτήσεων. Μία *ερώτηση* Q είναι ένα σύνολο από λέξεις-κλειδιά $\{k_1, k_2, \ldots, k_n\}$. Ορίζεται η ένθεση μιας ερώτησης σε ένα δέντρο δεδομένων.

Ορισμός 4.1 (Στιγμιότυπο ερώτησης). *Ένα* στιγμιότυπο *μια ερώτησης Q σε ένα δέντρο T είναι μια ένθεση της ερώτησης Q στο T (δηλ.,μία συνάρτηση από το Q στους κόμβους του T που αντιστοιχίζει κάθε λέξη-κλειδί k της ερώτησης Q σε ένα στιγμιότυπο του k στο T).*

Το στιγμιότυπο I μιας ερώτησης Q αναφέρεται επίσης στα στιγμιότυπα των λέξεων-κλειδιών του Q, δηλ. αποτελούν το πεδίο τιμών της συνάρτησης I. Οι κόμβοι του στιγμιοτύπου μιας ερώτησης σχηματίζουν ένα υποδέντρο του T. ςοννεςτεδ το φορμ α τρεε.

Ορισμός 4.2 (Στιγμιότυπο δέντρου, IT). *Έστω Q μια ερώτηση, T ένα δέντρο δεδομένων και I ένα στιγμιότυπο της ερώτησης Q στο T. Το* στιγμιότυπο δέντρου (instance tree, IT) *του I είναι το ελάχιστο υποδέντρο S του T τέτοιο ώστε: α) η ρίζα του S ταυτίζεται με τη ρίζα του T και περιλαμβάνει όλους τους κόμβους του I, και β) κάθε κόμβος n στο S επισημειώνεται με τις λέξεις-κλειδιά που αντιστοιχίζεται από το I στο n. Το* ελάχιστον συνδετικό υποδέντρο (MCT) *του I είναι το ελάχιστο υποδέντρο του S που περιέχει όλους τους κόμβους του I. Η ρίζα του MCT είναι ο* χαμηλότερος κοινός πρόγονος *(LCA) των κόμβων του I στο T.*

Έστω το δέντρο της Εικόνας **;;** και η ερώτηση $Q = \{Advanced\, Database\, Systems\}$. Οι Εικόνες 4.4(α) και (β) δείχνουν το IT και το MCT, αντίστοιχα, του στιγμιοτύπου $\{(Advanced,\ 12)\ (Database,\ 25)\ (Systems,\ 25)\}$ τηω Q στο T. Στις εικόνες, η επισημείωση των κόμβων με τις λέξεις-κλειδιά της ερώτησης που περιέχουν, περικλείεται σε αγκύλες πλάι στους αντίστοιχους κόμβους. Είναι φανερό ότι οι δομές των MCT και IT αναπαριστούν εκτός από δοκιμική πληροφορία και σημασιολογική, μέσω των ετικετών των κόμβων.

Το IT του στιγμιοτύπου μιας ερώτησης Q σε ένα δέντρο T ονομάζεται επίσης IT του Q στο T. Ένα IT είναι μια πλουσιότερη αναπαράσταση ενός αποτελέσματος μιας ερώτησης σε σχέση με τον LCA αφού δείχνει τόσο: α) τον τρόπο που τα στιγμιότυπα των λέξεων-κλειδιών συνδέονται στο υποδέντρο ενός LCA σχηματίζοντας ένα MCT, όσο και β) τον τρόπο που ο LCA συνδέεται με τη ρίζα του δέντρου δεδομένων.

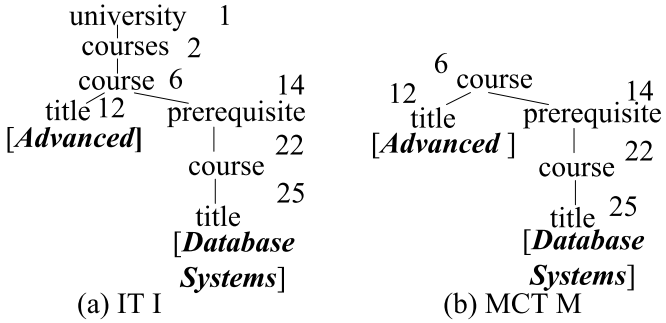
Ορισμός 4.3 (Απάντηση ερώτησης). *Η απάντηση μιας ερώτησης Q σε ένα δέντρο T είναι το σύνολο των IT του Q στο T.*

4.2 Συσταδοποίηση και κατηγοριοποίηση δενδρικών προτύπων

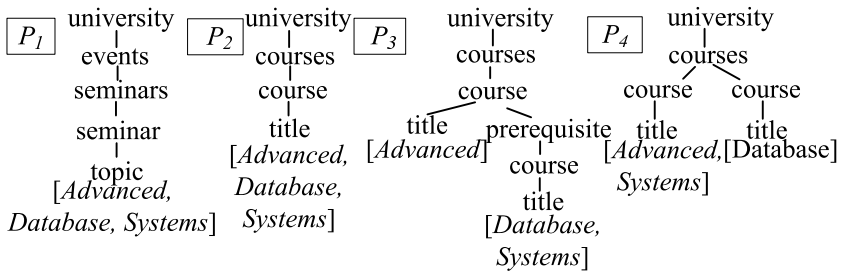
Στη συνέχεια παρουσιάζεται η συσταδοποίηση αποτελεσμάτων μιας ερώτησης με λέξεις-κλειδιά σε ένα δεντρο δεδομένων σε τρία επίπεδα. Οι συστάδες πρώτου επιπέδου ομαδοποιούν και διαχωρίζουν μεταξύ τους τα αποτελέσματα της ερώτησης. Οι συστάδες στα επόμενα επίπεδα συσταδοποιούν αυτές του πρώτου επιπέδου. Με αυτή την έννοια, ορίζεται μια ιεραρχική συσταδοποίηση, όπου η συσταδοποίηση εφαρμόζεται με διαφορετικούς βαθμούς λεπτομέρειας, οι οποίοι αυξάνονται στα χαμηλότερα επίπεδα συσταδοποίησης. Για κάθε συστάδα ορίζεται ένας *αντιπρόσωπος*. Ο χρήστης μπορεί να περιηγείται στην ιεραρχία επιλέγοντας συστάδες που τον ενδιαφέρουν από το τρίτο επίπεδο συσταδοποίησης και σιγά σιγά να φτάνει στο επίπεδο αυξημένης λεπτομέρειας (δηλ. το πρώτο) και τελικά στα ίδια τα αποτελέσματα. Επιπλέον, η επιλογή της κατάλληλης συστάδας κατά την περιήγηση διευκολύνεται από την ταξινόμηση των συστάδων ανά επίπεδο σύμφωνα με την οποία παρουσιάζονται οι συστάδες στο χρήστη σε κάθε βήμα της περιήγησής του.

4.2.1 Συσταδοποίηση πρώτου επιέδου

Στην πράξη, αποδεικνύεται πως διαφορετικά IT με όμοια δομικά και σημασιολογικά χαρακτηριστικά, μπορούν να αποτελούν στιγμιότυπα μιας ερώτησης, δηλ. να ακολουθούν το ίδιο δενδρικό πρότυπο. Τα μεμονωμένα IT δεν είναι πιθανώς ιδιαίτερα ενδιαφέροντα για το χρήστη. Αντίθετα, πολύ περισσότερη πληροφορία μπορούν οι χρήστες να λάβουν από τα διάφορα δενδρικά πρότυπα που τα IT ορίζουν. Ορίζουμε, λοιπόν, στη



Σχήμα 4.4: (a) Ένα IT και (b) το αντίστοιχο MCT.



Σχήμα 4.5: Μερικά πρότυπα για την ερώτηση $Q = \{Advanced\, Database\, Systems\}$ στο δέντρο της Εικόνας 4.3.

συνέχεια τα δεδνδρκά πρότυπα των IT τα οποία αποτελούν και τη βάση για τη μέθοδο ιεραρχικής συσταδοποίησης που εισάγεται ακολούθως.

Ορισμός 4.4 (Δενδρικό πρότυπο IT). *Ένα* πρότυπο P *μιας ερώτησης* Q *σε ένα δέντρο* T *είναι ένα δέντρο ισομορφικό (συμπεριλαμβανομένων των επισημειώσεων) προς ένα IT της ερώτησης* Q *στο* T*. Το* MCT *(δηλ. ελάχιστο συνδετικό δέντρο) ενός προτύπου* P *είναι το* P *χωρίς το μονοπάτι μεταξύ του* LCA *των επισημειωμένων κόμβων και της ρίζας του* P*.*

Ένα πρότυπο περιέχει όλη την πληροφορία ενός IT εκτός από την ακριβή του θέση στο δέντρο δεδομένων. Ως παράδειγμα, η Εικόνα 4.5 δείχνει τέσσερα πρότυπα (από τα 32 συνολικά) της ερώτησης $Q = \{Advanced, Database, Systems\}$ στο δέντρο T της Εικόνας **;;**. Το πρότυπο P_3 είναι το πρότυπο του IT της Εικόνας 4.4(a). Στο πρότυπο P_2 αντιστοιχούν δύο IT: αυτό του στιγμιότυπου της ερώτησης $\{(Advanced, 8), (Database, 8), (Systems, 8)\}$ και αυτό του στιγμιοτύπου $\{(Advanced, 10), (Database, 10), (Systems, 10)\}$.

Στο πρώτο επίπεδο συσταδοποίησης, μια συστάδα είναι των σύνολο των IT που αντιστοιχούν σε ένα κοινό πρότυπο, με το πότυπο αυτό να αποτελεί τον αντιρπόσωπό της. Για την ακρίβεια, σε όλα τα επίπεδα οι αντιπρόσωποι των συστάδων είναι πάντα πρότυπα. Ο *αντιπρόσωπος* μιας συστάδας $\mathcal{C}$ συμβολίζεται με $repr(\mathcal{C})$.

Ένα πρότυπο εκπροσωπεί μια πιθανή ερμηνεία μιας ερώτησης σε ένα δέντρο δεδομένων. Το πρώτο επίπεδο συσταδοποίησης περιλαμβάνει όλες τις πιθανές ερμνείες της ερώτησης στο δέντρο, αφού όλα τα IT μιας ερώτησης ομαδοποιούνται με βάση το πρότυπό τους. Το μόνο που λείπει είναι η φυσική θέση των εκάστοτε IT στο δέντρο,

4.2.2 Συσταδοποίηση δεύτερου επιέδου

Διαφορετικά πρότυπα μπορεί να είναι παρόμοια, αντιστοιχίζοντας τις λέξεις-κλειδιά με τον ίδιο τρόπο, δηλαδή να είναι ίδιο το μονοπάτι από τη ρίζα του δέντρου μέχρι τον επιδημειωμένο κόμβο ανά λέξη-κλειδί και ο LCA των επισημειωμένων κόμβων να είναι επίσης ο ίδιος. Τέτοιου είδους πρότυπα είναι σημασιολογικά κοντά, καθώς διαφέρουν μόνο στον τρόπο που συνδέονται αναμεταξύ τους τα στιγμιότυπα των λέξεων κλειδιών στο υποδέντρο του LCA, για να σχηματίσουν τους εκάστοτε μερικούς LCA. Τέτοια πρότυπα ομαδοποιούντα μαζί στο δεύτερο επίπεδο συσταδοποίησης, σχηματίζοντας *κλάσεις προτύπων*. Η Εικόνα 4.6 παρουσιάζει μία κλάση προτύπων,

Αυστηρά, η κλάση ορίζεται μέσω της σχέσης ισοδυναμίας $\approx$, που ορίζεται στη συυνέχεια..

Ορισμός 4.5 (Σχέση ισοδυναμίας $\approx$). *Έστω P και P' δύο πρότυπα μιας ερώτησης σε ένα δέντρο. $P \approx P'$ αν ισχύουν οι δύο επόμενες συνθήκες:*

α) το μονοπάτι από τη ρίζα ως τον LCA του P και P' είναι ίδιο, και

β) για κάθε λέξη-κλειδί της ερώτησης, το μονοπάτι από τον LCA στο στιγμιότυπο της λέξης-κλειδιού του P και του P' είναι ίδιο.

Η Εικόνα 4.7 δείχνει δύο πρότυπα, τα P_4 και P_5. Όπως υποδεικνύουν οι διακεκομένες γραμμές αντιστοίχισης, τα μονοπάτια αυτών των προτύπων ικανοποιούν και τις δύο συνθήκες του Ορισμού 4.5. Συνεπώς, $P_4 \approx P_5$.

Ορισμός 4.6 (Κλάση). *Δεδομένου ενός συνόλου προτύπων μιας ερώτησης Q σε ένα δέντρο T, μία κλαση του Q στο T είναι μια κλάση ισοδυναμίας ($\approx$) δενδρικών προτύπων.*

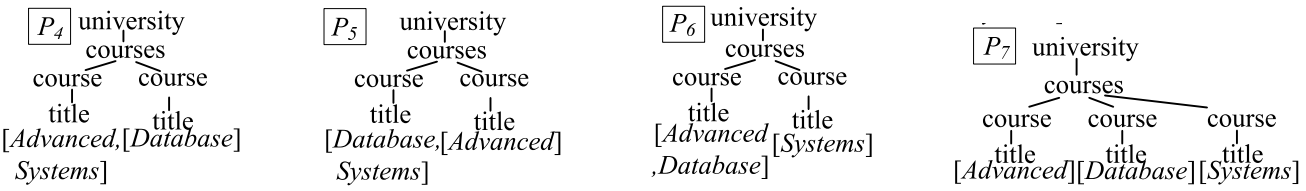
Η Εικόνα 4.6 απεικονίζει τέσσερα πρότυπα. Όπως είναι φανερό, αυτά τα πρότυπα ανήκουν στην ίδια κλάση. Αν εξεταστούν όλες οι ανά δύο συσχετίσεις των προτύπων αυτών, μπορεί κανείς να επιβεβαιώσει ότι ισχύουν πάντα οι συνθήκες του Ορισμού 4.5 και επομένως τα πρότυπα αυτά σχηματίζουν μια κλάση.

Ορίζεται ως το *μέγεθος* ενός προτύπου P ο αριθμός των ακμών στου P. Ένα από τα πρότυπα με το μικρότερο μέγεθος επιλέγεται τυχαία ως αντιπρόπσωπος μιας κλάσης. Για παράδειγμα, για την κλάση της Εικόνας 4.6 επιλέγεται τυχαία το πρότυπο P_4 να είναι αντιπρόσωπος μεταξύ των P_4, P_5 και P_6, που έχουν όλα μέγεθος 5. Το πρότυπο P_7 είναι μεγαλύτερο με μέγεθος 7.

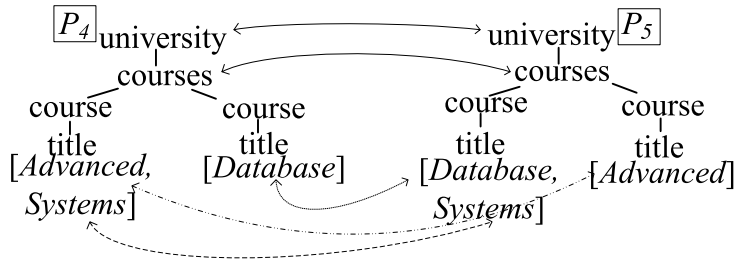
4.2.3 Συσταδοποίηση τρίτου επιπεδου

Η σχέση ισοδυναμίας $\approx$ εκφράζει ομοιότητες μεταξύ προτύπων μέσω εντοπισμού πανομοιότυπων μονοπατιών σάυτά. Ωστόσο, ομοιότητες μπορεί να είναι παρούσες και με μια λιγότερο αυστηρή λογική ομοιότητας, αν υπάρχει μια ένθεση των μονοπατιών ενός προτύπου στα μονοπάτια ενός άλλου. Η ένθεση ενός μονοπατιού p_1 σε ένα μονοπάτι p_2 ορίζεται όταν οι ακμές του p_1 μπορούν να αντιστοιχισθούν σε ακολουθίες ακμών του p_2. Με άλλα λόγια, μια σχέση πατέρα-παιδιού μεταξύ κόμβων του ενός μονοπατιού αντιστοιχίζεται σε μια σχέση προγόνου-απογόνου κόμβων του άλλου. Η σχέση αυτού του τύπου μεταξύ δύο προτύπων εκφράζεται μέσω του *ομομορφισμού μονοπατιών* και της σχέσης $\prec_{dph}$. Η σχέση $\prec_{dph}$ αξιοποιείται για τη συσταδοποίηση των κλάσεων προτύπων σε συλλογές, στο πλαίσιο του τρίτου επιπεδου συσταδοποίησης.

Ορισμός 4.7 (Ομομρφισμός μονοπατιών). *Έστω p_1 και p_2 δύο μονοπάτια δύο προτύπων, όπου οι τελευταίοι κόμβοι φέρουν την ίδια ετικέτα και την ίδια επισημείωση λέξκης-κλειδιού. Ορίζεται ένας* ομομορφισμός μονοπατιού *από το p_1 στο p_2 αν και μόνο αν υπάρχει συνάρτηση dph από τους κόμβους του p_1 στους κόμβους του p_2 τέτοια ώστε:*



Σχήμα 4.6: Μία κλάση τεσσάρων διαφορετικών προτύπων.



Σχήμα 4.7: Αντιστοιχίσεις μονοπατιών μεταξύ των προτύπων P_4 και P_5.

α) για κάθε κόμβο n του p_1, οι κόμβοι n και $dph(n)$ έχουν την ίδια ετικέτα.
β) αν ο n' είναι παιδί του n στο p_1, τότε ο $dph(n')$ είναι απόγονος του $dph(n)$ στο p_2.

Για παράδειγμα, στην Εικόνα 4.8, το μονοπάτι `courses/course/title[`$Database$`]` του προτύπου P_4 εμφανίζει έναν ομομορφισμό μονοπατιού προς το μονοπάτι του προτύπου P_8, `courses/course/prerequisite/course/title[`$Database$`]`. Όμοια, το μονοπάτι `courses/course/title[`$Advanced$`]` του P_4 έχει ομορφισμό προς το μονοπάτι `courses/course/title[`$Advanced$`]` του P_8, αφού είναι πανομοιότυπα.

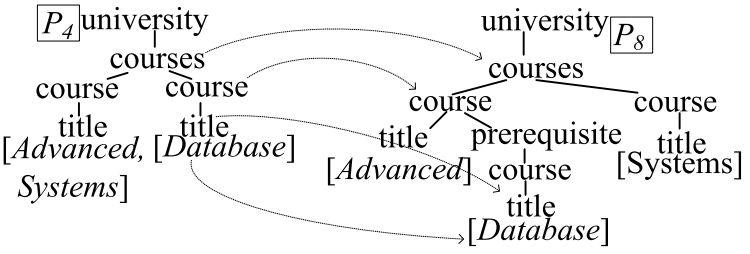
Η έννοια του ομομορφισμού μονοπατιών χρησιμοποιείται για να οριστεί η σχέση ομομορφισμού $\prec_{dph}$ μεταξύ προτύπων:

Ορισμός 4.8 (Σχέση $\prec_{dph}$). *Έστω P και P' δύο πρότυπα μιας ερώτησης Q σε ένα δέντρο δεδομένων. Ισχύει $P \prec_{dph} P'$ αν και μόνο αν*
α) τα P και P' έχουν το ίδιο μονοπάτι μεταξύ της ρίζας και του LCA τους.
β) για κάθε λέξη κλειδί k στο Q, το μονοπάτι από τον LCA μέχρι τον κόμβο που είναι επισημειωμένος με το k στο P εμφανίζει έναν ομομορφισμό μονοπατιού προς ένα μονοπάτι στο MCT του P'.

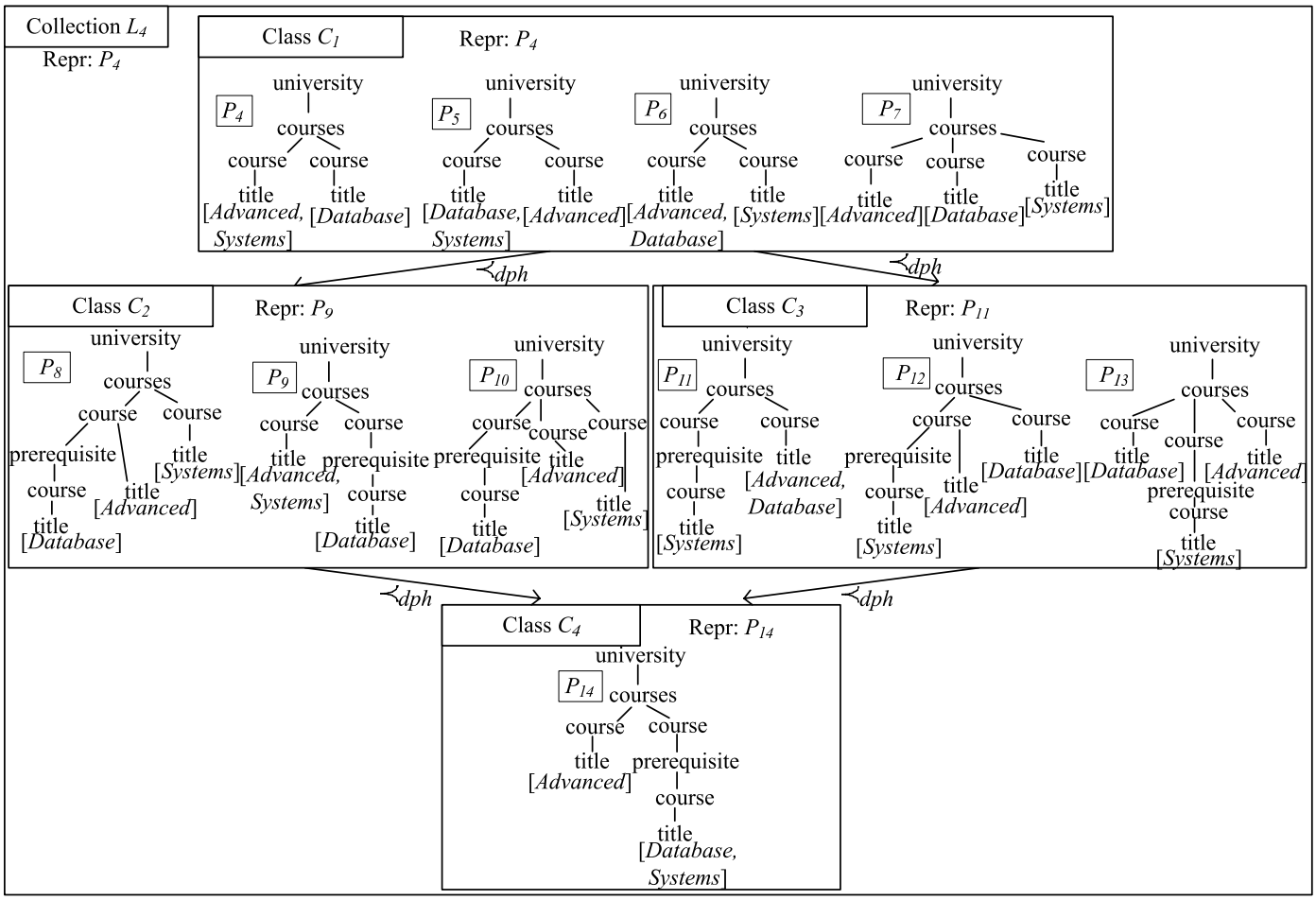
Ας πάρουμε τα πρότυπα P_4 και P_8 της Εικόνας 4.8. Όπως είναι φανερό,$P_4 \prec_{dph} P_8$ αφού για κάθε λέξη-κλειδί από τις $advanced$, $database$, $systems$, το μονοπάτι από τον κόμβο `courses` προς τον επισημειωμένο κόμβο από την αντιστοιχή λέξη-κλειδί εμφανίζει έναν ομομορφισμό μονοπατιού προς ένα μονοπάτι του MCT του P_8.

Η επόμενη πρόταση συσχετίζει τις σχέσεις $\prec_{dph}$ και $\approx$.

Πρόταση 4.1. *Έστω P και P' δύο πρότυπα μιας ερώτησης Qστο δέντρο T. $P \prec_{dph} P'$ και $P' \prec_{dph} P$ αν και μόνο αν $P \approx P'$.*



Σχήμα 4.8: Ομομορφισμός μονοπατιού από ένα μονοπάτι του P_4 προς ένα μονοπάτι του P_8



Σχήμα 4.9: Μια συλλογή κλάσεων για την ερώτηση $Q = \{Advanced\, Database\, Systems\}$.

Πράγματι, αν δύο πρότυπα P και P' ικανοποιούν τις προϋποθέσεις της Πρότασης 4.1, τα μονοπάτια από τη ρίζα στον LCA καθενός από τα P και P' είναι ίδια μεταξύ τους, και για κάθε λέξη-κλειδί τα μονοπάτια από του LCA στους επισημειωμένους κόμβους με τις αντίστοιχες λέξεις-κλειδιά είναι επίσης όμοια. Κατά συνέπεια, δύο πρότυπα που συνδέονται αμοιβαία μέσω της σχέσης $\prec_{dph}$ ανήκουν στην ίδια κλάση. Το αντίστροφο είναι προφανές: αν $P \approx P'$, τότε $P \prec_{dph} P'$ και $P' \prec_{dph} P$.

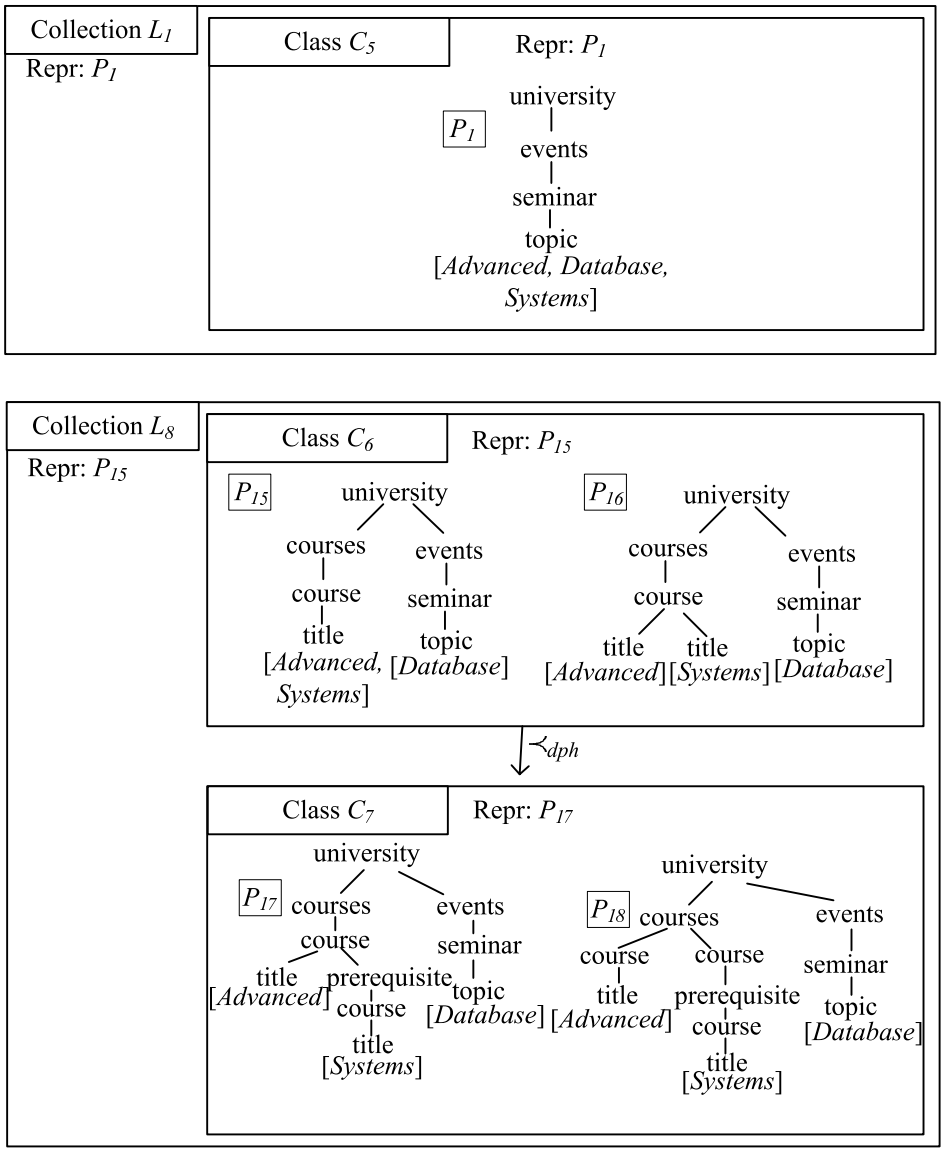
Πρόταση 4.2. *Έστω $\mathcal{R}$ το σύνολο των αντιπροσώπων των κλάσεων προτύπων μιας ερώτησης σε ένα δέντρο δεδομένων. Η σχέση $\prec_{dph}$ είναι ανακλαστική, αντισυμμετρική και μεταβατική, και ορίζει μια μερική διάταξη στο $\mathcal{R}$.*

Εφόσον η $\prec_{dph}$ είναι σχέση μερικής διάταξης στο $\mathcal{R}$, έχει τουλάχιστον ένα ελάχιστο στο $\mathcal{R}$. Η σχέση $\prec_{dph}$ χρησιμοποιείται για να εισαχθεί η έννοια της συλλογής.

Ορισμός 4.9 (Συλλογή). *Έστω το σύνολο κλάσεων μιας ερώτησης σε ένα δέντρο δεδομένων. Έστω, επίσης, ότι $\mathcal{R}$ είναι το σύνολο των αντιπροσώπων των κλάσεων αυτών. Μια συλλογή είναι το σύνολο L των κλάσεων, το οποίο περιέχει ακριβώς: α) μια κλάση C της οποίας ο αντιπρόσωπος είναι ένα ελάχιστο στοιχείο στο $\mathcal{R}$ σε σχέση με την $\prec_{dph}$, και β) όλες τις κλάσεις C' για τις οποίες ισχύει $repr(C) \prec_{dph} repr(C')$.*

Αυτό σημαίνει ότι για τα L και C, ισχύει $\forall C' \in L, C' \neq C$, $repr(C) \prec_{dph} repr(C')$ και για κάθε κλάση $C'' \notin L$, $repr(C) \not\prec_{dph} repr(C'')$ και $repr(C'') \not\prec_{dph} repr(C)$.

Προφανώς, ο αντιπρόσωπος $repr(C)$ είναι το ελάχιστο στοιχείο από το σύνολο των αντιπροσώπων του L σε σχέση με την $\prec_{dph}$. Ο αντιπρόσωπος μιας συλλογής L ορίζεται να είναι ο αντιπρόσωπος της κλάσης C, δηλαδή $repr(L) = repr(C)$.



Σχήμα 4.10: Δύο συλλογές κλάσεων, συμπληρωματικά στη συλλογή της Εικόνας 4.9, για την ερώτηση $Q = \{Advanced\, Database\, Systems\}$

Ο αριθμός των συλλογών για την ερώτηση Q στο T είναι ίσος με τον αριθμό των ελάχιστων στοιχείων στο σύνολο των αντιπροσώπων των κλάσεων $\mathcal{R}$ σε σχέση με την $\prec_{dph}$. Οι συλλογές είναι δυνατόν να έχουν επικαλύψεις. Ωστόσο, δεν είναι δυνατόν στα κοινά τους στοιχεία να περιλαμβάνεται η αντιπρόσωπευτική κλάση (δηλ. η κλάση της οποίας ο αντιπρόσωπος είναι ο αντιπρόσωπος της συλλογής). Κατά συνέπεια, καμία συλλογή δεν μπορεί να περικλείεται από μία άλλη.

Η Εικόνα 4.9 δείχνει μια συλλογή που περιέχει τέσσερις κλάσεις. Είναι εύκολο να δει κανείς ότι μια συλλογή ομαδοποιεί πρότυπα, τα οποία, παρόλο που δομικά παρουσιάζουν διαφορές, σημασιολογικά είναι παρόμοια, αφού όλα αναπαριστούν μαθήματα και

αντιστοιχίζουν τις λέξεις-κλειδιά της ερώτησης στους τίτλους των μαθημάτων αυτών. Οι ακμές μεταξύ των κλάσεων στην εικόνα αντιστοιχούν στις σχέσεις $\prec_{dph}$ μεταξύ των αντιπροσώπων των κλάσεων. Η ακμή από την κλάση C_1 στην C_4 δεν εμφανίζεται καθώς προκύπτει λόγω της μεταβατικής ιδιότητας της σχέσης $\prec_{dph}$ από τις άλλες ακμές. Το πρότυπο P_4 είναι ο αντιπρόσωπος της συλλογής, αφού είναι το ελάχιστο στοιχείο της μερικής διάταξης των αντιπροσώπων των τεσσάρων κλάσεων που ορίζει η σχέση $\prec_{dph}$. Δύο ακόμα συλλογές για την ίδια ερώτηση παρουσιάζονται στην Εικόνα 4.10.

4.2.4 Ταξινόμηση και περιήγηση στην ιεραρχία συστάδων δενδρικών προτύπων

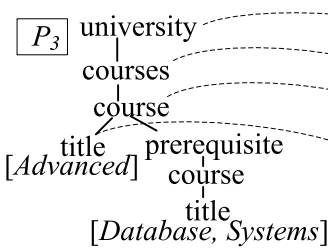
Ο στόχος της συσταδοποίησης των αποτελεσμάτων είναι η διευκόλυνση του χρήστη στον εντοπισμό των σχετικών αποτελεσμάτων μιας ερώτησης. Για το σκοπό αυτό, εκτός από τη συσταδοποίηση, ορίζεται επιπλέον κι ένα μοντέλο ταξινόμησης: οι συλλογές παρουσιάζονται ταξινομημένες στο πρώτο επίπεδο συσταδοποίησης, οι κλάσεις με τη σειρά τους εντός των συλλογών επίσης ταξινομημένες και τα πρότυπα εντός των κλάσεων το ίδιο. Ο σκοπός της ταξινόμησης είναι να παρουσιαστούν στο χρήστη πρώτα οι συστάδες που είναι πιθανότερο να περιέχουν τα πιο σχετικά αποτελέσματα, ώστε να ελαχιστοποιηθεί ο συνολικός αριθμός συστάδων που χρειάζεται να εξετασθούν από το χρήστη.

4.2.5 Ταξινόμηση συστάδων

Ταξινόμηση προτύπων. Εντός μιας κλάσης, τα πρότυπα ταξινομούνται με βάση το μέγεθός τους σε αύξουσα σειρά. Πρότυπα με ίδιο μέγεθος κατατάσσονται στην ίδια θέση. Σε αντιστοιχία με τη σημασιολογία μεγέθους, που εκφράζει την εγγύτητα των λέξεων-κλειδιών σε ένα στιγμιότυπο ερώτησης, αν το μέγεθος ενός προτύπου είναι μικρότερο από το μέγεθος ενός άλλου στην ίδια κλάση, τότε το πρώτο θεωρείται πιο σχετικό με την ερώτηση αφού συνδέει πιο στενά τα στιγμιότυπα των λέξεων-κλειδιών.

Ταξινόμηση κλάσεων. Η σχέση $\prec_{dph}$ χρησιμοποιείται για να ταξινομηθούν οι κλάσεις μέσα σε μια συλλογή. Σύμφωνα με την Πρόταση 4.2, η $\prec_{dph}$ είναι μια μερική διάταξη επί του συνόλου των προτύπων μιας ερώτησης. Βάσει του Ορισμού 4.9 ο αντιπρόσωπος μιας συλλογής είναι το ελάχιστο στοιχείο (ως προς την $\prec_{dph}$) στο σύνολο των αντιπροσώπων όλων των κλάσεων της συλλογής. Συνεπώς, κάθε συλλογή είναι ένα κατευθυνόμενος γράφος χωρίς κύκλους (DAG) με ρίζα, όπου οι κόμβοι αντιστοιχούν σε κλάσεις και οι ακμές σε σχέσεις $\prec_{dph}$ μεταξύ των αντιπροσώπων των κλάσεων. Μια ακμή από την κλάση C_1 στην κλάση C_2 υποδηλώνσει ότι $repr(C_1) \prec_{dph} repr(C_2)$. Ταξινομούμε τοπολογικά τις κλάσεις σε μια συλλογή, χρησιμοποιώντας επιπλέον τη μέγιστη απόσταση ενός κόμβου (που αναπαριστά μια κλάση) από τη ρίζα του γράφου, για να διευθετήσουμε την περίπτωση μη συγκρίσιμων κλάσεων. Η σχετική σειρά δύο κλάσεων με την ίδια μέγιστη απόσταση από τη ρίζα του γράφου είναι απροσδιόριστη, κατατάσσοντας τις δύο κλάσεις στην ίδια θέση της ταξινόμησης. Για παράδειγμα, στη συλλογή της Εικόνας 4.9, $max\_dist(C_2) = max\_dist(C_3) = 1$ και $max\_dist(C_4) = 2$. Συνεπώς, μια πιθανή ταξινόμηση των κλάσεων της συλλογής είναι: C_1, C_2, C_3, C_4.

Ταξινόμηση συλλογών. Το υψηλότερο επίπεδο της ιεραρχικής συσταδοποίησης σχηματίζεται από τις συλλογές. Για την ταξινόμηση των συλλογών, ορίζουμε μια νέα σχέση την $\prec_{opi}$ σε πρότυπα ερωτήσεων. Ο δείκτης opi αντιστοιχεί στα αρχικά των



Σχήμα 4.11: Ισομορφισμός μοναδικού μονοπατιού μεταξύ των προτύπων P_3 και P_4.

λέξεων "one-path isomorphism", δηλ. 'ισομορφισμός μοναδικού μονοπατιού¨, όπως δικαιολογείται στον ακόλουθο ορισμό.

Ορισμός 4.10 (Σχέση $\prec_{opi}$). *Έστω P και P' δύο πρότυπα μιας ερώτησης σε ένα δέντρο δεδομένων. Τότε, $P \prec_{opi} P'$ αν και μόνο αν υπάρχουν δύο μονοπάτια p και p' στο P και P', αντίστοιχα, από τη ρίζα σε δύο κόμβους των προτύπων με την ίδια επισημείωση λέξης-κλειδιού, τέτοια ώστε:*

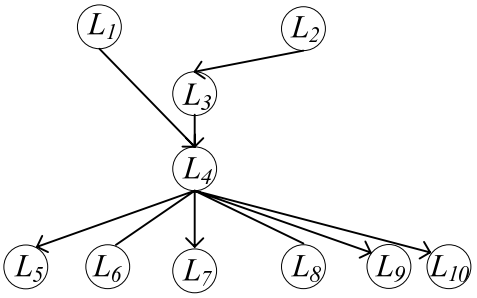
α) τα p και p' να είναι ισομορφικά και

β) ο LCA του P είναι απόγονος του LCA του P' στο p.

Η σχέση $\prec_{opi}$ συσχετίζει δύο πρότυπα, P και P' που μοιράζονται ένα κοινό μονοπάτι από τη ρίζα τους προς έναν επισημειωμένο με την ίδια λέξη-κλειδί κόμβο. Στον παραπάνω ορισμό, το P θεωρείται πιο σχετικό από το P', αφού το P υποδηλώνει μια πιο εξειδικευμένη σχέση των στιγμιοτύπων των λέξεων-κλειδιών σε σχέση με το P': η ρίζα του MCT του (δηλ. ο LCA των στιγμιοτύπων) εντοπίζεται σε μεγαλύτερο βάθος στο δέντρο δεδομένων. Στην Εικόνα 4.11, $P_3 \prec_{opi} P_4$.. Εξαιτίας της συνθήκης β) του Ορισμού 4.10, η σχέση $\prec_{opi}$ είναι *ακυκλική*.

Όπως συμβαίνει και με τις κλάσεις, για την ταξινόμηση των συλλογών δημιουργούμε ένα γράφο. Οι κόμβοι του γράφου αυτού είναι συλλογές. Προστίθεται μία ακμή στο γράφο μεταξύ των συλλογών L_1 και L_2 αν και μόνο αν $repr(L_1) \prec_{opi} repr(L_2)$. Αφού η σχέση $\prec_{opi}$ είναι ακυκλική, ο γράφος αυτός είναι επίσης ένας κατευθυνόμενος ακυκλικός γράφος. Η Εικόνα 4.12 δείχνει το των συλλογών του παραδείγματός μας. Οι αντιπρόσωποι των συλλογών L_1, L_2 και L_3 είναι τα πρότυπα P_1, P_2 και P_3, αντιστοίχως, τα οποία φαίνονται στην Εικόνα 4.5.

Χρησιμοποιείται και πάλι η τοπολογική σειρά των συλλογών στο γράφο για την ταξινόμησή τους, με τη μέγιστη απόσταση από τη ρίζα του γράφου να αποφασίζει τη σχετική σειρά μη συγκρίσιμων συλλογών. Επιπροσθέτως, συλλογές με ίση μέγιστη απόσταση από τη ρίζα του γράφου ταξινομούνται με βάση α) το μήκος του μονοπατιού από τη ρίζα ως τον LCA των αντιπροσώπων τους, που επίσης αντικατοπτρίζει το βάθος των LCA, και β) το μέγεθος των αντιπροσώπων τους. Πρότυπα με LCA σε μεγαλύτερο βάθος προτιμώνται καθώς, εν γένει, είναι πρότυπα με μικρότερο μέγεθος. Και στις δύο περιπτώσεις, θεωρούνται πιο σχετικά με την ερώτηση καθώς φέρνουν πιο κοντά τα στιγμιότυπα των λέξεων-κλειδιών. Συλλογές και για τις τρεις μετρικές (δηλ. τοπολογική απόσταση, βάθος LCA και μέγεθος) ταξινομούνται με τυχαία σειρά. Για παράδειγμα, μια πιθανή σειρά ταξινόμησης για τις συλλογές του παραδείγματός μας που απεικονίζονται στην Εικόνα 4.12 είναι L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}.



Σχήμα 4.12: Γράφος των συλλογών της ερώτησης

4.2.6 Περιήγηση στο ιεραρχικό σχήμα συστάδων

Η περιήγηση στην ιεραρχική οργάνωση των συστάδων ξεκινά στο τρίτο επίπεδο. Στο χρήστη παρουσιάζεται μια λίστα με συλλογές ταξινομημένη, όπως περιγράφηκε στην προηγούμενη ενότητα. Σ′ αυτό το σημείο προσφέρονται δυο επιλογές διάσχισης της ιεραρχίας από το χρήστη: είτε κατά βάθος, είτε κατά πλάτος. Κατά βάθος, επιλέγει μια συλλογή και προχωρά μέσα της για να εξετάσει τις κλάσεις που περιέχει. Οι κλάσεις παρουσιάζονται επίσης ταξινομημένες. Όταν επιλέξει μια κλάση, τα πρότυπα πια παρουσιάζονται ταξινομημένα. Τελικά, ο χρήστης επιλέγει ένα σχετικό πρότυπο για να του επιστραφούν τα εποτελέσματα που αντιστοιχούν σάυτό. Αν περισσότερα αποτελέσματα είναι επιθυμητά, προχωρά και με το επόμενο πρότυπο κ.ο.κ.. Αν εξετάσει και το τελευταίο πρότυπο της κλάσης, τότε γυρνά να εξετάσει την επόμενη κλάση αυτής που βρισκόταν, και αντίστοιχα από την τελευταία κλάση γυρνά στην επόμενη συλλογή αυτής που βρισκόταν.

Αν ο χρήστης περιηγηθεί στην ιεραρχία με κατά πλάτος διάσχιση, προχωρά επίπεδο, επίπεδο στην εξέταση των συστάδων. Έτσι, αρχικά ο χρήστης επιλέγει τις συλλογές που θεωρεί πιο σχετικές. Στη συνέχεια, οι κλάσεις των επιλεγμένων συλλογών παρουσιάζονται ταξινομημένες, ακολουθώντας τόσο τη σειρά ταξινόμησης των συλλογών που ανήκουν όσο και την εσωτερική ταξινόμηση ανά συλλογή. Αφού ο χρήστης επιλέξει τις πιο σχετικές γι′ αυτόν κλάσεις, εμφανίζονται στο χρήστη τα πρότυπα που ανήκουν στις κλάσεις αυτές ταξινομημένα με βάση τη σειρά κατάταξης των κλάσεών τους αλλά και τη δική τους σχετική θέση εντός των κλάσεων. Τελικά, με βάση τη σειρά ταξινόμησης των προτύπων παρουσιάζονται τα επιμέρους αποτελέσματα που αντιστοιχούν στα πρότυπα αυτά. Με την κατά πλάτος περιήγηση στην ιεραρχία, δε χρειάζεται η παρουσίαση να γυρίσει πίσω σε προηγούμενο επίπεδο συσταδοποίησης.

Αν όλα τα σχετικά αποτελέσματα για μια ερώτηση είναι επιθυμητό να επιστραφούν, τότε και οι δυο τεχνικές περιήγησης είναι ισότιμα κατάλληλες. Αντίθετα, αν τα κορυφαία k ($k \geq 1$) είναι επιθυμητά, τότε η κατά βάθος διάσχιση είναι πιο αποδοτική, καθώς ο χρήστης μπορεί να αποφύγει την εξέταση συστάδων (προτύπων, κλάσεων ή συλλογών) ταξινομημένων χαμηλότερα από την τρέχουσα συστάδα, όταν τα k αποτελέσματα έχουν επιστραφεί. Με την κατά πλάτος διάσιση, αυτο δεν είναι δυνατό, εκτός και περιοριστεί ο αριθμός των συστάδων, που εξετάζεται σε κάθε επίπεδο της ιεραρχίας, σε αριθμό ίσο με k.

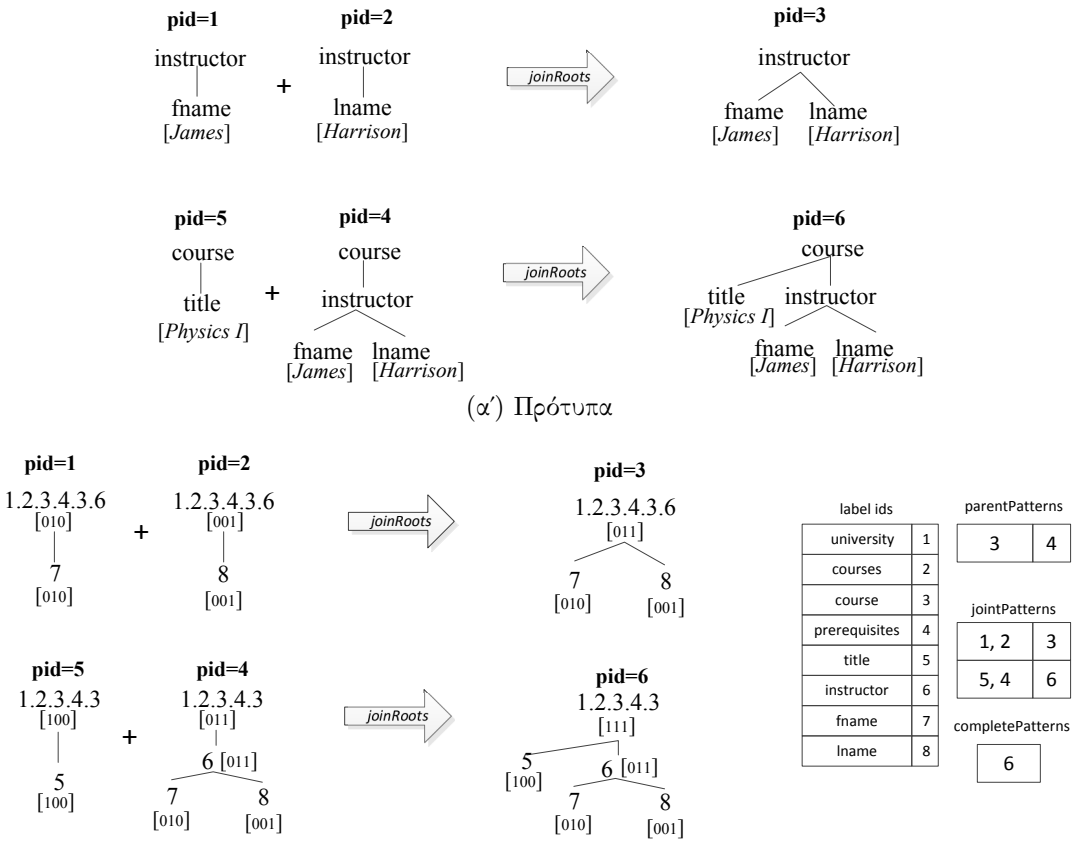
4.3 Αλγόριθμος εξαγωγής και ομαδοποίησης προτύπων

Η εύρεση όλων των δενδρικών προτύπων για μια ερώτηση σ΄ ένα δένδρο είναι μια διαδικασία αυξημένης πολυπλοκότητας. Για την υλοποίηση του συστήματος ιεραρχικής περιήγησης στις συστάδες των προτύπων μιας ερώτησης, σχεδιάστηκε ένας αποδοτικός αλγόριθμος. Ο αλγόριθμος ClusterStack υπολογίζει όλα τα δενδρικά πρότυπα και τα στιγμιότυπά τους στο δένδρο δεδομένων και χτίζει την ιεραρχία που που αναπτύχθηκε στην προηγούμενη ενότητα. Βασίζεται στη χρήση στοίβας και η αποδοτικότητα του συνίσταται στην αποφυγή υπολογισμού των ίδων προτύπων πολλές φορές. Δεδομένου του μικρού σχήματος ενός δένδρου δεδομένων σε σχέση με το συνολικό μέγεθος της βάσης δεδομένων αλλά και τον πεπερασμένο αριθμό τρόπων που ενδέχεται στιγμιότυπα λέξεων - κλειδιών να συνδέονται μεταξύ τους, εύκολα συνάγεται γιατί ο ClusterStack, υπολογίζοντας κάθε πρότυπο μια φορά μόνο, είναι τοσο αποδοτικός. Ο εντοπισμός των προτύπων και των αποτελεσμάτων ακολουθείται από την κατασκευή του γράφου της διάταξης των προτύπων και των συστάδων τους. Εξετάζεται η ύπαρξη ομομορφισμών μεταξύ δενδρικών προτύπων και κατασκευάζεται σταδιακά ο γράφος. Ο αλγόριθμος εκμεταλλευόμενος κάποιες ιδιότητες αποφεύγει τη δημιουργία ακμών μεταξύ δενδρικών προτύπων που συνδέονται μεταβατικά μέσω άλλων ακμών.

Ο αλγόριθμος *ClusterStack* είναι ένας αλγόριθμος που χρησιμοποιεί στοίβα. Δέχεται ως είσοδο μια ερώτηση με λέξεις-κλειδιά και τις ανεστραμμένες λίστες των λέξεων κλειδιών ενός δέντρου δεδομένων. Υπολογίζει τα αποτελέσματα της ερώτησης (δηλ. τα IT) και τα πρότυπά τους και ταυτόχρονα παράδει τις κλάσεις των προτύπων αυτών. Στη συνέχεια, κατασκευάζει και ταξινομεί τις συλλογές από κλάσεις. Οι κλάσεις που δημιουργούνται οργανώνονται σε μορφή γράφου με βάση τη σχέση $\prec_{dph}$. Μέσω του γράφου τους, οι κλάσεις ταξινομούνται εσωτερικά στη συλλογές που ανήκουν. Στο τέλος, δημιουργείται ο γράφος των συλογών με βάση τη σχέση $\prec_{opi}$ στο σύνολο των αντιπροσώπων τους, απ΄ όπου προκύπτει και η τρίτου επιπέδου ταξινόμηση. Το κύριο μέρος του αλγορίθμου και οι ενέργειες στη στοίβα κατά τη διάρκεια της πρώτης φάσης αποτίμησης της ερώτησης, φαίνονται στον ψευδωκώδικα του Αλγορίθμου 3.

Ο αλγόριθμος *ClusterStack* χρησιμοποιεί μια στοίβα. Τα στοιχεία της στοίβας αντιστοιχούν σε κόμβους του δέντρου δεδομένων και περιέχουν ένα σύνολο από μερικά και ολικά πρότυπα της υπεβεβλημένης ερώτησης. Τα ολικά πρότυπα περιλαμβάνουν επισημειώσεις για όλες τις λέξεις-κλειδιά της ερώτησης, αντίθετα από τα μερικά που αναφέρονται σε μέρος αυτών.Οι ανεστραμμένες λίστες περιλαμβάνουν για κάθε στιγμιότυπο λέξης-κλειδιού: α) τον κωδικό Dewey [10] του στιγμιοτύπου και β) και το μονοπάτι του στιγμιοτύπου μέχρι τη ρίζα του δέντρου, κωδικοποιημένο. Η κωδικοποίηση αυτή είναι μια κωδικοποίησης μορφής Dewey, η οποία αποτελείται από τα αναγνωριστικά των ετικετών του μονοπατιού, αντί των Dewey των προγόνων ενός κόμβου. Σε κάθε διακριτή ετικέτα του δέντρου αποδίδεται ένας μοναδικός ακέραιος. Η Εικόνα 4.13 παρουσιάζει μερικά στιγμιότυπα και πρότυπα (Εικόνα 4.13α΄), καθώς επίσης και τις κωδικοποιημένες μορφές τους «Εικόνα 4.13β΄», για την ερώτηση $Q = \{Physics\, James\, Harrison\}$. Ο πίνακας στην Εικόνα 4.13β΄ καταγράφει τα αναγνωριστικά των διαφορετικών ετικετών του δέντρου του παραδείγματός μας.

Τα αποτελέσματα μιας ερώτησης και τα πρότυπά τους υπολογίζονται στους επαναληπτικούς βρόζους των γραμμών 3-7 και 8-9. Κάθε στιγμιότυπο λέξης-κλειδιού από τις ανεστραμμένες λίστες εισάγεται στη στοίβα με τη σειρά εμφάνισης στο δέντρο. Αν το συγκεκριμένο μονοπάτι (δηλ. αλληλουχία ετικετών) επισημειωμένο με την αντίστοιχη



(α′) Πρότυπα

(β′) Κωδικοποιημένα πρότυπα

Σχήμα 4.13: Κωδικοποίηση στιγμιοτύπων και προτύπων από τον ClusterStack

λέξη-κλειδί δεν έχει εμφανιστεί ξανά, αποθηκεύεται στο πίνακα annotatedLabelPaths (γραμμή 4). Η εισαγωγή ενός στοιχείου στη στοίβα μπορεί να γίνει μόνο αν το κορυφαίο στοιχείο της στοίβας εκείνη τη στιγμή ατιστοιχεί στον πατέρα του κόμβου που ετοιμάζεται να εισαχθεί. Γι′ αυτό, όλοι οι κόμβοι που δεν είναι πρόγονοι αυτού του στιγμιοτύπου εξάγονται από τη στοίβα (γραμμές 5-6) και όλοι οι πρόγονοί του, που είναι απόγονοι του κόμβου που έμεινε στην κορυφή της στοίβας, εισάγονται στη στοίβα (διαδικασία *push*, γραμμές 13-14).

Η εισαγωγή ενός στοιχείου προκαλεί τη δημιουργία ενός μερικού προτύπου, δηλ. του επισημειωμένου μονοπατιού ετικετών (γραμμές 15-16). Αν ο κόμβος που αντιστοιχεί στο υπό επεξεργασία στιγμιότυπο υπάρχει ήδη στη στοίβα, τότε είναι πιθανό να περιλαμβάνει ήδη κάποια μερικά πρότυπα. Σ′ αυτήν την περίπτωση εξετάζεται η δυνατότητα συνδυασμού του νέου προτύπου με τα υπάρχοντα (γραμμή 17). Αν τελικά, παραχθούν νέα πρότυπα μ′ αυτόν τον τρόπο αποθηκεύεται τελικά στο στοιχείο της στοίβας η ένωση των συόλων των νέων με τα υπάρχοντα (γραμμή 18). Όταν οι ανεστραμμένες λίστες έχουν εξαντληθεί, η στοίβα αδειάζει με μια σειρά από ενέργειες εξαγωγής (δηλ. ενέργειες *pop*, γραμμές 8-9).

Τα αποτελέσματα της ερώτησης και τα πρότυπά τους κατασκευάζονται μέσα στη διαδικασία *pop* (γραμμές 19-34). Αν υπάρχουν ολικά πρότυπα στο κορυφαίο στοιχείο της στοίβας, τότε αυτά αφαιρούνται και ενημερώνεται το σύνολο των ολικών προτύπων και αποτελεσμάτων, όπου προστίθενται τα πρότυπα και οι συγκεκριμένοι LCA των αποτελεσμάτων, δηλ. των IT (γραμμές 23-25). Οι κλάσεις των προτύπων δημιουργούνται επίσης σ′ αυτό το στάδιο, σύμφωνα με τον Ορισμό 4.5: δημιουργείται μία κλάση για κάθε μονοπάτι ετικετών από τη ρίζα του δέντρου μέχρι κάποιον LCA και

Algorithm 3: ClusterStack

```
1  ClusterStack(k_1, ..., k_n: keyword query, invL: inverted lists)
2      s ← new Stack()
3      while n ← getNextNodeFromInvertedLists() do
4          annotatedLabelPaths.addIfNotExists(n.kw, n.labelPath)
5          while s.topNode is not ancestor of n do
6              pop(s)
7          push(s, n)
8      while s is not empty do
9          pop(s)
10     classCollections ← computeCollections(patternClasses)
11     collectionsGraph ← generateOPIgraph(classCollections)
12 push(Stack s, Node n)
13     while s.topNode is not parent or self of n do
14         push(s, ancestor or self of n at s.topNode.depth+1, "")
15     newP ← new Pattern(n.labelPath,flags.set(id(n.kw)))
16     newPid ← checkIfExistsOrAddToPatterns(newP)
17     newPatternIds ← composePatterns(s.top.patterns, newPatternId)
18     s.top.PatternIds ← union(s.top.PatternIds, newPatternIds)
19 pop(Stack s)
20     for curPid ← s.top.patterns.next() do
21         curP ← patterns.get(curPid)
22         if curP is complete then
23             s.top.removePatternId(curPid)
24             completePatterns.add(curPid)
25             curP.addLCA(s.top.dewey())
26             if curP.LCApath and curP.signature are not in patternClasses then
27                 newclass ← new PatternClass(curP)
28                 patternClasses.add(newClass)
29             newclass.add(curPid)
30         else
31             childPatterns.add(extendToParent(curPid))
32     s.pop()
33     newPatternIds ← composePatterns(s.top.patterns, childPatterns)
34     s.top.add(newPatternIds)
35 composePatterns(patternIdsA, patternIdsB)
36     foreach PatternIdsA as idA do
37         patA ← patterns[idA]
38         foreach currentPatternIdsB as idB do
39             patB ← patterns[idB]
40             if patA.kwFlags AND patB.kwFlags = 0 then
41                 if jointPatterns[min(idA,idB),max(idA,idB)] is set then
42                     idAB ← jointPatterns[min(idA,idB),max(idA,idB)]
43                 else
44                     patAB ← joinRoots(patA,patB)
45                     idAB ← checkIfExistsOrAddToPatterns(patAB)
46                     jointPatterns[min(idA,idB),max(idA,idB)] ← idAB
47                 newPatterns.add(idAB)
48     return newPatterns
49 joinRoots(patternA, patternB)
50     patAB ← patB.replaceRoot(patA.root)
51     patAB.kwFlags ← patA.kwFlags OR patB.kwFlags
52     patAB.size ← patA.size + patB.size
```

για κάθε σύνολο επισημειωμένων μονοπατιών ετικετών από τον LCA στα στιγμιότυπα

των λέξεων-κλειδιών. Η κωδικοποίηση, που εφαρμόζει ο ClusterStack, για τις ετικέτες (βλ. Εικόνα 4.13), χρησιμοποιούνται για την κατασκευή μιας υπογραφής ανά πρότυπα, που αποκαλύπτει όλα τα μονοπάτια του προτύπου από τη ρίζα ως τα φύλλα. Αν η κλάση ενός προτύπου δεν υπάρχει, δημιουργείται και το π•ροτυπο που προκάλεσε αυτή τη δημιουργία γίνεται το πρώτο μέλος της κλάσης αυτής (γραμμές 26-29). Τα μερικά πρότυπα του κορυφαίου στοιχείου της στοίβας, προωθούνται στο προηγούμενο στοιχείο, που αντιστοιχεί στον κόμβο πατέρα, με τις κατάλληλες προσαρμογές (γραμμή 31) και συνδυάζονται με υπάρχοντα πρότυπα εκεί για να παραχθούν νέα (γραμμές 33-34).

Η σύνθεση των προτύπων (γραμμές 35-48) συνίσταται στην ταύτιση των ριζών τους (γραμμή 50). Το νέο πρότυπο αποκτά τις επισημειώσεις των αρχικών προτύπων και το άθροισμα των μεγεθών τους (γραμμές 51-52). Η παραγωγή νέων προτύπων από τον αλγόριθμο συμβαίνει είτε μέσα στη διαδικασία *composePatterns* (γραμμές 17,33), είτε στην *extendToParent* (γραμμή 31). Με στόχο τον επαναλαμβανόμενο συνδυασμό των ίδων προτύπων, που καταλήγει στη δημιουργία επίσης ίδιων προτύπων που έχουν προκύψει ξανά, γίνεται έλεγχος δύο μεταβλητών, των *jointPatterns* και *parentPatterns* αντίστοιχα, όπου έχουν αποθηκευτεί οι ενέργειες κατασκευής προτύπων στο παρελθόν. Η διαδικασία κατασκευής και οι μεταβλητές αυτές φαίνονται στην Εικόνα 4.13.

Μετά την κατασκευή των προτύπων, η επεξεργασία προχωρά με τη διαδικασία *computeCollections* (γραμμή 10), η οποία δημιουργεί και ταξινομεί συλλογές από τις δημιουργημένες κλάσεις, μεβάση τη σχέση $\prec_{dph}$. Η διαδικασία *computeCollections* παρουσιάζεται αναλυτικά στον ψευδοκώδικα του Αλγορίθμου 4. Αρχικά, τα επισημειωμένα μονοπάτια από τη ρίζα ως τα φύλλα των αντιπροσώπων των κλάσεων συγκρίνονται ανά δύο μεταξύ τους για ανακάλυψη ομομορφισμών μονοπατιού. Το αποτέλεσμα των συγκρίσεων αποθηκεύεται στη μεταβλητή pathHomomorphisms (γραμμή 3) και αξιοποιείται για να αποκαλυφθούν οι σχέσεις $\prec_{dph}$ μεταξύ των κλάσεων. Ο *ClusterStack* αποφεύγει τον εξαντλητικό έλεγχο όλων των ζευγαριών κλάσεων κατά τη διαδικασία αυτή. Αντίθετα, μειώνει τις συγκρίσεις αξιοποιώντας την ακόλουθη παρατήρηση.

Για ένα πρότυπο P σε μια κλάση C, έστω l το μονοπάτι του P από τη ρίζα στον κόμβο LCA και $p_i,\ i = 1, \ldots, n$, το μονοπάτι από τον LCA ως το φύλλο του προτύπου, για κάθε μία από τις n λέξεις-κλειδιά που επισημειώνουν τα μονοπάτια του προτύπου. Τότε, το *μέγιστο μέγεθος προτύπου maximum pattern size, mps* για την κλάση C είναι

$$mps(C) = length(l) + \sum_{i=1}^{n} length(p_i)$$

Το μέγεθος $mps(C)$ είναι το μέγιστο μέγεθος που μπορεί να έχει ένα πρότυπο της C και αντιστοιχεί στο πρότυπο, του οποίου τα μονοπάτια από τον LCA ως τον κάθε επισημειωμένο κόμβο δε μοιράζονται καμία ακμή. Το μέγεθος αυτό είανι θεωρητικό, καθώς το πρότυπο αυτό μπορεί να μην εμφανίζεται στην κλάση C.

Παρατήρηση 4.1. *Έστω $repr(C_1)$ και $repr(C_2)$ οι αντιπρόσωποι δύο διαφορετικών κλάσεων C_1 και C_2 της ίδιας συλλογής. Αν $repr(C_1) \prec_{dph} repr(C_2)$, τότε $mps(C_1) < mps(C_2)$.*

Η διαδικασία *computeCollections* διχωρίζει τις κλάσεις με βάση τα μονοπάτια τους από τη ρίζα ως τον LCA (γραμμές 4-5). Επιπλέον, ομαδοποιεί τις κάσεις με τα ίδια μονοπάτια, ανάλογα με το *mps* μέγεθός τους (γραμμές 6-7). Εκμεταλλευόμενη την Παρατήρηση 4.1, η διαδικασία *computeCollections* εξετάζει για ύπαρξει σχέσεων $\prec_{dph}$ μόνο μεταξύ κλάσεων με το ίδιο μονοπάτι από τη ρίζα στον LCA, αλλά με διαφορετικό μέγιστο μέγεθος προτύπου. Οι κλάσεις ίδιου μονοπατιού εξετάζονται σε αύξουσα

Algorithm 4: computeCollections procedure

```
1  computeCollections(patternClasses, annotatedLabelPaths)
2      collections ← ∅
3      descPathHomomorphisms ← generateDPH(annotatedLabelPaths)
4      for each lca in classes do
5          lcaClasses ← getLcaClasses(classes, lca)
6          for each maxPatternSize observed in lcaClasses, in ascending order do
7              sizeClasses ← getSizeClasses(lcaClasses, maxPatternSize)
8              if maxPatternSize is the minimum in the sizeClasses then
9                  for each cl in sizeClasses do
10                     col ← new Collection(cl)
11                     collections.add(col)
12             else
13                 connected ← false
14                 for each childClass in sizeClasses do
15                     for each mps < mps(childClass) in descending size order do
16                         smallerSizeClasses ← getSizeClasses(lcaClasses, mps)
17                         for each parentClass in smallerSizeClasses do
18                             if parentClass.hasDPHRto(childClass) then
19                                 parentClass.connectTo(childClass)
20                                 childClass.rank ← parentClass.rank+1
21                                 connected ← true
22                     if not connected then
23                         col ← new Collection(childClass)
24                         collections.add(col)
25     return collections
```

σειρά *mps* (γραμμές 6-7). Για κάθε κλάση με ελάχιστο *mps*, δημιουργείται μία συλλογή (γραμμές 8-11), καθώς σύμφωνα με την Παρατήρηση 4.1 οι κλάσεις αυτές είναι ελάχιστα στοιχεία ως προς τη σχέση $\prec_{dph}$. Για την ανακάλυψη σχέσεων $\prec_{dph}$ μεταξύ κλάσεων, μια κλάση συγκρίνεται μόνο με κλάσεις μικρότερου *mps*, σε φθίνουσα σειρά *mps* (γραμμές 15-21). Αν βρεθεί μια τέτοια σχέση, οι δύο κλάσεις συνδέονται με μια ακμή (γραμμή 19), και η κλάση με μεγαλύτερο *mps* συνδέεται με τις συλλογές της κλάσης με το μικρότερο. Η θέση στην ταξινόμηση της κλάσης που συνδέθηκε μόλις με το γράφο, τίθεται στην τιμή που αντιστοιχεί στη θέση της κλάσης πατέρα, αυξημένης κατά 1 (γραμμή 20). Αν μια κλάση δεν μπορεί να συνδεθεί με καμιά κλάση μικρότερου *mps*, μια νέα συλλογή γι′ αυτήν την κλάση δημιουργείται (γραμμές 22-24).

Τελείως ανάλογα, η διαδικασία *generateOPIgraph* κατασκευάζει το γράφο των συλλογών με βάση τη σχέση $\prec_{opi}$, και τη χρησιμοποιεί γιαν καταλήξει στην ταξινόμηση των συλλογών. Προχωρά εξετάζοντας συλλογές σε αύξουσα σειρά βάθους που βρίσκεται ο LCA. Εξετάζοντας, στη συνέχεια, σχέσεις προγόνου-απογόνου μεταξύ των LCA ενός ζευγαριού προτύπων, περιορίζει ακόμα περισσότερο τις συγκρίσεις, που απαιτούνται για την ανακάλυψη των $\prec_{opi}$ σχέσεων μεταξύ συλλογών.

Ανάλυση πολυπλοκότητας. Η πολυπλοκότητα σε χρόνο του αλγορίθμου *ClusterStack* προκύπτει από το κόστος σε χρόνο της φάσης κατασκευής των προτύπων και κλάσεων αλλά και της φάσης δημιουργίας των γράφων κλάσεων και συλλογών. Στην πρώτη φάση, οι ενέργειες εισαγωγής και εξαγωγής στη στοίβα κατά την επεξεργασία των ανεστραμμένων λιστών καθορίζουν την πολυπλοκότητα χρόνου. Έστω k ο αριθμός των λέξεων-κλειδιών μιας ερώτησης και L_i η ανετραμμένη λίστα της λέξης-κλειδιού i. Ο συνολικός αριθμός στιγμιοτύπων λέξεων-κλειδιών είναι $|L| = \sum_i |L_i|, i \in [1, k]$.

Κάθε εισαγωγή ενός στιγμιοτύπου στη στοίβα θα χρειαστεί κατά μέγιστο h εξαγωγές και h εισαγωγές στοιχείων στη στοίβα, με h να υποδηλώνεται το ύψος του δέντρου. Αν υπάρχουν p μερικά πρότυπα στο κορυφαίο στοιχείο της στοίβας, το πρότυπο ενός μονοπατιού του νέου στιγμιοτύπου μπορεί να συνδυαστεί το πολύ με p πρότυπα σε χρόνο $O(p)$. Όταν ένα στοιχείο εξάγεται από τη στοίβα, όλα τα πρότυπα που περιέχει επεκτείνονται με μία ακμή μέχρι τον κόμβο πατέρα, του οποίου το στοιχείο θα καταστεί το νέο κορυφαίο στοιχείο της στοίβας. Θεωρώντας ότι κανένα πρότυπο δεν έχει συμπληρωθεί (δηλ. δεν είναι ολικό πρότυπο), η ενέργεια αυτή είναι κόστους $O(p)$. Τα μερικά πρότυπα συνδυάζονται επίσης, με τα μερικά πρότυπα του πατέρα κόμβου για παραγωγή νέων, σε χρόνο $O(p^2)$. Έτσι, συνολικά, η πολυπλοκότητα της διαδικασίας της πρώτησς φάσης είναι $O(|L|hp^2)$.

Κατά τη δεύτερη φάση του αλγορίθμου, κάθε κλάση (συλλογή) υπό εξέταση συνδέεται με άλλες κλάσεις (συλλογές). Ο αριθμός των κλάσεων (συλλογών) καθορίζεται από τον αριθμό των διακριτών μονοπατιών ετικετών των στιγμιοτύπων των λέξεων-κλειδιών και από τι ύψος του δέντρου. Αν l είναι ο μέγιστος αριθμός διακριτών μονοπατιών ετικετών ανά λέξη-κλειδί, ο μέγιστος αριθμός κλάσεων που μπορεί να δημιουργηθούν είναι hl^k, δηλ. ο αριθμός όλων των δυνατών συνδυασμών όλων των μονοπατιών ετικετών, με όλους τους δυνατούς LCA σε όλα τα βάθη. Ο μέγιστος αριθμός διακριτών LCA είναι hl, το οποίο σημαίνει ότι η διαδικασία *computeCollections* εξετάζει ομάδες από το πολύ l^{k-1} κλάσεις. Ο αριθμός των συγκρίσεων μεταξύ των κλάσεων σε κάθε ομάδα μεγιστοποιείται όταν όλα τα δυνατά *mps* εμφανίζονται στην ομάδα. Σ΄ αυτήν την περίπτωση, διενεργούνται $l^{k-1}k^2l^2$ συγκρίσεις ανά ομάδα. Συνεπώς, η πολυπλοκότητα της *computeCollections* είναι $O(hk^2l^k)$. Η διαδικασία *generateOPIgraph* επιδεικνύει ανάλογη πολυπλοκότητα. Ο χρόνος αυτός απαιτείται στη χειρότερη περίπτωση, κατά την οποία μία διακριτή συλλογή δημιουργείται για κάθε κλάση. Εφόσον, ο χρόνος κυριαρχείται από την 1η φάση του αλγορίθμου, η πολυπλοκότητα του *ClusterStack* είναι $O(|L|hp^2)$.

4.4 Αξιολόγηση αποτελεσματικότητας και επίδοσης της μεθόδου

Η ποιότητα της ανάκτησης αποτελεσμάτων μέσω της μεθοδολογίας συσταδοποίησης που προτάθηκε και η αποδοτικότητα του αλγορίθμου μελετήθηκαν πειραματικά. Η αποτελεσματικότητα της συσταδοποίησης και ταξινόμησης μέσω της ιεραρχίας τριών επιπέδων επίσης συγκρίθηκε με το σύστημα XMean, [34], που είναι το πιο πρόσφατο σύστημα που έχει προταθεί στη βιβλιογραφία και προτείνει μια συγκρίσιμη μέθοδο κατάταξης αποτελεσμάτων μέσω ιεραρχίας.

Για τη διεξαγωγή της πειραματικής ανάλυσης, χρησιμοποιήθηκαν τα σύνολα δεδομένων DBLP[1], Mondial[2], SIGMOD[2] και NASA datasets[2]. Τα στατιστικά χαρακτηριστικά τους καταγράφονται στον Πίνακα 4.1. Όπως συμβαίνει και στη σχετική βιβλιογραφία [5, 34], τα σύνολα δεδομένων Mondial και SIGMOD χρησιμοποιήθηκαν στα πειράματα ανάλυσης της αποτελεσματικότητας της μεθόδου. Τα σύνολα δεδομένων NASA και DBLP αξιοποιήθηκαν για τη μέλετη της επίδοσης της προσέγγισης, καθώς αυτά τα σύνολα δεδομένων είναι μεγαλύτερου μεγέθους. Μάλιστα, για να αυξηθεί το μέγεθος του NASA, έγινε πολλαπλασιασμός του δέντρου δεδομένων του τέσσερις

[1]http://dblp.uni-trier.de/xml/

[2]http://www.cs.washington.edu/research/xmldatasets/

	Mondial	SIGMOD	DBLP	NASA
Size	1 MB	467 KB	1.15 GB	956 MB
# nodes	69,846	15,263	34,141,216	21,318,481
# distinct tags	50	12	44	70
# distinct label paths	119	12	196	111
Average depth	3.00	4.60	1.93	4.56
Maximum depth	5	6	5	7

Πίνακας 4.1: Στατιστικά των συνόλων δεδομένων Mondial, SIGMOD, DBLP και NASA.

φορές κάτω από τη ρίζα. Τα πειράματα διενεργήθηκαν σε ένα σύστημα 2.9 GHz Intel Core i7 με μνήμη 3 GB και λειτουργικό σύστημα Ubuntu.

Στη συνέχεια, παρουσιάζονται οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση της μεθόδου συσταδοποίησης και ταξινόμησης των αποτελεσμάτων ερωτήσεων λέξεων-κλειδιών σε δενδρικά δεδομένα. Ακολουθεί η παρουσίαση των αποτελεσμάτων αξιολόγησης της αποτελεσματικότητας της μεθόδου και τέλος η μελέτη επίδοσης.

4.4.1 Μετρικές αξιολόγησης ιεραρχίας συστάδων

Όπως αναφέρεται και στην εργασία [34], τα κλασικά μεγέθη του πεδίου IR, όπως η ακρίβεια (precision) και η πληρότητα (recall) δεν είναι κατάλληλα για την αξιολόγηση μιας μεθόδου ταξινόμησης αποτελεσμάτων σε ιεραρχία, όπου απαιτείται αλληλεπίδραση με το χρήστη ώστε να βρεθούν τα αποτελέσματα μιας ερώτησης βήμα, βήμα. Για το λόγο αυτό, προσαρμόσαμε το *χρόνο εντοπισμού* (reach time), που έχει εισαχθεί στην εργασία [24]. Ο χρόνος εντοπισμού εκφράζει το χρόνο που χρειάζεται ο χρήστης να δαπανήσει στο σύστημα, ώστε να εντοπίσει τα σωστά αποτελέσματα στην ερώτησή του. Θεωρούμε ότι όλα τα αποτελέσματα (IT) που αντιστοιχούν σε ένα κοινό δενδρικό πρότυπο είναι ισόποσα ενδιαφέρονται για το χρήστη. Αν ένα τέτοιο πρότυπο επιλεχθεί από το χρήστη στην ιεραρχία του συστήματος, όλα τα επιμέρους IT που του αντιστοιχούν θα επιστραφούν ως αποτελέσματα. Η τιμή του χρόνου εντοπισμού, ως μετρική αξιολόγησης, ποικίλλει ανάλογα με τη διεπαφή χρήστη του συστήματος και τη διαδικασία ανάκτησης, που είναι υπό εξέταση. Όπως συζητήθηκε και στην Ενότητα ;;, ο χρήστης περιηγείται στην ιεραρχία του συστήματός μας ξεκινώντας από το τρίτο επίπεδο, δηλ. τις συλλογές. Θεωρούμε ότι η διάσχιση της ιεραρχίας γίνεται κατά βάθος. Για να αξιολογηθεί η επίδραση της ταξινόμησης στο χρόνο εντοπισμού, εξετάζουμε δύο παραλλαγές της διεπαφής χρήστη: α) με τις συστάδες να παρουσιάζονται στο χρήστη ταξινομημένες με βάση τα κριτήρια που συζητήθηκαν στην Ενότητα 4.2.5 και β) με τις συστάδες τυχαία ταξινομημένες.

Ορίζουμε το χρόνο εντοπισμού αυστηρά, στη συνέχεια:Ωε φορμαλλψ δεφινε τηε ρεαςη τιμε ας σηοων βελοω:

$$t_{reach} = (\sum_{i=1}^{h} n_i) \tag{4.1}$$

όπου n_i είναι ο αριθμός των συστάδων που εξετάζονται από το χρήστη στο επίπεδο i, και h είναι ο αριθμός από επίπεδα της ιεραρχίας. Στην περίπτωση του συστήματός μας, το βάθος της ιεραρχίας, h, είναι πάντα ίσο με 3. Η εξέταση μιας συστάδας από το χρήστη υπονοεί την εξέταση του αντιπροσώπου της συστάδας, για εξαγωγή συμπεράσματος αν η συγκεκριμένη συστάδα περιέχει ενδιαφέροντα αποτελέσματα.

Η τιμή της παραμέτρου n_i εξαρτάται από τη διαδικασία ανάκτησης, καθώς επίσης κι από την ταξινομημένη ή όχι παρουσίαση των συστάδων σε κάθε βήμα. πολογίζουμε την τιμή του n_i για τα διαφορετικά σενάρια ως εξής:

Ανάκτηση όλων των σχετικών προτύπων και αποτελεσμάτων Ο χρήστης χρειάζεται να εξετάσει όλες τις συστάδες του κορυφαίου επιπέδου (δηλ. τις συλλογές), και στη συνέχεια να εξετάσει αναδρομικά όλα τα παιδιά των συστάδων που θεωρεί σχετικά με την ερώτηση. Σ′ αυτήν την περίπτωση, η ταξινόμηση δεν παίζει ρόλο, αφού ο χρήστης πρέπει να εξετάσει όλες τις συστάδες σε κάθε επίπεδο, οπότε ο χρόνος εντοπισμού, t_{reach}, δεν επηρεάζεται.

Ανάκτηση το πολύ k σχετικών προτύπων. Εφόσον οι χρήστες ακολουθούν μια κατά βάθος διάσχιση της ιεραρχίας, μπορούν να σταματήσουν την αναζήτηση επιπλέον συστάδων στο σημείο που έχουν εντοπίσει k σχετικά πρότυπα. Δεδομένου ότι ο χρόνος εντοπισμού εξαρτάται από τη σειρά εμφάνισης των συστάδων στο χρήστη, υπολογίζουμε την *ελάχιστη*, *μέγιστη* και *αναμενόμενη* τιμή για την παράμετρο n_i (βλ. Εξίσωση 4.1), λαμβάνοντας υπόψη όλες τις δυνατές ταξινομήσεις των συστάδων παιδιά μιας συστάδας πατέρα στην ιεραρχία. Αν οι συστάδες είναι ταξινομημένες, παράγουμε μόνο τις διαφορετικές ταξινομήσεις που προκύπτουν από τη αναταξινόμηση των συστάδων που έχουν ταξινομηθεί στην ίδια θέση,

Για τον καθορισμό της βάσης αληθείας (ground truth) για τα πειράματα αποτελεσματικότητας της μεθόδου, επιστρατεύτηκαν πέντε χρήστες, μη σχετική με την εργασία μας, οι οποίοι χαρακτήρισαν τα πρότυπα των ερωτήσεων ως σχετικά ή όχι. Η σχετικότητα ανά πρότυπο, καθορίστηκε τελικά από την πλειοψηφία των χαρακτηρισμών εκ μέρους των χρηστών.

4.4.2 Αποτελεσματικότητα μεθόδου RTCluster

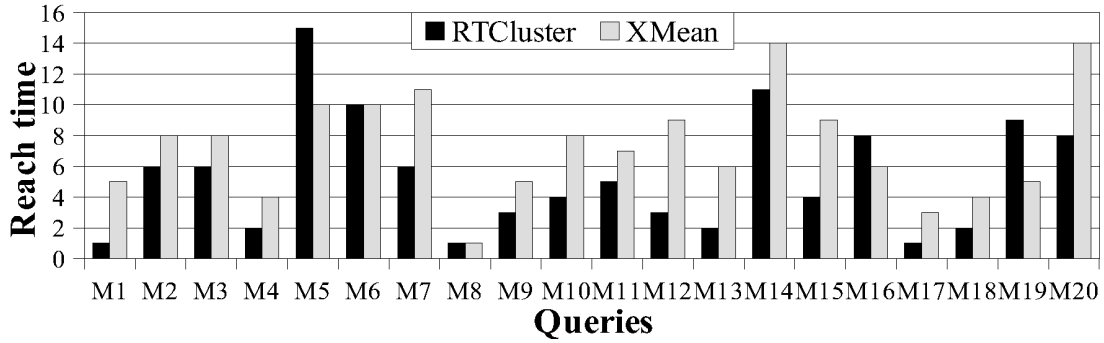
Για τα πεοράματα αποτελεσματικότητας, συγκρίθηκε το σύστημά μας (αναφερόμενο εφεξής ως *Result Tree Cluster—RTCluster*) με το σύστημα *XMean* [34], όπου πρόσφατα έχει προταθεί μια μεθοδολογία ιεραρχικής συσταδοποίησης αποτελεσμάτων σε XML δεδομένα. Το σύστημα *XMean*. Το σύστημα *XMean* εξάγει επίσης πρότυπα από τις απαντήσεις και κατασκευάζει μια ιεραρχία συστάδων, χαλαρώνοντας δομικά τα πρότυπα αυτά. Η χαλάρωση αυτή πραγματοποιείται με διαδοχική αφαίρεση κόμβων και ακμών από τα φύλλα προς τη ρίζα των προτύπων, με όλους τους δυνατούς τρόπους. Ο γράφος που κατασκευάζεται τελικά καλείται γράφος χαλάρωσης. Στη συνέχεια, μειώνεται ακόμα περισσότερο το μέγεθός του, μετατρέποντάς τον τελικά σε μια ιεραρχική δενδρική δομή. Η προσέγγιση αυτή εκμεταλλεύεται πληροφορία σχήματος XML, δηλ. το DTD του συνόλου δεδομένων, ώστε να διαχωρίσει τους κόμβους που μπορούν να επιστραφούν ως αποτελέσματα, μια και αναπαριστούν οντότητες και όχι πληροφοριακά στοιχεία. Για λόγους ευθείας σύγκρισης των δύο προσεγγίσεων, κατά την υλοποίηση του συστήματος *XMean*, αφαιρέσαμε το χαρακτηρισμό των κόμβων, ώστε οι δύο μέθοδοι να συσταδοποιούν τον ίδιο αριθμό αποτελεσμάτων. Όπως έχει σημειωθεί στο παρελθόν [49], αποτελέσματα των οποίων ο LCA ταυτίζεται με τη ρίζα του δέντρου δεδομένων δεν έχουν νόημα, καθώς επί της ουσίας δε συσχετίζουν τα στιγμιότυπα των λέξεων-κλειδιών, παρά μόνο στο πλαίσιο ολόκληρου του συνόλου δεδομένων.

Ανάκτηση όλων των αποτελεσμάτων. Εκτελέστηκαν 20 ερωτήσεις (βλ. Πίνακα 4.2) στα σύνολα δεδομένων Mondial και SIGMOD. Όλες οι ερωτήσεις επιλέχθηκαν από παλαιότερες δημοσιεύσεις εργασιών, διαφορετικών ερευνητών: M1-M7 και S1-S8 από την [38], M8-M12 και S9-S11 από την [34], M13 από την [35], M14-M16 από την

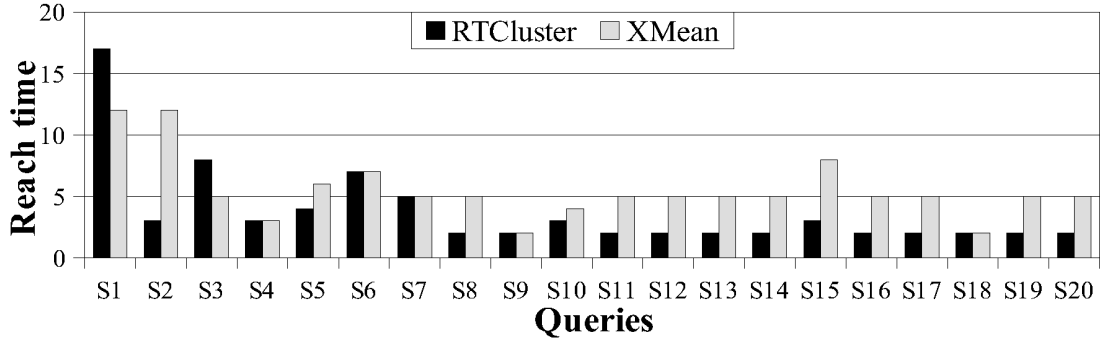
Dataset	ID	Query
Mondial	M1	*torneaelv country province*
	M2	*roman catholic percentage united states*
	M3	*population 87 albania city*
	M4	*organization name members*
	M5	*country government republic*
	M6	*country ethnicgroups german*
	M7	*city washington province*
	M8	*france territory*
	M9	*lake located*
	M10	*singapore country*
	M11	*religions christian muslim*
	M12	*province houston dallas*
	M13	*belarus population*
	M14	*united states birmingham population*
	M15	*ethnicgroups chinese indian capital*
	M16	*country muslim*
	M17	*international monetary fund established*
	M18	*government democracy muslim*
	M19	*jewish percentage*
	M20	*japan tokyo population*
SIGMOD	S1	*author position 01 harry article*
	S2	*jim gray title initpage endpage*
	S3	*initpage 3 endpage 7*
	S4	*author nicolas*
	S5	*article title author*
	S6	*initpage 7 article endpage*
	S7	*volume 11 article*
	S8	*asuman pinar article*
	S9	*directions database research*
	S10	*jennifer widom jeffrey d ullman*
	S11	*relational model author date*
	S12	*karen title*
	S13	*anthony data*
	S14	*article data john*
	S15	*database volume number*
	S16	*divesh srivastava database*
	S17	*michael stonebraker postgres*
	S18	*database systems security*
	S19	*christos faloutsos signature files*
	S20	*efficient maintenance materialized views subrahmanian*

Πίνακας 4.2: Ερωτήσεις πειραματικής μελέτης RTCluster.

[37], S12-S15 από την [30] και M17-M20 και S16-S20 από την [2]. Ο χρόνος εντοπισμού για τις προσεγγίσεις *RTCluster* και το *XMean* για όλες τις ερωτήσεις και στα δύο σύνολα δεδομένων παρουσιάζονται στις Εικόνες 4.14α′ και 4.14β′, αντιστοίχως. Όπως είναι φανερό, η προσέγγισή μας ξεπερνά το σύστημα *XMean* σχεδόν σε όλες τις ερωτήσεις και στα δύο σύνολα δεδομένων, επιτυγχάνοντας ταχύτερους χρόνους εντοπισμού. Αυτό αποδεικνύει ότι η ιεραρχία των συστάδων, που παράγεται από το



(α′) Mondial



(β′) SIGMOD

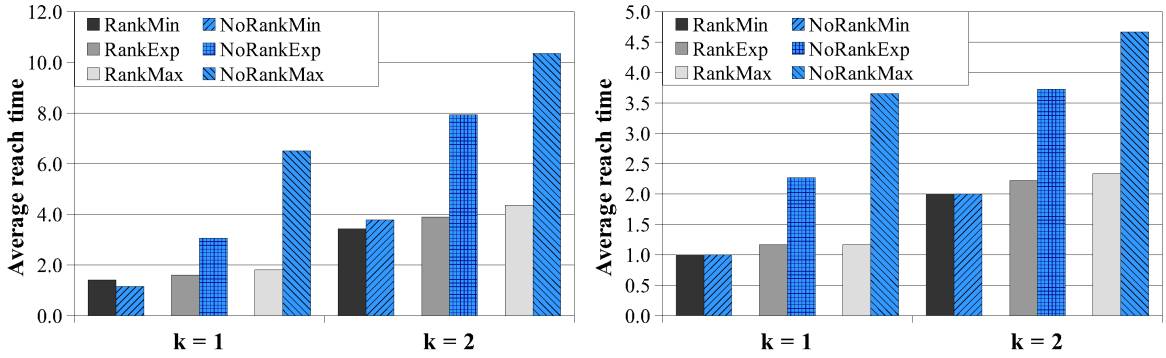
Σχήμα 4.14: Χρόνος πρόσβασης (για ανάκτηση όλων των αποτελεσμάτων) για τις ερωτήσεις του Πίνακα 4.2 στα σύνολα δεδομένων Mondial και SIGMOD.

RTCluster, βοηθά πιο αποτελεσματικά τους χρήστες να ανακτήσουν τα αποτελέσματα που τους ενδιαφέρουν. Οι μέσες τιμές χρόνου εντοπισμού για όλες τις ερωτήσεις και στα δύο σύνολα δεδομένων που εξετάζουμε για τις δύο προσεγγίσεις, που περιλαμβάνονται στον Πίνακα 4.3 επιβεβαιώνουν αυτό το συμπέρασμα.

Ανάκτηση το πολύ k σχετικών προτύπων. Σ′ αυτό το σενάριο, η ταξινόμηση των συστάδων επηρεάζει το χρόνο εντοπισμού. Για το λόγο αυτό, συγκρίνουμε πρώτα την ανάκτηση το πολύ k σχετικών προτύπων με και χωρίς ταξινόμηση των συστάδων. Οι Εικόνες 4.15α′ και 4.15β′ παρουσιάζουν τις μέσες τιμές χρόνου εντοπισμού γι′ αυτές τις δύο παραλλαγές του *RTCluster* κατά την αποτίμηση των ερωτήσεων του Πίνακα 4.2 στα σύνολα δεδομένων Mondial και SIGMOD, αντίστοιχα, για $k = 1$ και $k = 2$. Αξίζει να υπενθυμίσουμε, ότι για ένα πρότυπο, μπορεί να υπάρχουν πολλά επιμέρους αποτελέσματα. Για κάθα παραλλαγή, τρεις τιμές χρόνου εντοπισμού δίνονται: ο ελάχιστος, ο αναμενόμενος και ο μέγιστος. Όπως φαίνεται, η ταξινόμηση των συστάδων μειώνει το χρόνο εντοπισμού κατά 50% περίπου, στο μέσο όρο και των δύο συνόλων δεδομένων. Αυτή η βελτίωση είναι ιδιαίτερα ορατή στο μέγιστο χρόνο εντοπισμού. Συνεπώς, η

Dataset	Approach	Avg. Reach time	Avg. Hierarchy size
Mondial	*RTCluster*	5.35	14.60
	XMean	7.35	23.30
SIGMOD	*RTCluster*	3.75	33.35
	XMean	5.55	142.25

Πίνακας 4.3: Μέσος χρόνος πρόσβασης (για ανάκτηση όλων των σχετικών αποτελεσμάτων) και μέγεθος ιεραρχίας για τις ερωτήσεις του Πίνακα 4.2.



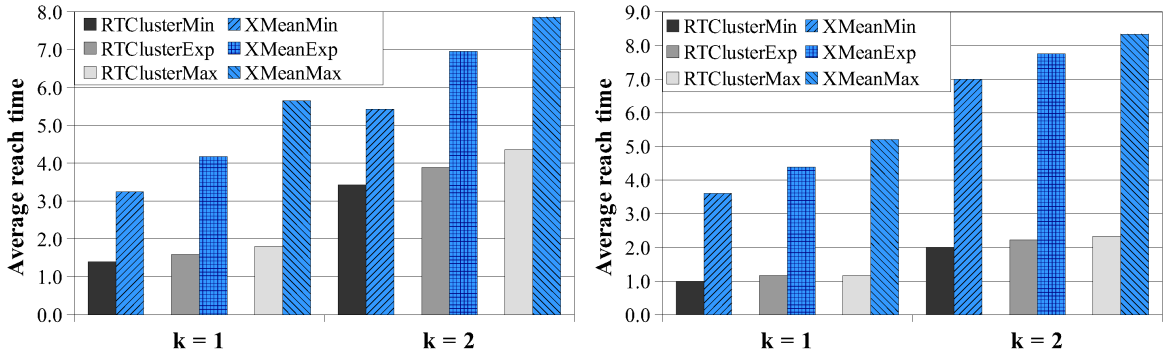
(α′) Mondial (β′) SIGMOD

Σχήμα 4.15: Μέσος ελάχιστος, αναμενόμενος και μέγιστος χρόνος πρόσβασης (για ανάκτηση το πολύ k προτύπων) για την προσέγγιση *RTCluster* με και χωρίς ταξινόμηση των συστάδων για τις ερωτήσεις του Πίνακα 4.2.

τεχνική ταξινόμησης συστάδων που προτείνεται βελτιώνει σημαντικά την ποιότητα του συστήματος ιεραρχικής συσταδοποίησης, μειώνοντας τον αριθμό αντιπροσώπων των συστάδων που ο χρήστης χρειάζεται να εξετάσει.

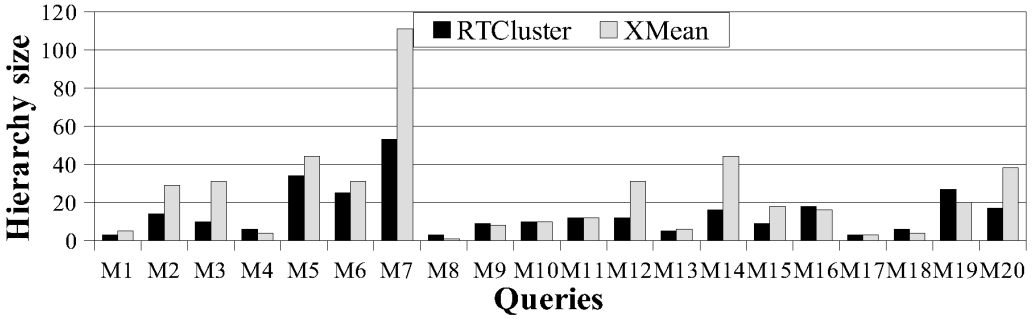
Επιλέον, γίνεται σύγκριση του *RTCluster* (με ταξινόμηση συστάδων) και του *XMean* στην ανάκτηση το πολύ k σχετικών προτύπων. Οι Εικόνες 4.16α′ και 4.16β′ δείχνουν τις μέσες τιμές χρόνων πρόσβασης για τα συστήματα *RTCluster* και *XMean*, για $k = 1$ και $k = 2$ και τις ερωτήσεις του Πίνακα 4.2. Όπως και στην περίπτωση της ανάκτησης όλων των αποτελεσμάτων, το *RTCluster* ξεπερνά το *XMean* σε όλες τις ερωτήσεις και των δύο συνόλων δεδομένων. Ωστόσο, η διαφορά τους μεγιστοποιείται στο σενάριο ανάκτησης το πολύ k σχετικών προτύπων. Αυτό οφείλεται στην τεχνική ταξινόμησης των συστάδων που προτείνεται, η οποία αυξάνει την αποτελεσματικότητα της προσέγγισης, όπως φάνηκε και στην προηγούμενη παράγραφο. Η δυνατότητα αυτή δεν προσφέρεται από την προσέγγιση *XMean*.

Μέγεθος ιεραρχίας συστάδων. Στην παράγραφο αυτή παρουσιάζεται η σύγκριση των δύο προσεγγίσεων σε σχέση με το μέγεθος της ιεραρχίας που κατασκευάζουν. Τό μέγεθος της ιεραρχίας εκφράζεται από τον αριθμό των κόμβων (δηλ. συστάδων) που περιέχει. Οι Εικόνες 4.17α′ και 4.17β′ δείχνουν τα μεγέθη των ιεραρχιών που

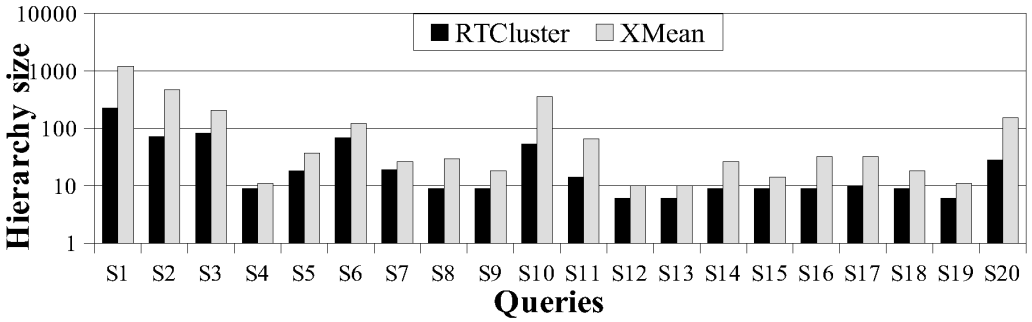


(α′) Mondial (β′) SIGMOD

Σχήμα 4.16: Μέσος ελάχιστος, αναμενόμενος και μέγιστος χρόνος πρόσβασης (για ανάκτηση το πολύ k προτύπων) για τις προσεγγίσεις *RTCluster* και *XMean*, για τις ερωτήσεις του Πίνακα 4.2.



(α′) Mondial



(β′) SIGMOD

Σχήμα 4.17: Μέγεθος ιεραρχίας των *RTCluster* και *XMean* για τις ερωτήσεις του Πίνακα 4.2

κατασκευάζονται από τα δύο συστήματα για τις ερωτήσεις του Πίνακα 4.2. Τα μέσα μεγέθη για το σύνολο των ερωτήσεων καταγράφονται στον Πίνακα 4.3. Παρατηρήστε ότι η κλίμακα του άξονα των y, στην Εικόνα 4.17β′, είναι λογαριθμική. Οι ιεραρχία που κατασκευάζεται από το *RTCluster* είναι σημαντικά μικρότερη από αυτή του *XMean* στις περισσότερες περιπτώσεις.

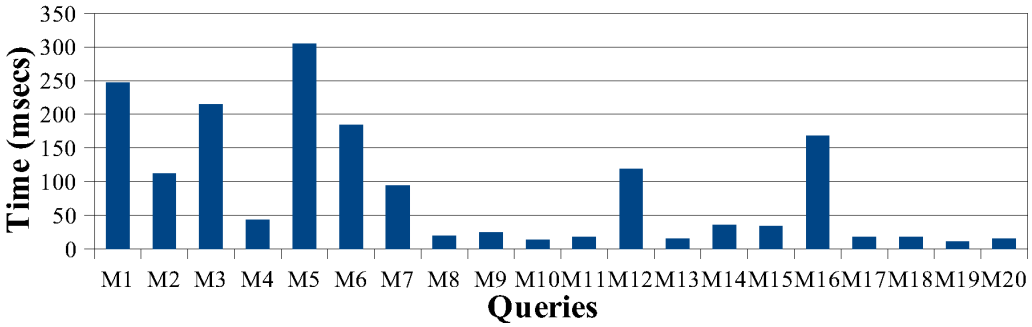
Συνοπτικά, το *XMean* δημιουργεί την ιεραρχία χαλαρώοντας δομικά τα πρότυπα των αποτελεσμάτων. Το *XMean* αναγκατικά προσθέτει πολυάριθμα επιλέον πρότυπα σαν εσωτερικούς κόμβους στην ιεραρχία, τα οποία δεν έχουν αντίκρυσμα στο σύνολο των αποτελεσμάτων. Αυτό το χαρακτηριστικό επιδρά αρνητικά στο ύψος και κατ′ επέκταση το συνολικό μέγεθος της ιεραρχίας, αυξάνοντας παράλληλα και τον αριθμό των κόμβων που ένας χρήστης χρειάζεται να εξετάσει κατά την περιήγησή του. Αντίθετα, για το *RTCluster* το ύψος της ιεραρχίας είναι πάντα ίσο με 3. Οι εσωτερικοί κόμβοι αντιστοιχούν σε πραγματικά πρότυπα των αποτελεσμάτων μιας ερώτησης, καθώς πρόκειται για τους αντιπροσώπους κλάσεων, και το ίδιο πρότυπο δεν εξετάζεται πάνω από μια φορά κατά την περιήγηση στην ιεραρχία. Επιπροσθέτως, το *RTCluster* ταξινομεί τις συστάδες σε κάθε επίπεδο, μια δυνατότητα, που το σύστημα *XMean* δεν προσφέρει. Έτσι, παρόλο που οι δυο προσεγγίσεις ξεκινούν με τον ίδιο αριθμό προτύπων (δηλ. τον ίδιο αριθμό φύλλων στην ιεραρχία), κατά μέσο όρο, ο χρόνος αναζήτησης μέσα στην ιεραρχία του *RTCluster* είναι μικρότερος από αυτόν που χρειάζεται ο χρήστης για περιήγηση στο *XMean*.

4.4.3 Επίδοση αλγορίθμου αποτίμησης και συσταδοποίησης

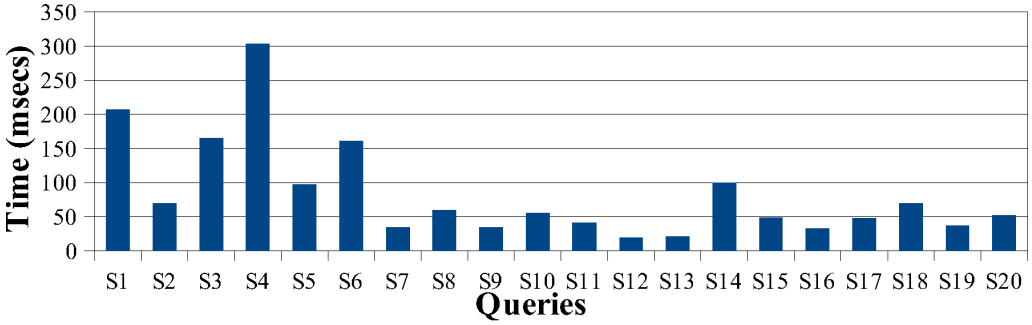
Για την αποδοτικότητα του αλγορίθμου ClusterStack, μελετήσαμε πειραματικά το χρόνο αποτίμησης ερωτήσεων αλλά και την κλιμάκωση του αλγορίθμου με μεταβαλλόμενο

αριθμό από λέξεις-κλειδιά και μέγεθος εισόδου.

Χρόνος αποτίμησης. Οι Εικόνες 4.18α΄ και 4.18β΄ παρουσιάζουν τους χρόνους αποτίμησης των ερωτήσεων του Πίνακα 4.2 από τον αλγόριθμο *ClusterStack*. Ο χρόνος αποτίμησης περιλαμβάνει όλες τις φάσεις επεξεργασίας, δηλ. τον υπολογισμό των αποτελεσμάτων, τη συσταδοποίησή τους και την κατασκευή της ιεραρχίας τους. Όπως είναι φανερό, όλοι οι χρόνοι αποτίμησης είναι μικρότεροι από μισό δευτερόλεπτο και στα δύο σύνολα δεδομένων των ερωτήσεων, δηλ. του Mondial και του ISGMOD. Αυτή η επίδοση είναι συγκρίσιμη με πραγματικά εμπορικά συστήματα, παρόλο που πρόκειται για μια πρωτότυπη υλοποίηση χωρίς τις βελτιστοποιήσεις ενός τέτοιου συστήματος.



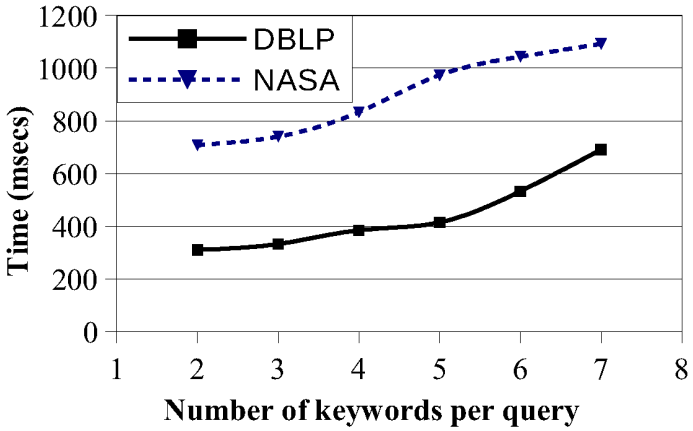
(α΄) Mondial



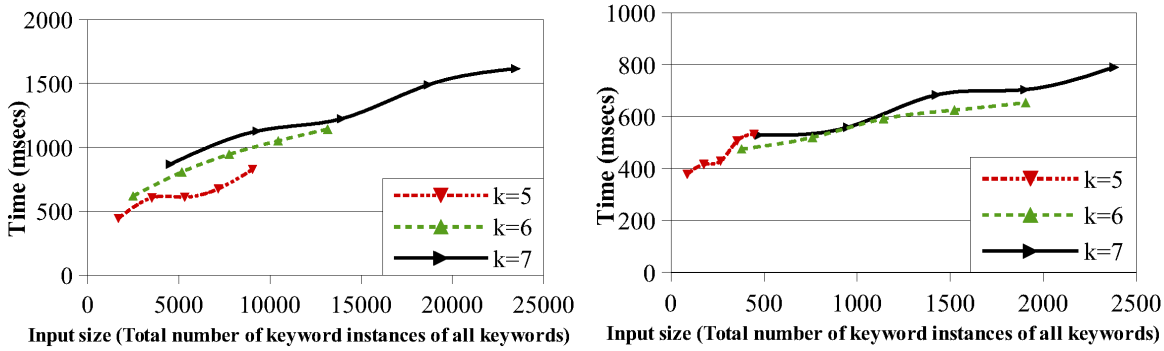
(β΄) SIGMOD

Σχήμα 4.18: Χρόνος υπολογισμού για τις ερωτήσεις του Πίνακα 4.2.

Κλιμάκωση αλγορίθμου. Για τη μελέτη της κλιμάκωσης του αλγορίθμου, χρησιμοποιήθηκαν τα σύνολα δεδομένων DBLP και NASA. Από αυτά τα σύνολα δεδεομένων,



Σχήμα 4.19: Μέσος χρόνος υπολογισμού του αλγορίθμου *ClusterStack* σε σχέση με τον αριθμό των λέξεων-κλειδιών, για ερωτήσεις 2 έως 7 λέξεων-κλειδιών στα σύνολα δεδομένων DBLP και NASA.



(α′) NASA (β′) DBLP

Σχήμα 4.20: Χρόνος υπολογισμού του αλγορίθμου *ClusterStack* σε σχέση με το μέγεθος της εισόδου, για ερωτήσεις 5,6 και 7 λέξεων-κλειδιών στα σύνολα δεδομένων DBLP και NASA.

λόγω του μεγέθους τους, προκύπτουν λίστες μεγάλου μήκους, οι οποίες επιδέχονται περικοπή σε διάφορα σημεία για τη μελέτη της κλιμάκωσης.

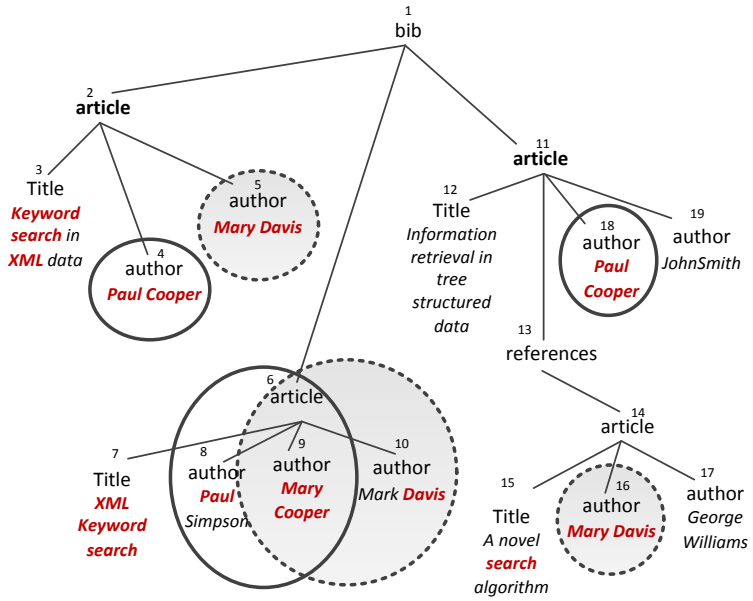
Για την παρατήρηση της κλιμάκωσης του αλγορίθμου σε συνάρτηση με τον αριθμό των λέξεων-κλειδιών, δημιουργήθηκαν τυχαία 10 ερωτήσεις 7 λέξεων-κλειδιών και στα δύο σύνολα δεδομένων. Για κάθε μία απ΄ αυτές τις ερωτήσεις παράχθηκαν έξι υποερωτήσεις με 2 έως 7 λέξεις-κλειδιά αφαιρώντας σταδιακά λέξεις-κλειδιά. Ετσι για καθε αριθμό λέξεων-κλειδιών προέκυψαν 10 ερωτήσεις. Η Εικόνα 4.19 δείχνει τους μέσους χρόνους αποτίμησης των δέκα ερωτήσεων για κάθε μήκος ερώτησης. Όπως φαίνεται στο διάγραμμα, οι χρόνοι αποτίμησης κλιμακώνονται ομαλά από 300msec έως 800msec για το DBLP και από 700msec έως 1200msec για το NASA , μεταβάλλοντας τον αριθμό από λέξεις-κλειδιά από 2 έως 7.

Για τη μελέτη της κλιμάκωσης σε σχέση με το μέγεθος της εισόδου, λαμβάνονται υπόψη τα μήκη των ανεστραμμένων λιστών των λέξεων-κλειδιών των ερωτήσεων των πειραμάτων. Επιλέχθηκαν με τυχαίο τρόπο 7 λέξεις-κλειδιά από κάθε σύνολο δεδομένων. Απ΄ αυτά τα σύνολα κατασκευάστηκαν τρεις ερωτήσεις 3, 5 και 7 λέξεων-κλειδιών, αντίστοιχα. Οι ανεστραμμένες λίστες περικόπηκαν για κάθε πείραμα σε διαφορετικά μήκη, δηλ. στο 20%, 40%, 60% και 80%. Οι χρόνοι υπολογισμού παρουσιάζονται στα διαγράμματα των Εικόνων 4.20α΄ και 4.20β΄. Όπως φαίνεται, οι καμπύλες είναι σχεδόν γραμμικές, οπότε οι χρόνοι αποτίμησης κλιμακώνονται ομαλά. Τετραπλασιάζονται στη χειρότερη περίπτωση, όταν και το μέγεθος εισόδου επίσης τετραπλασιάζεται.

Κεφάλαιο 5

Συνεκτικότητα λέξεων - κλειδιών

Στο πεδίο της κλασικής ανάκτησης πληροφορίας (information retrieval), εκτός από τις απλές ερωτήσεις λέξεων - κλειδιών έχουν οριστεί γλώσσες ερωτήσεων που περιέχουν ολόκληρες φράσεις προς αναζήτηση στη βάση δεδομένων. Σ΄ αυτήν την περίπτωση, η απαίτηση είναι να εντοπιστεί μια ακολουθία λέξεων μέσα στο κείμενο και όχι οι μεμονωμένες λέξεις. Όπως μελετήσαμε στο πλαίσιο αυτής της διατριβής, σε ημιδομημένα δεδομένα, υπάρχει η ανάγκη ομαδοποίησης λέξεων - κλειδιών, αλλά με τελείως διαφορετική σημασιολογία.



Σχήμα 5.1: Παράδειγμα υποδέντρου βιβλιογραφικής βάσης δεδομένων

Ας πάρουμε το παράδειγμα μιας βιβλιογραφικής βάσης δεδομένων, παράδειγμα της οποίας απεικονίζεται στο Σχήμα 5.1. Έστω ότι ένας χρήστης εκτελεί μια αναζήτηση με την ερώτηση Q_1 = {`XML Paul Cooper Mary Davis`}. Τόσο το άρθρο article (2) όσο και το άρθρο article (6) είναι έγκυρες απαντήσεις στην ερώτησή του. Αν όμως ο χρήστης ενδιαφερόταν για άρθρα, συγκεκριμένα της Mary Davis και του Paul Cooper, τότε το άρθρο article (6) δεν είναι ένα καλό αποτέλεσμα για την αναζήτησή του. Προτείνουμε, λοιπόν, τον εμπλουτισμό των ερωτήσεων λέξεων - κλειδιών με τα εργαλεία εκείνα που θα δίνουν τη δυνατότητα στο χρήστη να περιορίζει την ασάφεια της ερώτησής του, χωρίς να τον επιβαρύνουν με την ανάγκη εκμάθησης μιας δομημένης σύνταξης.

Αν ένας χρήστης μπορεί να εκφράσει το γεγονός ότι τα στιγμιότυπα των Mary και Davis θα έπρεπε να σχηματίζουν μια αδιαίρετη οντότητα, χωρίς να επιτρέπεται οι υπόλοιπες λέξεις-κλειδιά να υπεισέρχονται σάυτήν (δηλ. μια συνεκτική μονάδα), τότε το σύστημα θα μπορούσε να επιστρέφει πιο ακριβή αποτελέσματα. Κατά συνέπεια, θα μπορούσε να αποκλείει δημοσιεύσεις με θέμα XML και συγγραφέα τη Mary Cooper. Τέτοιες δημοσιεύσεις είναι άσχετες μ΄ αυτό που αναζητά ο χρήστης, και καμία από τις προηγούμενες προσεγγίσεις φιλτραρίσματος (π.χ. ELCA, VLCA, CVLCA, SLCA, MLCA, MaxMach κ.τ.λ.) δε θα μπορούσε μέσω της σημασιολογίας της να τις απορρίπτει. Εξάλλου, ο ορισμός συνεκτικών σχέσων εξοικονεί χρόνο στον υπολογισμό, που στις άλλες περιπτώσεις σπαταλάται στην αναζήτηση τέτοιου είδους άσχετων με το σκοπό του χρήστη αποτελεσμάτων. Ορίζουμε, λοιπόν, συνεκτικές σχέσεις μεταξύ λέξεων-κλειδιών περικλείοντάς τις σε ένα ζεύγος παρενθέσεων. Έτσι, η ερώτηση Q_1 διατυπώνεται ως `(XML (Paul Cooper) (Mary Davis))`.

Η αξία του γεφυρώματος της απόστασης μεταξύ επίπεδων, τελείως αδόμητων ερωτήσεων με λέξεις-κλειδιά και δομημένων ερωτήσεων που διατυπώνονται σε μια επίσημη γλώσσα ερωτήσεων, με στόχο τη βελτίωση της αποτελεσματικότητας ενός συστήματος αναζήτησης έχει επισημανθεί και στο παρελθόν σε επίπεδο κείμενο. Η μηχανή αναζήτησης Google[1] δίνει τη δυνατότητα σε ένα χρήστη να χρησιμοποιήσει εισαγωγικά για να δηλώσει την αντιστοίχηση μιας φράσης αυτούσιας στα επιστρεφόμενα αποτελέσματα. Παρ΄ όλ΄ αυτά, η χρήση συνεκτικών σχέσεων σε ένα σύνολο από λέξεις-κλειδιά είναι μια τελείως διαφορετική προσέγγιση από την αναζήτηση φράσεων σε επίπεδο κείμενο στο πεδίο του IR. Μία συνεκτική σχέση μεταξύ λέξεων-κλειδιών δεν επιβάλει συντακτικούς περιορισμούς (π.χ. τη σειρά εμφάνισης) στον εντοπισμό των στιγμιοτύπων τους στο δέντρο δεδομένων. Αυτό που ορίζεται είναι μόνο ότι όπως και αν βρεθούν αυτές οι λέξεισ-κλειδιά θα πρέπει να ορίζουν μια συνεκτική οντότητα σε σχέση με τις υπόλοιπες λέξεις-κλειδιά της ερώτησης.

Οι συνεκτικές σχέσεις μπορεί να εμφανίζονται εμφωλευμένες μεταξύ τους. Η ερώτηση `(Information Retrieval (Paul Cooper) (references (Mary Davis)))`, για παρέδειγμα, αναζητά κάποια δημοσίευση στο πεδίο του IR του Paul Cooper που κάνει αναφορά σε δουλειά της Mary Davis. Τέλος, επιτρέπεται η επανάληψη λέξεων-κλειδιών σε μια ερώτηση. Έτσι, μια ερώτηση σαν την προηγούμενη θα μπορούσε να είναι `(Information Retrieval (Mary Cooper) (references (Mary Davis)))`, με ανάλογο νήμα όπως και η προηγούμενη, μόνο που ο συγγραφέας της αναζητούμενης δημοσίευσης θα έπερεπε να είναι η Mary Cooper αντί του Paul Cooper.

Οι συνεκτικές σχέσεις, λοιπόν, εκφράζουν καλύτερα το σκοπό αναζήτησης του χρήστη ενώ ταυτόχρονα δεν υπονομεύουν την απλότητα των ερωτήσεων με λέξεις-κλειδιά. Παρά την εκφραστική του δύναμη, διατηρούν και τις δύο ιδιότητες των απλών ερωτήσεων με λέξεις-κλειδιά: δεν απαιτούν καμία πρώτερη γνώση ούτε κάποιας εξειδικευμένης γλώσσας ερωτήσεων ούτε το σχήματος των δεδομένων στα οποία διενεργείται η αναζήτηση. Οι χρήστες μπορούν φυσικά να ορίσουν συνεκτικές σχέσεις σε μια κοινή ερώτηση με λέξεις-κλειδιά. Το κέρδος, ωτόσο, στην ποιότητα της απάντησης που θα λάβουν αλλά και στην ταχύτητα απόκρισης είναι ξεκάθαρο, όπως δε δειχθεί και στη συνέχεια.

[1] http://www.google.com

5.1 Γλώσσα ερωτήσεων με συνεκτικές σχέσεις λέξεων-κλειδιών

Στο μοντέλο δεδομένων στο οποίο ορίζεται η γλώσσα ερωτήσεων που εισάγεται σ' αυτήν την ενότητα, η πληροφορία αναπαρίσταται ως δέντρο με ετικέτες στους κόμβους του. Κάθε κόμβος διαθέτει επιπλέον ένα αναγνωριστικό και πιθανώς μια τιμή. Τα αναγνωριστικά ακολουθούν την κωδικοποίηση Dewey [10], σύμφωνα με την οποία αποδίδονται αναγνωρστικά στους κόμβους ενός δέντρου με προδιατεταγμένη σειρά και εκφράζει φυσικά τις σχέσεις προγόνου-απογόνου και πατέρα-παιδιού μεταξύ των κόμβων ενός δέντρου. Μία λέξη-κλειδί k μπορεί να εμφανίζεται στην ετικέτα ή την τιμή ενός κόμβου n μία ή περισσότερες φορές. Στην περίπτωση αυτή, ο κόμβος n καλείται στιγμιότυπο της λέξης-κλειδιού k. Με βάση τα ροηγούμενα, ένα κόμβος μπορεί να αποτελεί στιγμιότυπο πολλών λέξεων-κλειδιών.

5.1.1 Σημασιολογία συνεκτικότητας

Μια *ερώτηση συνεκτικών λέξεων-κλειδιών* ορίζεται ως μια ερώτηση με λέξεις-κλειδιά, η οποία μπορεί να περιέχει και ομάδες λέξεων-κλειδιών, που ονομάζονται *συνεκτικοί όροι*. Διαισθητικά, ένας συνεκτικός όρος εκφράζει μια σχέση συνεκτικότητας ανάμεσα στις λέξεις-κλειδιά ή τους υπόλοιπους όρους που περιέχει. Μια ερώτηση συνεκτικών λέξεων-κλειδιών ορίζεται αυστηρά ως εξής:

Ορισμός 5.1 (Ερώτηση συνεκτικών λέξεων-κλειδιών)**.** *Ένας* συνεκτικός όρος *είναι ένα πολυσύνολο από τουλάχιστον δύο λέξεις-κλειδιά ή συνεκτικούς όρους. Μία* ερώτηση συνεκτικών λέξεων-κλειδιών *είναι: α) το σύνολο μίας μοναδικής λέξης-κλειδιού, ή β) ένας συνεκτικός όρος. Τα σύνολα και τα πολυσύνολα οριοθετούνται σε μια ερώτηση με τη χρήση παρενθέσεων.*

Για παράδειγμα, η έκφραση `((title XML) ((John Smith) author))` είναι μία τέτοια ερώτηση. Μερικοί από τους συνεκτικούς της όρους είναι οι T_1 = `(title XML)`, T_2 = `((John Smith) author)`, T_3 = `(John Smith)`, και λέμε ότι ο όρος T_3 είναι *εμφωλευμένος* στον T_2.

Μία λέξη-κλειδί μπορεί να εμφανίζεται πολλές φορές σε μια ερώτηση. Για παράδειγμα, στην ερώτηση `((journal (Information Systems) ((Information Retrieval) Smith))` η λέξη-κλειδί `Information` εμφανίζεται δύο φορές, μία στον όρο `(Information Systems)` και μία στον `(Information Retrieval)`

Στη συνέχεια αυτού του κεφαλαίου, θα αναφερόμαστε στις ερωτήσεις συνεκτικών λέξεων-κλειδιών απλά ως συνεκτικές ερωτήσεις ή ερωτήσεις, για συντομία. Το συντακτικό της γλώσσας ερωτήσεων με συνεκτικές λέξεις-κλειδιά ορίζεται από την ακόλουθη γραμματική, στην οποία το μη τερματικό σύμβολο T υποδηλώνει ένα συνεκτικό όρο και το τερματικό σύμβολο k μία λέξη-κλειδί:

$$\begin{array}{lcl} Q & \rightarrow & (k) \mid T \\ T & \rightarrow & (S\ S) \\ S & \rightarrow & S\ S \mid T \mid k \end{array}$$

Ακολουθεί, τώρα, ο ορισμός της σημασιολογίας των ερωτήσεων συνεκτικών λέξεων-κλειδιών. Οι ερωτήσεις με λέξεις-κλειδιά εντίθενται σε δέντρα δεδομένων. Για να οριστεί η απάντηση μιας ερώτησης, χρειάζεται να εισαχθεί η έννοια της ένθεσης. Στις συνεκτικές ερωτήσεις, m εμφανίσεις της ίδιας λέξης-κλειδιού σε μια ερώτηση, αντιστοιχίζεται σε ένα ή περισσότερα στιγμιότυπα αυτής της λέξης-κλειδιού, αρκεί τα στιγμιότυπα

αυτά να περιέχουν συνολικά m φορές την εν λόγω λέξη-κλειδί. Οι συνεκτικές ερωτήσεις μπορεί, επίσης, να περιέχουν όρους, οι οποίοι όπως ήδη αναφέρθηκε εκφράζουν μια συνεκτική σχέση μεταξύ των στιγμιοτύπων των λέξεων-κλειδιών που περιέχουν. Σε δενδρικά δεδομένα, τα στιγμιότυπα των λέξεων-κλειδιών σε ένα δέντρο δεδομένων (δηλ. οι κόμβοι που τις περιέχουν) εκπροσωπούνται από τον χαμηλότερο κοινό τους πρόγονο, δηλ. τον LCA τους [46, 19, 39]. Τα στιγμιότυπα των λέξεων-κλειδιών σ΄ ένα δέντρο δεδομένων πρέπει να σχηματίζουν μια συνεκτική οντότητα. Αυτό σημαίνει πως το υποδέντρο με ρίζα τον LCA των στιγμιοτύπων των λέξεων-κλειδιών του συνεκτικού όρου πρέπει να είναι αδιαπέραστο από τα στιγμιότυπα των λέξεων-κλειδιών της ερώτησης που δεν περιλαμβάνονται σ΄ αυτόν τον όρο. Γι΄ αυτό, αν l είναι ο LCA ενός συνόλου στιγιοτύπων των λέξεων-κλειδιών σε έναν όρο T, i είναι ένα απ΄ αυτά τα στιγμιότυπα και i' ένα στιγμιότυπο μις λέξης-κλειδιού εκτός του T, τότε $lca(i', i) = lca(i', l) \neq l$.

Σαν παράδειγμα, ας θεωρήσουμε την ερώτηση Q_1 =(`XML keyword search (Paul Cooper) (Mary Davis)`), που τίθεται στο δέντρο δεδομένων D_1 της Εικόνας 5.1. Στην Εικόνα 5.1, τα στιγμιότυπα των λέξεων-κλειδιών υποδεικνύονται με έντονα γράμματα και τα στιγμιότυπα των συνεκτκών όρων φέρουν κυκλικό περίγραμμα. Η αντιστοίχιση, που αντιστοιχίζει το `Paul` στον κόμβο 8, το `Mary` και το `δοπερ` στον κόμβο 9 και το `Davis` στον κόμβο 10, δεν είναι μια ένθεση της ερώτησης Q_1 στο δέντρο D_1, καθώς το στιγμιότυπο `Mary` υπεισέρχεται στο περιγεγραμμένο υποδέντρο των στιγμιοτύπων `Paul` και `Cooper` με ρίζα τον κόμβο article (6): οι δύο κύκλοι του κόμβου 6 επικαλύπτονται. Η ιδέα αυτή ορίζεται επίσημα στη συνέχεια.

Ορισμός 5.2 (Ένθεση συνεκτικής ερώτησης). *Έστω Q μια συνεκτική ερώτηση σε ένα δέντρο δεδομένων D. Μία ένθεση του Q στο D είναι μια συνάρτηση e με πεδίο ορισμού τις εμφανίσεις των λέξεων-κλειδιών στο Q και πεδίο τιμών τα στιγμιότυπα των αντίστοιχων λέξεων-κλειδιών στο D τέτοια ώστε:*

α. αν $k_1, \ldots, k_m$ είναι διαφορετικές εμφανίσεις της ίδιας λέξης-κλειδιού k στο Q και $e(k_1) = \ldots = e(k_m) = n$, τότε ο κόμβος n περιέχει το k τουλάχιστον m φορές.

β. αν $k_1, \ldots, k_n$ είναι οι εμφανίσεις των λέξεων-κλειδιών ενός συνεκτικού όρου T, k μια εμφάνιση λέξης-κλειδιού εκτός του T, και $l = lca(e(k_1), \ldots, e(k_n))$ τότε: ι) $e(k_1) = \ldots = e(k_n)$, ή ιι) $lca(e(k), l) \neq l$.

Δεδομένης μιας ένθεσης e μιας ερώτησης Q με εμφανίσεις λέξεων-κλειδιών $k_1, \ldots, k_m$ σε ένα δέντρο δεδομένων D, το ελάχιστο συνδετικό υποδέντρο (MCT) M του e στο D είναι το ελάχοστο υποδέντρο του D που περιέχει τους κόμβους $e(k_1), \ldots, e(k_m)$. Το δέντρο M, λέγεται επίσης χάριν συντομίας, ότι είναι ένα MCT της ερώτησης Q στο D. Η ρίζα του M είναι ο χαμηλότερος κοινός πρόγονος (LCA) του $e(k_1), \ldots, e(k_m)$ και ορίζει ένα αποτέλεσμα του Q στο D. Για παράδειγμα, οι κόμβοι article (2) και (11) είναι αποτελέσματα της ερώτησης Q_1 στο δέντρο D_1 του παραδείγματος. Αντίθετα, το article (6) δεν είναι αποτέλεσμα του Q_1.

Τα αποτελέσματα μιας ερώτησης ταξινομούνται με βάση το μέγεθος LCA, το οποίο παρουσιάστηκε στην Ενότητα 3.2. Ανάλογα με την αναζήτηση με λέξεις-κλειδιά σε επίπεδο κείμενο, το μέγεθος LCA αντικατοπτρίζει με φυσικό τρόπο την εγγύτητα σύνδεσης των λέξεων-κλειδιών σε ένα υποδέντρο του δέντρου δεδομένων.

Για παράδειγμα, το μέγεθος του αποτελέσματος article (2) για την ερώτηση Q_1 στο δέντρο D_1 είναι 3, ενώ το μέγεθος του αποτελέσματος article (11) είναι 6 (παρατηρήστε τα διαφορετικά MCT με ρίζα τον κόμβο (11), με το μικρότερο απ΄ αυτά να έχει μέγεθος 6.).

Ορισμός 5.3. Απάντηση ερώτησης με συνεκτικές λέξεις-κλειδιά *Η α-*

πάντηση σε μία συνεκτική ερώτηση Q που τίθεται σε ένα δέντρο D είναι μια λίστα $[l_1, \ldots, l_n]$ των LCA του Q στο D τέτοια ώστε $size(l_i) \leq size(l_j), i < j$.

Για παράδειγμα, ο κόμβος article (2) κατατάσσεται ψηλότερα από το article (11) στην απάντηση του Q_1 στο D_1.

5.1.2 Λεπτομερής ταξινόμηση με βάση τους συνεκτικούς όρους των ερωτήσεων

Το μέγεθος LCA αποδίδει φυσικά το βαθμό εγγύτητας των στιγμιοτύπων των λέξεων-κλειδιών μιας ερώτησης στο υποδέντρο ενός LCA. Κάθε LCA, που είναι απάντηση μιας ερώτησης περιέχει μερικούς LCA που αντιστοιχούν στους εμφωλευμένους συνεκτικούς όρους της ερώτησης. Αυτοί οι μερικοί LCA συνεισφέρουν με το μέγεθός τους στο συνολικό μέγεθος του πλήρους LCA, που είναι αποτέλεσμα της ερώτησης. Εκτός αυτού, όμως, είναι διαισθητικά σωστό να ληφθεί υπόψιν και πόσο συμπαγώς είναι συνδεδεμένα μεταξύ τους τα στιγμιότυπα των λέξεων-κλειδιών και στους μερικούς LCA, που αντιστοιχούν στους επιμέρους συνεκτικούς όρους μιας ερώτησης. Ας θεωρήσουμε, για παράδειγμα, το δέντρο D_1 της Εικόνας 5.1 και την ερώτηση Q_1=`(XML keyword search (Paul Cooper) (Mary Davis))`. Ο κόμβος article (2) είναι ένας LCA για το Q_1 και ο κόμβος author (4) είναι ένας μερικός LCA που αντιστοιχεί στο συνεκτικό όρο (`Paul Cooper`), συνεισφέροντας με το μέγεθος 0 στο συνολικό μέγεθος του LCA (2). Το γεγονός ότι τα στιγμιότυπα των `Paul` και `Cooper` είναι πολύ συμπαγώς συσχετισμένα, σχηματίζοντας ένα μερικό LCA με μέγεθος 0, είναι εξίσου σημαντικό με το συνολικό μέγεθος του αποτελέσματος article (2).

Για τον αναλογισμό της εγγύτητας των λέξεων-κλειδιών εντός ενός συνεκτικού όρου, προτείνεται ένα νέο μοντέλο ταξινόμησης, το οποίο λαμβάνει υπόψιν τους συνεκτικούς όρους μιας ερώτησης και των μεγεθών των μερικών LCA, που τους αντιστοιχούν, σε κάθε αποτέλεσμα μιας ερώτησης. Το μοντέλο αυτό ταξινόμησης, όχι μόνο προσφέρει ένα λεπτομερέστερο τρόπο κατάταξης σε σχέση με την κλιμακωτή ταξινόμηση με βάση το συνολικό μέγεθος ενός LCA, αλλά επιπλέον παρουσιάζει μερικά ενδιαφέροντα χαρακτηριστικά σε σχέση με άλλα μοντέλα ταξινόμησης. Δεν απαιτεί την προεπεξεργασία του συνόλου δεδομένων για εξαγωγή στατιστικών στοιχείων και ταυτόχρονα λαμβάνει υπόψιν τις λέξεις-κλειδιά της ερώτησης και όχι χαρακτηριστικούς όρους, που έχουν επιλεχθεί μέσω κατάλληλων διαδικασιών εξαγωγής χαρακτηριστικών (feature extraction), από επεξεργασία της συλλογής δεδομένων [4]. Έτσι, υπολογίζονται ad-hoc κατά τη διάρκεια αποτίμησης της ίδιας της ερώτησης.

Κάθε αποτέλεσμα μιας ερώτησης αναπαριστάται ως ένα διάνυσμα στο χώρο των συνεκτικών όρων μιας δεδομένης ερώτησης Q. Έστω ότι το Q είναι μια ερώτηση με m συνεκτικούς όρους, συμπεριλαμβανομένου του εξώτατου όρου, δηλ. της ίδιας της ερώτησης. Κάθε LCA l_j της ερώτησης Q σε ένα δέντρο δεομένων D αναπαριστάται με το ακόλουθο διάνυσμα:

$$\overrightarrow{l_j} = (C_1 s_{1,j}, C_2 s_{2,j}, \ldots, C_m s_{m,j})$$

όπου C_i είναι το βάρος του συνεκτικού όρου T_i στην ερώτηση Q σε σχέση με το σύνολο δεδομένων D και $s_{i,j}$ είναι το μέγεθος του μερικού LCA για τον όρο T_i που συνειφέρεται στον LCA l_j. Διαισθητικά, η παράμετρος C_i εκφράζει τη συνεκτικότητα του όρου T_i στο σύνολο δεδομένων D. Δηλαδή, αποτελεί έναν τρόπο να εκφραστεί ποσοτικά πόσο κοντά συνδέονται οι λέξεις-κλειδιά που περιλαμβάνονται στον όρο T_i

γενικά στο σύνολο δεδομένων D. Έστω P_i το σύνολο των LCA του όρου T_i στο D. Τότε, το βάρος C_i ορίζεται όπως παρακάτω:

$$C_i = \frac{|P_i|}{1 + \sum_{p \in P_i} size(p)}$$

Όσο μικρότερο είναι το μέσο μέγεθος LCA ενός όρου στο σύνολο δεδομένων D τόσο πιο συμπαγής είναι ο όρος στο D, δηλ. τόσο πιο στενά συσχετισμένες οι λέξεις-κλειδιά που περιέχει. Το διάνυσμα $\overrightarrow{l_j}$ ενός LCA l_j χρησιμοποιείται για να οριστεί η βαθολογία του LCA l_j ως εξής:

$$score(l_j) = |\overrightarrow{l_j}|$$

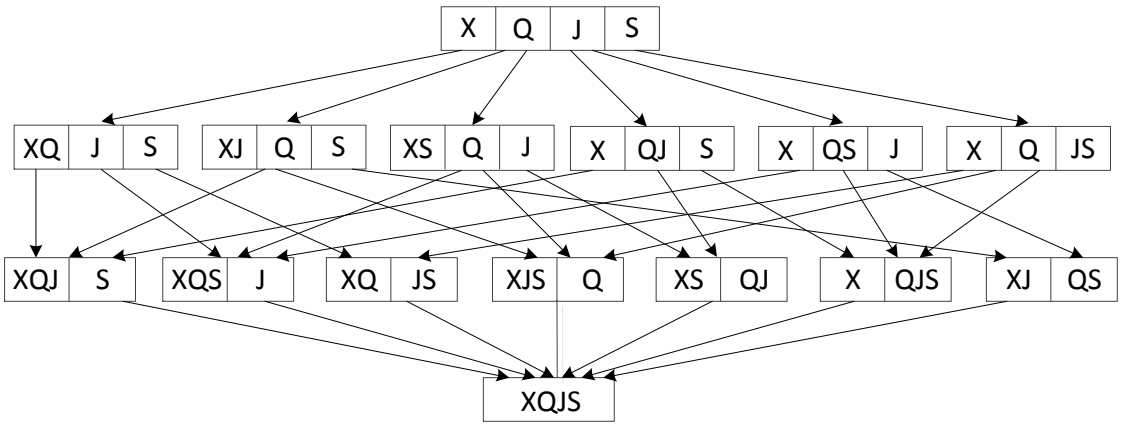
Τα αποτελέσματα μιας ερώτησης ταξινομούνται σε αύξουσα σειρά της βαθμολογίας τους, δηλ. των τιμών της συνάρτησης *score*. Το βάρος C_i επιβραβεύει αποτελέσματα που επιδεικνύουν μικρά μεγέθη για μη συνεκτικούς όρους, εν γένει, στο σύνολο δεδομένων. Ταυτόχρονα τιμωρεί αποτελέσματα, στα οποία όροι, που εμφανίζονται στενά συνδεδμένοι στο σύνολο δεδομένων, συνεισφέρουν απρόσμενα μεγάλα μεγέθη μέσω των μερικών LCA που τους αντιστοιχούν.

5.2 Αποδοτική αποτίμηση ερωτήσεων με συνεκτικές σχέσεις λέξεων - κλειδιών σε δένδρα

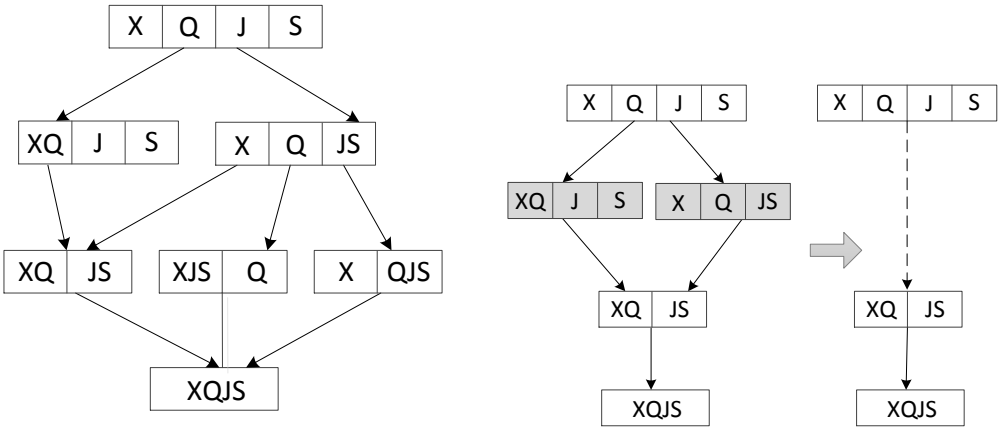
Οι συνεκτικές σχέσεις μεταξύ των λέξεων - κλειδιών εκφράζονται από την ενότητα που ορίζεται από το μερικό χαμηλότερο κοινό πρόγονο των στιγμιοτύπων των συνεκτικών λέξεων. Στο Σχήμα 5.1 για την ερώτηση Q_1, οι χαμηλότεροι κοινοί πρόγονοι του όρου `(Paul Cooper)` είναι οι κόμβοι author (4), article (6) και author (18) ενώ του όρου `(Mary Davis)` οι κόμβοι author (5), article (6) και author (16). Για τα στιγμιότυπα των λέξεων που δε συμμετέχουν σε μια συνεκτική σχέση, το υποδέντρο του χαμηλότερου κοινού προγόνου της συνεκτικής σχέσης λειτουργεί ως μαύρο κουτί. Άλλη λέξη - κλειδί μπορεί να συσχετιστεί με τις λέξεις της συνεκτικής σχέσης μόνο συσχετιζόμενη μέσω του χαμηλότερου κοινού προγόνου της σχέσης. Έτσι η αποτίμηση μιας ερώτησης με συνεκτικές σχέσεις λέξεων - κλειδιών οδηγείται από τους επιτρεπτούς τύπους χαμηλότερων κοινών προγόνων, που ορίζονται από τη σύνταξη της ερώτησης (βλ. Κεφάλαιο 3).

Ο αλγόριθμος LCAsz που εξηγήθηκε στο προηγούμενο κεφάλαιο, προσφέρεται για την αποτίμηση των ερωτήσεων με συνεκτικές σχέσεις, καθώς προοδευτικά δημιουργεί χαμηλότερους κοινούς προγόνους των στιγμιοτύπων ενός συνόλου λέξεων - κλειδιών από το συνδυαμό στιγμιοτύπων και μερικών χαμηλότερων κοινών προγόνων υποσυνόλου των λέξεων - κλειδιών, προχωρώντας από κάτω προς το πάνω στο δέντρο δεδομένων. Αυτό που χρειάζεται είναι να εξαιρεθούν οι συνδυασμοί εκείνοι που οδηγούν σε μη επιτρεπτούς τύπους χαμηλότερων κοινών προγόνων.

Ο αλγόριθμος CohesiveLCA, που είναι μια εξέλιξη του αλγορίθμου LCAsz, υπολογίζει τα αποτελέσματα μιας ερώτησης με συνεκτικές σχέσεις λέξεων-κλειδιών και τα ταξινομεί με σειρά συνολικού μεγέθους, όπως ορίστηκε στην προηγούμενη ενότητα από τη συνάρτηση *score*. Όπως και ο LCAsz έτσι και ο CohesiveLCA αξιοποιεί το δικτυωτό των διαμερίσεων του συνόλου των λέξεων-κλειδιών. Στο πλαίσιο των ερωτήσεων με



(α′) (XML Query John Smith)



(β′) (XML Query (John Smith)) (γ′) ((XML Query) (John Smith))

Σχήμα 5.2: Δικτυωτά των διαμερίσεων του συνόλου των λέξεων - κλειδιών (`XML Query John Smith`) με διαφορετικές συνεκτικές σχέσεις

συνεκτικές σχέσεις, συνεκτικοί όροι μεγαλύτερης πληθυκότητας παράγονται από δυνδυασμό δύο όρων μικρότερης πλυθηκότητας σε προηγούμενο επίπεδο του δικτυωτού. Για την ερώτηση (`XML Query John Smith`), στην οποία δεν έχουν οριστεί συνεκτκές σχέσης πλην της εξωτερικής της ίδιας της ερώτησης, το δικτυωτό αποτυπώνεται στην Εικόνα 5.2α′.

5.2.1 Μείωση της διάστασης του δικτυωτού

Το δικτυωτό των διαμερίσεων των λέξεων-κλειδιών μιας δεδομένης ερώτησης αποτελείται από όλες τις δυνατές διαμερίσεις των λέξεων-κλειδιών. Οι διαμερίσεις αντιστοιχούν σε όλους τους δυνατούς τρόπους που οι λέξεις-κλειδιά μπορούν να συνδυαστούν για να σχηματίσουν μερικούς και τελικά πλήρεις LCA. Οι συνεκτικές σχέσεις περιορίζουν τους τρόπους που τα στιγμιότυπα των λέξεων-κλειδιών μπορούν να συνδυαστούν στην ένθεση μιας ερώτησης για το σχηματισμό ενός αποτελέσματος. Ο περιορισμός προκύπτει από το γεγονός ότι λέξεις-κλειδιά μπορούν να συνδυαστούν μόνο αν ανήκουν στους ίδιους συνεκτικούς όρους: αν μια λέξη-κλειδί a είναι 'κρυμμένη' από μια άλλη λέξη-κλειδί b εντός ενός συνεκτικού όρου T_a, τότε ένα στιγμιότυπο του b μπορεί να

συνδυαστεί μόνο με έναν LCA στιγμιοτύπων όλων των λέξεων-κλειδιών που περιλαμβάνει ο T_a και όχι μεμονωμένα με ένα στιγμιότυπο του a. Αυτός ο περιορισμός στους επιτρεπτούς συνδυασμούς λέξεων-κλειδιών οδηγεί σε σημαντική μείωση του μεγέθους του δικτυωτού όπως καταδεικνύεται με το επόμενο παράδειγμα.

Οι Εικόνες 5.2β΄ και 5.2γ΄ απεικονίζουν τα δικτυωτά των διαμερίσεων των λέξεων-κλειδιών δύο συνεκτικών ερωτήσεων. Οι ερωτήσεις περιλαμβάνουν τις ίδιες λέξεις-κλειδιά, δηλ. `XML`, `Query`, `John` και `Smith` αλλά συσχετισμένες με διαφορετικές συνεκτικές σχέσεις μεταξύ τους. Το δικτυωτό της Εικόνας 5.2α΄ είναι το πλήρες δικτυωτό 15 διαμερίσεων των λέξεων-κλειδιών και επιτρέπει όλους τους δυνατούς συνδυασμούς λέξεων-κλειδιών και μερικών LCA που αντιστοιχούν σε υποσύνολα των λέξεων-κλειδιών. Η ερώτηση της Εικόνας 5.2β΄ επιβάλλει μια συνεκτική σχέση σνάμεσα στα `John` και `Smith`. Αυτή η συνεκτική σχέση καθιστά αρκετές διαμερίσεις του πλήρους δικτυωτού της Εικόνας 5.2α΄ μη χρησιμοποιήσιμες. Για παράδειγμα, στην Εικόνα 5.2β΄, η διαμέριση `[XJ, Q, S]` απαλείφεται, εφόσον ένα στιγμιότυπο του `XML` δεν μπορεί να συνδυαστεί με ένα στιγμιότυπο του `John`, εκτός και το στιγμιότυπο του `John` έχει ήδη συνδυαστεί με ένα στογμιότυπο του `Smith`, όπως στην περίπτωση της διαμέρισης `[XJS, Q]`. Κατά συνέπεια η συνεκτική σχέση που ορίζεται μεταξύ `John` και `Smith` μειώνει το μέγεθος του δικτυωτού από 15 σε 7. Μια δεύτερη συνεκτική σχέση μεταξύ `XML` και `Query` μειώνει ακόμα περισσότερο τη διάσταση του δικτυωτού σε 3 διαμερίσεις, όπως φαίνεται στην Εικόνα 5.2γ΄. Παρατηρούμε ότι, σ΄ αυτήν την περίπτωση, εκτός από διαμερίσεις που καταργούνται ως μη επιτρεπτές από την επιπλέον συνεκτική σχέση λέξεων-κλειδιών (π.χ. η διαμέριση `[XJS, Q]`), υπάρχουν μερικλες διαμερίσεις ακόμα, που καθίστανται μη παραγωγικές άρα και μη αναγκαίες. Μία τέτοια διαμέριση είναι η `[XQ, J, S]`. Ο μόνος έγκυρος συνδυασμός, που μπορεί να παραχθεί από τη διαμέριση αυτή, είναι η `[XQ, JS]`, η οποία είναι μια διαμέριση που μπορεί να προκύπσει απ΄ ευθείας και από την αρχική διαμέριση `[X, Q, J, S]` του δικτυωτού. Το ίδιο ισχύει και για τη διαμέριση `[X, Q, JS]`. Συνεπώς, και αυτές οι δυο διαμερίσεις, παρόλο που αντιστοιχούν σε επιτρεπτούς συνδυασμούς λέξεων-κλειδιών σύμφωνα με τη σημασιολογία συνεκτικότητας, μπορούν να παραλειφθούν από το δικτυωτό.

5.2.2 Ο αλγόριθμος CohesiveLCA

Ο αλγόριθμος CohesiveLCA δέχεται ως είσοδο μία ερώτηση με συνεκτικές σχέσεις λέξεων-κλειδιών και τις αντίστοιχες ανεστραμμένες λίστες, για να επιστρέψει όλους τους LCA που ικανοποιούν τις απαιτούμενες συνεκτικές σχέσεις, ταξινομημένους κατά σειρά συνεκτικού μεγέθους (βλ. Ενότητα 5.1.2).

Ο αλγόριθμος CohesiveLCA λειτουργεί ανάλογα με τον LCAsz αλλά με κάποιες βελτιώσεις και προσαρμογές. Οι αλλαγές αυτές έχουν να κάνουν, κατ΄ αρχάς, με τον ορισμό των συνεκτικών όρων και τη διαχείριση τους. Για παράδειγμα, στη γραμμή 25, η συνάρτηση pop πρέπει να πραγματοποιήσει κάποιους ελέγχους και να συνδυάσει σε ένα νέο όρο τους δύο όρους που δίδονται. Ο νέος όρος είναι ενήμερος των συνεκτικών σχέσεων που είναι ορισμένες εσωτερικά στου περιλαμβανόμενους όρους, κι έτσι για ένα υποσύνολο λέξεων κλειδιών μπορεί να παράγονται περισσότεροι όροι απ΄ ό,τι για τον LCAsz στον οποίο οι λέξεις-κλειδιά συνδυάζονται στο ίδιο επίπεδο και όχι εμφωλευμένα.

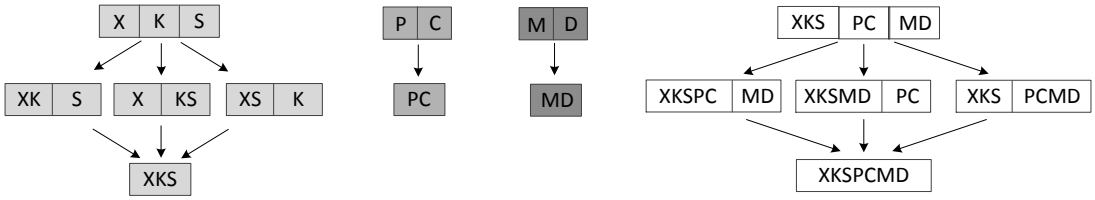
Η άλλη διαφορά είναι η δημιουργία του δικτυωτού, που υλοποιείται από τη συνάρτηση buildLattice() (γραμμή 2), η οποία ακολουθεί τη λογική που συζητήθηκε στην

προηγούμενη παράγραφο.

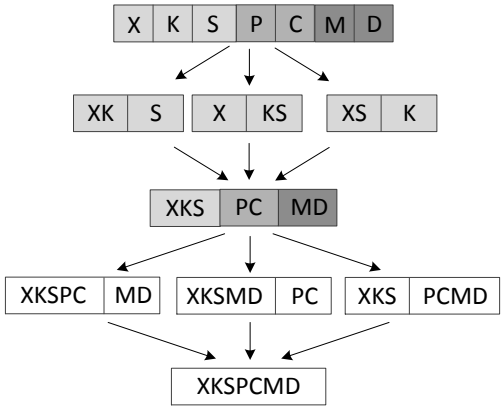
Κατασκευή του δικτυωτού. Το κύριο χαρακτηριστικό του αλγορίθμου CohesiveLCA είναι η μείωση της διάστασης του δικτυωτού, που προκαλείται από τον ορισμό των συνεκτικών σχέσεων. Η μείωση αυτή όπως συζητείται στην ανάλυση πολυπλοκότητας του αλγορίθμου και επιβεβαιώνεται από την πειραματική ανάλυση, έχει μια σημαντική επίδραση στην επίδοση του αλγορίθμου. Η συνάρτηση buildLattice() δεν

Algorithm 5: CohesiveLCA

```
1  CohesiveLCA(Q: cohesive keyword query, invL: inverted lists)
2      buildLattice()
3      while currentNode ← getNextNodeFromInvertedLists() do
4          curPLCA ← PartialLCA(currentNode.dewey, currentNode.kw, 0, null)
5          push(initStack, curPLCA)
6          for every coarsenessLevel cL do
7              while pl ← next partial LCA of cL do
8                  for every stack S of cL containing pl.term do
9                      push(S, pl)
10     emptyStacks()
11 push(S: stack, pl: partial LCA)
12     while S.dewey not ancestor of pl.node do
13         pop(S)
14     while S.dewey ≠ pl.node do
15         addEmptyRow(S)
16     replaceIfSmallerWith(S.topRow, pl.term, pl.size)
17 pop(S: stack)
18     p ← S.pop()
19     if S.columns = 1 then
20         addResult(S.dewey, p[0].size)
21     if S.columns > 1 then
22         for i←0 to S.columns do
23             for j←i to S.columns do
24                 if p[i] and p[j] contain sizes and p[i].provenance ∩
                   p[j].provenance = ∅ then
25                     t ← findTerm(p[i].term, p[j].term)
26                     sz ← p[i].size+p[j].size
27                     prv ← p[i].provenance ∪ p[j].provenance
28                     pLCA ← PartialLCA(S.dewey, t, sz, prv)
29     if S is not empty and S.columns > 1 then
30         for i=0 to S.columns do
31             if p[i].size+1 < S.topRow[i].size then
32                 S.topRow[i].size ← p[i].size+1
33                 S.topRow[i].provenance ← {lastStep(S.dewey)}
34     removeLastDeweyStep(S.dewey)
```



(α′) Δικτυωτά όρων: (i)(`XML Keyword Search`), (ii)(`Paul Cooper`), (iii)(`Mary Davis`) and (iv)(`(XML Keyword Search)(Paul Cooper)(Mary Davis)`)



(β′) Τελική μορφή δικτυωτού

Σχήμα 5.3: Σύνθεση δικτυωτών για την ερώτηση (`(XML Keyword Search) (Paul Cooper) (Mary Davis)`))

κατασκευάζει το πλήρες δικτυωτό, ώστε να το περικόψει στη συνέχεια, αλλά το δημιουργεί ως σύνθεση των δικτυωτών των συνεκτικών όρων που περιέχει μια ερώτηση. Αυτή η διαδικασία φαίνεται στην Εικόνα 5.3.

Ας θεωρήσουμε το δέντρο δεδομένων της Εικόνας 5.1 και την ερώτηση (`(XML Keyword Search) (Paul Cooper) (Mary Davis)`)) που τίθεται σ′ αυτό. Αν ο κάθε συνεκτικό όρος της ερώτηση αντιμετωπιστεί σαν μια ενιαία οντότητα, η αποτίμηση της ερώτησης χρειάζεται ένα δικτυωτό ενός συνόλου τριών στοιχείων. Αυτό είναι, για την ακρίβεια, το δικτυωτό της Εικόνας 5.3α′. Ωστόσο, η είσοδος του δικτυωτού δεν προέρχεται από μεμονωμένες λέξεις-κλειδιά, αλλά από συνδυασμούς τους. Ο καθένας απ′ αυτούς τους συνδυασμούς, ορίζει το δικό του δικτυωτό, όπως φαίνεται στο εριστερό μέρος της Εικόνας 5.3α′ (βλ. τα δικτυωτά (i), (ii) και (iii)). Το τελικό δικτυωτό που θα χρησιμοποιήσει ο CohesiveLCA κατασκευάζεται από σύνθεση των δικτυωτών (i), (ii) και (iii) με το δικτυωτό (iv) και φαίνεται στην Εικόνα 5.3β′. Παρατηρούμε πως αυτό είναι ένα δικτυωτό 9 μόλις διαμερίσεων, τη στιγμή που το πλήρες δικτυωτό 7 στοιχείων (μια και η εν λόγω ερώτηση αποτελείται από 7 λέξεις-κλειδιά) περιλαμβάνει 877 διαμερίσεις.

Η συνάρτηση buildLattice() κατασκευάζει το δικτυωτό για την αποτίμηση μια ερώτησης με συνεκτικές σχέσεις. Η συνάρτηση αυτή καλεί με τη σειρά της τη συνάρτηση buildComponentLattice() (γραμμή 8). Η συνάρτηση buildComponentLattice()

```
Function buildLattice
1  buildLattice(Q: query)
2      singletonTerms ← {keywords(Q)}
3      stacks.add(createSourceStack(singletonTerms)) constructControlSet(Q) for
       every control set cset in controlSets with not only singleton keywords do
4          stacks.add(createSourceStack(cset))
5      for every s in stacks do
6          buildComponentLattice(s)
7  constructControlSet(qp: query subpattern)
8      c ← new Set()
9      for every singleton keyword k in s do
10         c.add(k)
11     for every subpattern sqp in s do
12         subpatternTerm ← constructControlSet(sqp)
13         c.add(subpatternTerm)
14     controlSets.add(c)
15     return newTerm(c)
16 buildComponentLattice(s: stack)
17     for every pair t1, t2 of terms in s do
18         newS ← newStack(s, t1, t2) buildComponentLattice(newS)
```

(γραμμές 18-21) είναι μια αναδρομική συνάρτηση που δημιουργεί όλα τα δικτυωτά των συνεκτικών όρων, οι οποίοι μπορεί να είναι με οποιοδήποτε τρόπο εμφωλευμένοι ο ένας μέσα στον άλλον. Η διαδικασία δημιουργίας τους ελέγχεται από τη μεταβλητή *controlSets*, στην οποία αποθηκεύονται τα υποσύνολα λέξεων-κλειδιών που επιτρέπεται να συνδυάζονται από τις ορισμένες συνεκτικές σχέσεις. Η μεταβλητή αυτή ορίζεται στη συνάρτηση constructControlSet() (γραμμές 9-17).

5.2.3 Πολυπλοκότητα αλγορίθμου CohesiveLCA

Ο αλγόριθμος CohesiveLCA πεξεργάζεται τις ανεστραμμένες λίστες των λέξεων-κλειδιών μιας ερώτησης επωφελούμενος από τις ορισμένες συνεκτικές σχέσεις μεταξύ τους για να μειώσει το δικτυωτό των στοιβών που χρησιμοποιεί. Το μέγεθος του δικτυωτού ενός συνόλου k λέξεων-κλειδών εκφράζεται από τον αριθμό Βελλ του k, B_k, ο οποίος ορίζεται με την αναδρομική σχέση:

$$B_{n+1} = \sum_{i=0}^{n} \binom{n}{i} B_i,\ B_0 = B_1 = 1$$

Σε μία συνεκτική ερώτηση που περιέχει t συνεκτικούς όρους, ο αριθμός των δικτυωτών που συνθέτουν το τελικό είναι $t + 1$ λαμβάνοντας υπόψιν και το εξωτερικό όρο που αντιστοιχεί στην ίδια την ερώτηση. Το μέγεθος του δικτυωτού ενός όρου, που παίρνει μέρος στη δημιουργία του τελικού είναι, αν η πληθικότητά του είναι c_i, είναι B_{c_i}. Το στιγμιότυπο μιας λέξης-κλειδιού θα προκαλέσει στη χειρότερη περίπτωση μία ενημέρωση σε κάθε στοίβα καθενός από τα δικτυωτά των όρων στα οποία η συγκεκριμένη λέξη κλειδί συμμετέχει. Αν το μέγιστο βάθος εμφώλευσης σε μια ερώτηση είναι n και η

μέγιστη πληθικότητα συνεκτικού όρου στην ερώτηση είναι c, τότε ένα στογμιότυπο θα προκαλέσει $O(nB_c)$ ενημερώσεις στοιβών. Για ένα δέντρο δεδομένων με βάθος d, κάθε επεξεργασία ενός μερικού LCA από μια στοίβα περιλαμβάνει στη χειρότερη περίπτωση d ενέργειες pop και d ενέργειες *push*, δηλ. $O(d)$. Κάθε pop από μια στοίβα, που αντιστοιχεί σε μια διαμέριση με c στοιχεία, σημαίνει το πολύ $c(c-1)/2$ συνδυασμούς για παραγωγή μερικών LCA ανδ c ενημερώσεις μεγεθών στον προηγούμενο στοιχείο της στοίβας, δηλ. $O(c^2)$. Συνοψίζοντας, η πολυπλοκότητα χρόνου του αλγορίθμου CohesiveLCA δίνεται από τον τύπο:

$$O(dnc^2 B_c \sum_{i=1}^{c} |S_i|)$$

όπου S_i είναι η ανεστραμμένη λίστα της λέξης-κλειδιού i.

Η μέγιστη πληθικότητα συνεκτικού όρου μιας ερώτησης με δεδομένο αριθμό λέξεων-κλειδιών εξαρτάται από τον αριθμό όρων στην ερώτηση. Μια ερώτηση μπορεί να περιέχει το μεγαλύτερο αριθμό από συνεκτικούς όρους, αν όλοι οι όροι περιέχουν μία μόνο λέξη-κλειδί και έναν άλλο όρο, εκτός από τον πιο εμφωλευμένο συνεκτικό όρο, ο οποίος σύμφωνα με τη γλώσσα συνεκτικών λέξεων-κλειδιών, θα περιέχει αναγκαστικά δύο λέξεις-κλειδιά. Συνεπώς, η μέγιστη πληθικότητα συνεκτικού όρου μιας ερώτησης είναι $k-t-1$ με βάθος εμφώλευσης t. Συνεπώς, η πολυπλοκότητα του CohesiveLCA καταλήγει να είναι:

$$O(dt(k-t-1)^2 B_{k-t-1} \sum_{i=1}^{k} |S_i|)$$

Αυτή είναι μια παραμετροποιημένη πολυπλοκότητα που είναι γραμμική στο μέγεθος της εισόδου (δηλ. $\sum |S_i|$) του αλγορίθμου για σταθερό αριθμό λέξεων-κλειδιών.

5.3 Αποτίμηση ερωτήσεων με συνεκτικές σχέσεις λέξεων - κλειδιών

Ο αλγόριθμος CohesiveLCA υλοποιήθηκε σε Java και εξετάστικαν πειραματικά: α) η αποτελεσματικότητα της σημασιολογίας συνεκτκότητας λέξεων-κλειδιών και (β) η αποδοτικότητα του αλγορίθμου. Τα πειράματα διενεργήθηκαν σε ένα μηχάνημα 1.8 GHz dual core Intel Core i5 με λειτουργικό σύστημα Mac OS 10.8.

5.3.1 Σύνολα δεδομένων και ερωτήσεις

Για την πειραματική μελέτη χρησιμοποιήθηκαν τέσσερα πραγματικά σύνολα δεδομένων: τη βιβλιογραφική βάση δεδομένων DBLP[2], την αστρονομική βάση δεδομένων NASA[3], τη βιολογική βάση δεδομένων Protein Sequence Database (PSD) [4] και τη βάση στατιστικών αθλητικών δεδομένων Baseball [5]. Επιπλέον, χρησιμοποιήθηκε και η συνθετική

[2]http://www.informatik.uni-trier.de/ ley/db/

[3]http://www.cs.washington.edu/research/xmldatasets/www/repository.html

[4]http://pir.georgetown.edu/

[5]http://ibiblio.org/xml/books/biblegold/examples/baseball/

	DBLP	XMark	NASA	PSD	Baseball
size	1.15 GB	116.5 MB	25.1 MB	683 MB	1.1 MB
maximum depth	5	11	7	6	5
# nodes	34,141,216	2,048,193	530,528	22,596,465	26,432
# keywords	3,403,570	140,425	69,481	2,886,921	1984
# distinct labels	44	77	68	70	46
# dist. label paths	196	548	110	97	46

Πίνακας 5.1: Στατιστικά συνόλων δεδομένων DBLP, XMark, NASA, PSD και Baseball

συλλογή δεδομένων δημοπρασιών XMark[6]. Όλες αυτές οι συλλογές δεδομένων καλύπτουν διαφορετικές περιοχές εφαρμογών και επιδεικνύουν διαφορετικά χαρακτηριστικά. Ο Πίνακας 5.1 παρουσιάζει στατιστικά για καθεμιά απ΄ αυτές.

Η δενδρική βάση δεοδμένων DBLP είναι η μεγαλύτερη και η XMark η βαθύτερη. Για τα πειράματα αποτελεσματικότητας, χρησιμοποιήθηκαν τα δεδομένα των DBLP, PSD, NASA και Baseball. Για την αξιολόγηση της επίδοσης χρησιμοποιήθηκαν τα DBLP, XMark και NASA, ώστε να δοκιμαστεί ο αλγόριθμος CohesiveLCA σε δεδομένα με διαφορετικά χαρακτηριστικά μεγέθους και δομής. οι ανεστραμμένες λίστες των λέξεων-κλιδιών, μετά την ανάλυση των δέντρων των συνόλοων δεδομένων αποθηκεύτηκαν σε μία σχεσιακή βάση δεδομένων MySQL.

Επιλέχθηκαν πέντε συνεκτκές ερωτήσεις για κάθε ένα από τα πραγματικά σύνολα δεδομένων με διαισθητικό νόημα. Στις ερωτήσεις 3-6 λέξεων-κλειδιών έχουν εφαρμοστεί διάφορα πρότυπα συνεκτικών σχέσεων. Οι ερωτήσεις παρουσιάζονται στον Πίνακα 5.2. Πέντε εξειδικευμένοι χρήστες βαθμολόγησαν τις απαντήσεις τόσο δυαδικά (δηλ. σωστό/λάθος), όσο και σε κλίμακα 0-4, με το 0 να υποδηλώνει ότι ένα αποτέλεσμα κρίθηκε άσχετο με την απάντηση. Για παροχή βοήθειας στους χρήστες που βαθμολόγησαν, ώστε να μη χρειαστεί να εξετάσουν το πλήρες σύνολο αποτελεσμάτων κάθε φορά που μπορεί να φτάσει σε μεγάλο μέγεθος, τους δόθηκαν τα δενδρικά πρότυπα των απαντήσεων. Έτσι, τα αποτελέσματα ομαδοποιήθηκαν με τρόπο που καταδεικνύει τόσο τον τρόπο που οι λέξεις-κλειδιά συνδέονται ακριβώς μεταξύ τους σε κάθε απάντηση, όσο και ο LCA με τη ρίζα του δενδρικού συνόλου δεδομένων. Κατόπιν, αποδόθηκε σε κάθε LCA η μέγιστη βαθμολογία των δενδρικών προτύπων για τα οποία ήταν ρίζα.

5.3.2 Αποτελεσματικότητα σημασιολογίας συνεκτικότητας

Στο 1ο μέρος της πειραματικής μελέτης, αξιολογήθηκε η αποτελεσματικότητα της προσέγγισης των ερωτήσεων με συνεκτικές σχέσεις μεταξύ των λέξεων-κλειδιών. Η προσέγγιση των συνεκτικών ερωτήσεων αξιολογήθηκε τόσο ως μηχανισμός ταξινόμησης όσο και ως μηχανισμός φιλτραρίσματος σε σχέση με παλαιότερες προσεγγίσεις.

Μηχανισμός φιλτραρίσματος συνεκτικής σημασιολογίας. Η προσέγγιση ερωτήσεων συνεκτικών λέξεων-κλειδιών συγκρίθηκε με τις σημασιολογίες φιλτραρίσματος των ελαχίστων LCA (SLCA) [22, 51, 48, 11], των αποκλειστικών LCA (ELCA) [19, 52, 57], των πολύτιμων LCA (VLCA) [13, 26] και των σημαντκών LCA

[6] http://www.xml-benchmark.org

DBLP	
Q_1^D	(proof (Scott theorem))
Q_2^D	((IEEE transactions communications) (wireless networks))
Q_3^D	((Lei Chen) (Yi Guo))
Q_4^D	((Wei Wang) (Yi Chen))
Q_5^D	((VLDB journal) (spatial databases))
PSD	
Q_1^P	((african snail) mRNA)
Q_2^P	((alpha 1) (isoform 3))
Q_3^P	((penton protein) (human adenovirus 5))
Q_4^P	(((B cell) stimulating factor) (house mouse))
Q_5^P	((spectrin gene) (alpha 1))
NASA	
Q_1^N	((ccd photometric system) magnitudes)
Q_2^N	((stars types) (spectral classification))
Q_3^N	((Astronomical (Data Center)) (Wilson luminosity codes))
Q_4^N	((year 1968) (Zwicky Abell clusters))
Q_5^N	((title Orion Nebula) (author Parenago))
Baseball	
Q_1^B	(Matt Williams (third base))
Q_2^B	(team (Johnson (first base)) (Wilson pitcher))
Q_3^B	(player surname (0 errors))
Q_4^B	(player (relief pitcher) (0 losses))
Q_5^B	(player (0 errors) (7 games))

Πίνακας 5.2: Ερωτήσεις για τα πειράματα αποτελεσματικότητας στα πραγματικά σύνολα δεδομένων

(MLCA) [31], που συζητήσθηκαν στο Κεφάλαιο ση:κως. Αυτές είναι οι κύριες σημασιολογίες φιλτραρίσματος που προτείνονται στη βιβλιογραφία. Συνοπτικά κατά τη σημασιολογία SLCA ένας LCA ειναι έγκυρος αν δεν υπάρχει άλλος στο υποδέντρο του. Χαλαρώνοντας τη σημασιολογία SLCA, κατή την ELCA σημασιολγία ένας LCA είναι έγκυρος ακόμα κι αν έχει άλλους LCA στο υποδέντρο του, αρκεί αυτός να είναι LCA για στιγμιότυπα των λέξεων-κλειδιών που δεν περιλαμβάνονται στα υποδέντρα των άλλων. Σύμφωνα με τη σημασιολογία VLCA, ένας LCA είναι έγκυρος αν δεν περιέχει στο υποδέντρο του κάποια ετικέτα δύο φορές, εκτός και πρόκειτα για φύλλα του υποδέντρου. Η σημασιολογία MLCA, τέλος, απαιτεί για οποιουσδήποτε κόμβους n_a και n_b με ετικέτες a και b αντίστοιχα, σε ένα ελάχιστο συνδετικό δέντρο στιγμιοτύπων λέξεων-κλειδιών, να μην υπάρχει κόμβος n'_b με ετικέτα b, ο οποίος να είναι πιο στενά συνδεδομένος με τον κόμβο n_a (δηλ. ο $lca(n_a, n'_b)$ να είναι απόγονος του $lca(n_a, n_b)$). Οι σημασιολογίες SLCA και ELCA βασίζονται αμιγώς σε δομικά χαρακτηριστικά, ενώ οι VLCA και MLCA λαμβάνουν υπόψιν και τις ετικέτες των κόμβων ενός δέντρου.

Ο Πίνακας 5.3 δείχνει τον αριθμό αποτελεσμάτων για κάθε ερώτηση που τίθεται

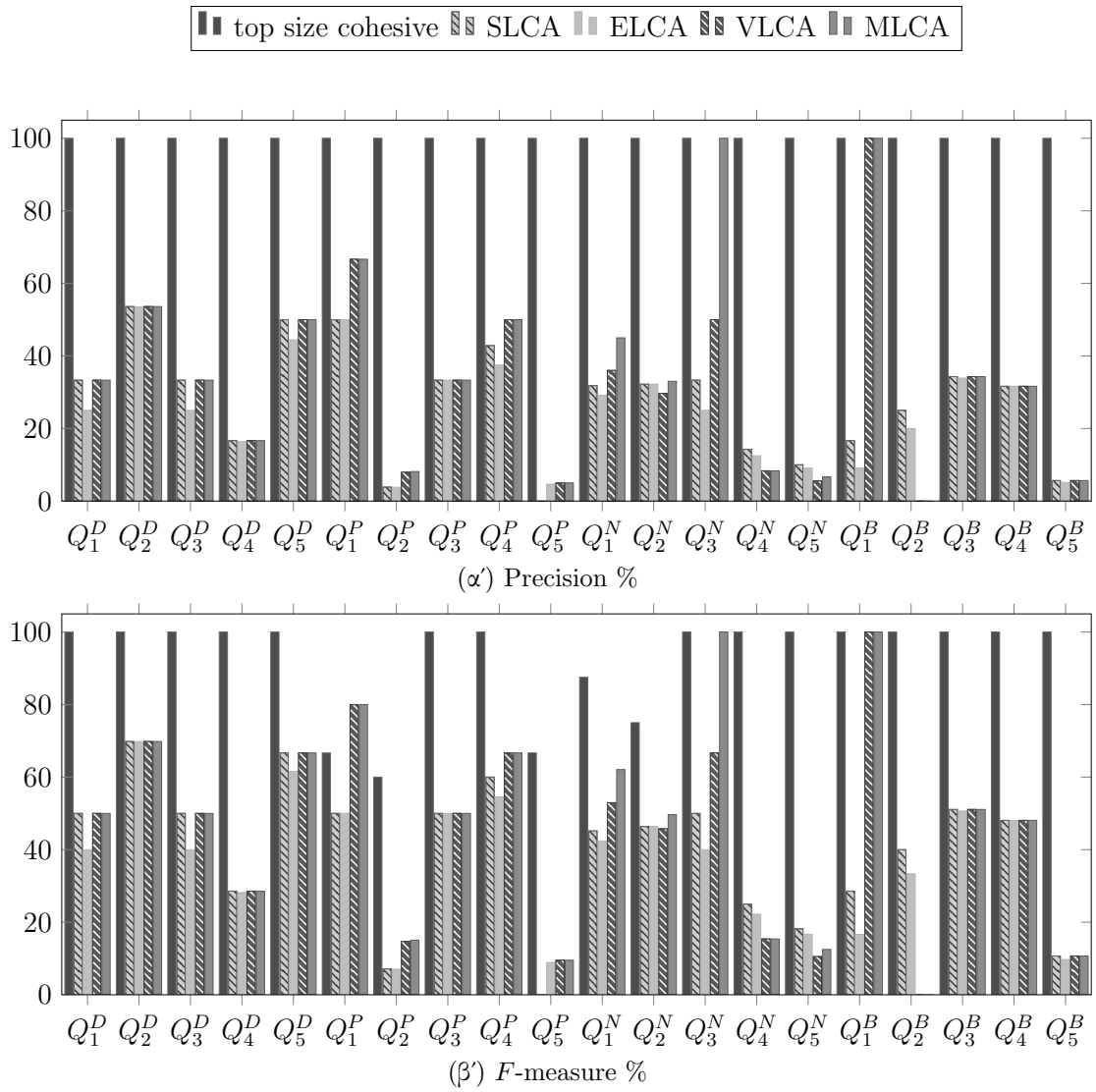
στα σύνολα δεδομένων DBLP, PSD, NASA και Baseball. Αξίζει να σημειωθεί, πως με εξαίρεση τις σημασιολογίες SLCA και ELCA που δίνουν σύνολα αποτελεσμάτων που σχετίζονται μεταξύ τους (SLCA $\subseteq$ ELCA), οι υπόλοιπες προσεγγίσεις δεν είναι συγκρίσιμες μεταξύ τους. Αυτό σημαίνει, πως η μια μπορεί να επιστρέψει αποτελέσματα που η άλλη απορρίπτει και αντίστροφα. Η προσέγγιση CohesiveLCA επιστρέφει όλα τα αποτελέσματα, αρκεί να ικανοποιούν τις συνεκτικές σχέσεις που ορίζονται μεταξύ των λέξεων-κλειδιών μιας ερώτησης. Εφόσον, οι σχέσεις αυτές ορίζονται από τον ίδιο το χρήστη, οποιοδήποτε επιπλέον αποτέλεσμα δεν τις ικανοποιεί, το οποίο μπορεί να επιστρέφεται από άλλες προσεγγίσεις, είναι εξ ορισμού άσχετο με το σκοπό της ερώτησης. Για παράδειγμα, για την ερώτηση Q_5^P, μόνο 3 αποτελέσματα ικανοποιούν τις συνεκτικές σχέσεις που έχει ορίσει ο χρήστης, οπότε οι προσεγγίσεις SLCA, VLCA, MLCA επιστρέφουν τουλάχιστον 37 και η ELCA τουλάχιστον 40 άσχετα αποτελέσματα.

Η σημασιολογία CohesiveLCA ταξινομεί όλα τα αποτελέσματα μιας ερώτησης κλιμακωτά με βάση το μέγεθος LCA τους. Για να γίνει εφικτή η σύγκριση με τις υπόλοιπες σημσιολογίες που είναι εγγενώς σημσιολογίες φιλτραρίσματος, περιορίζονται και τα αποτελέσματα των ερωτήσεων με συνεκτικές σχέσεις λέξεων-κλειδιών στο κορυφαίο επίπεδο (δηλ. τα αποτελέσματα μικρότερου μεγέθους LCA ή top-1-size). Η

dataset	query	# of results				
		CohesiveLCA	SLCA	ELCA	VLCA	MLCA
DBLP	Q_1^D	2	3	4	3	3
	Q_2^D	527	981	982	981	981
	Q_3^D	2	3	4	3	3
	Q_4^D	11	60	61	60	60
	Q_5^D	5	8	9	8	8
PSD	Q_1^P	3	2	3	3	3
	Q_2^P	14	78	79	88	85
	Q_3^P	2	4	4	3	3
	Q_4^P	4	7	8	6	6
	Q_5^P	3	40	43	40	40
NASA	Q_1^N	17	22	24	25	20
	Q_2^N	85	90	90	118	106
	Q_3^N	1	3	4	2	1
	Q_4^N	6	7	8	12	12
	Q_5^N	9	10	11	18	15
Baseball	Q_1^B	10	5	6	1	1
	Q_2^B	7	4	5	0	0
	Q_3^B	216	516	522	516	516
	Q_4^B	145	335	335	335	335
	Q_5^B	49	177	196	177	177

Πίνακας 5.3: Αριθμός αποτελεσμάτων των ερωτήσεων στα διαφορετικά σύνολα δεδομένων

σύγκριση βασίζεται στα ευρέως αποδεκτά μεγέθη *precision (P)*, *recall (R)* και $\mathcal{F}$-measure= $\frac{2P \times R}{P+R}$ [4]. Η Εικόνα 5.4 απεικονίζει τα αποτελέσματα για τις πέντε σημασιολογίες στα τέσσερα σύνολα δεδομένων.



(α′) Precision %

(β′) F-measure %

Σχήμα 5.4: Ακρίβεια και $\mathcal{F}$-measure των σημασιλογιών φιλτραρίσματος Cohesive LCA κορυφαίου μεγέθους, SLCA και ELCA

Το διάγραμμα της Εικόνας 5.4α′ καταδεικνύει ότι η προσέγγιση CohesiveLCA ξεπερνά με άνεση τις άλλες προσεγγίσεις. Το φιλτράρισμα 1ου επιπέδου (top-1-size) επιτυγχάνει τέλεια ακρίβεια για όλες τις ερωτήσεις σε όλες τις βάσεις δεδομένων. Αυτό δεν είναι έκπληξη, καθώς η συμασιολογία CohesiveLCA εκμεταλλεύεται την επιπλέον πληροφορία των συνεκτικών σχέσεων για να απορρίψει άσχετα αποτελέσματα. Επίσης, επιτυγχάνει τέλειες τιμές $\mathcal{F}$-measure στα σύνολα δεδομένων DBLP και Baseball. Παρ′ όλ′ αυτά, η τιμή $mathcal F$-measure για τα PSD και NASA εμφανίζεται χαμηλότερη. Αυτό οφείλεται στο εξής φαινόμενο: αντίθετα με τα DBLP και Baseball, τα PSD και NASA είναι βαθειά και σύνθετα δομικά δέντρα δεδομένων με αρκετό κείμενο να φιλξενείται συχνά στους κόμβους τους. Αυτή η πολυπλοκότητα οδηγεί σε αποτελέσματα που κυμαίνονται σε πολλά επίπεδα μεγεθών στις περισσότερες ερωτήσεις. Κάποια από τα σχετικά αποτελέσματα συμβαίνει, λοιπόν, να μην έχουν το ελάχιστο παρατηρούμενο

μέγεθος LCA και γι΄ αυτό απορρίπτονται από το top-1-size φιλτράρισμα της σημασιολογίας συνεκτικότητας. Ωστόσο, κανένα σωστό αποτέλεσμα δε χάνεται τη σημασιολογία συνεκτικότητας αν δεν επιβληθεί κάποιο φιλτράρισμα μεγέθους, όπως φαίνεται στον Πίνακα 5.4.

Ο Πίνακας 5.4 παρουσιάζει συνοπτικά τις μετρήσεις ακρίβειας, πληρότητας και $\mathcal{F}$-measure για όλες τις ερωτήσες σε όλα τα σύνολα δεδομένων. Ο πίνακας δείχνει τα αποτελέσματα για τις πέντε σημασιολογίες φιλτραρίσματος αλλά και για τη σημασιολογία συνεκτκότητας χωρίς φιλτράρισμα, δηλ. χωρίς περιορισμό στο μέγεθος των αποτελεσμάτων. Η σημασιολογία συνεκτικότητας τόσο η απλή (CohesiveLCA) όσο και αυτή με φιλτράρισμα (op-1-size CohesiveLCA) έχουν πολύ καλύτερες επιδόσεις και στις στρεις μετρικές. Η προσέγγιση top-1-size CohesiveLCA επιδεικνύει τέλεια ακρίβεια, ενώ η CohesiveLCA με ελαφρώς χαμηλότερη ακρίβεια εξασφαλίζει απόλυτη πληρότητα στα αποτελέσματα. Αυτά τα εντυπωσιακά χαρακτηριστικά οφείλονται καθαρά στις συνεκτικές σχέσεις μεταξύ λέξεων-κλειδιών, που όπως αποδεικνύεται είναι ένα δυνατό εργαλείο στα χέρια του χρήστη που μπορεί εύκολα και διαισθητικά να χρησιμοποιήσει για να βελτιώσει την επιτυχία της αναζήτησής του.

Μηχανισμός ταξινόμησης συνεκτκής σημασιολογίας. Το μοντέλο ταξινόμησης που προτάθηκε στην Ενότητα 5.1.2 αξιολογήθηκε με τον υπολογισμό των μεγεθών Mean Average Precision (MAP) [4] και Normalized Discounted Cumulative Gain (NDCG) [4] στις ερωτήσεις του Πίνακα 5.2. Το μέγεθος MAP είναι η μέση τιμή των επιμέρους τιμών ακρίβειας ενός ταξινομημένου συνόλου αποτελεσμάτων κάθε φορά που ένα σωστό αποτέλεσμα προστίθεται στην απάντηση μιας ερώτησης. Αν ένα σωστό αποτέλεσμα δεν επιστραφεί, η συνεισφορά του ισούται με 0. Το μέγεθος MAP τιμωρεί επιβαρύνει την αξιολόγηση ενός αλγορίθμου όταν σωστά αποτελέσματα δεν επιστρέφονται καθόλου ή όταν λάθος αποτελέσματα επιστρέφονται ψηλά στην ταξινόμηση. Δεδομένης μιας θέσης στην ταξινόμηση, το μέγεθος Discounted Cumulative Gain (DCG) ορίζεται ως το άθροισμα των βαθμολογιών των αποτελεσμάτων μιας απάντηση έως αυτήν τη θέση, διαιρεμένο με το λογάριθμο της θέσης αυτής. Το διάνυσμα DCG μιας απάντησης είναι το διάνυσμα των επιμέρους DCG τιμών των αποτελεσμάτων, με την κάθε τιμή να καταλαμβάνει στο διάνυσμα τη θέση του στην κατάταξη. Στη συνέχεια, το διάνυσμα NDCG παράγεται από την κανονικοποίηση του διανύσματος DCG με το διάνυσμα της ιδεατής, τέλειας κατάταξης (δηλ. αυτή που ακολουθεί την ταξινόμηση που ορίζουν οι βαθμολογίες των ειδικών). Το NDCG επιβαρύνει έναν αλγόριθμο όταν ευνοεί αποτελέσματα με χαμηλές βαθμολογίες έναντι αυτών με υψηλές στην ταξινόμηση.

Ο Πίνακας 5.5 καταγράφει τις τιμές των μεγεθών MAP και NDCG για τη συνεκτική ταξινόμηση των αποτελεσμάτων για τις ερωτήσεις του Πίνακα 3.9. Οι υψηλές τιμές του NDCG αποδεικνύουν ότι η ταξινόμηση των αποτελεσμάτων σε αύξουσα σει-

	CohesiveLCA	top-1-size CohesiveLCA	SLCA	ELCA	VLCA	MLCA
Precision %	67.4	100	25.1	27.6	32.6	35.7
Recall %	100	96.9	88.0	93.0	95.0	95.0
$\mathcal{F}$-measure %	76.8	92.8	39.8	36.8	44.4	46.8

Πίνακας 5.4: Μέση ακρίβεια, πληρότητα και $\mathcal{F}$-measure για όλες τις ερωτήσεις με βάση τις διαφορετικές σημασιολογίες

MAP (%)			
DBLP	PSD	NASA	Baseball
94	99	94	97

NDCG (%)			
DBLP	PSD	NASA	Baseball
100	99	98	100

Πίνακας 5.5: Τιμές MAP και NDCG για τα πργματικά σύνολα δεδομένων για τις ερωτήσεις του Πίνακα 5.2

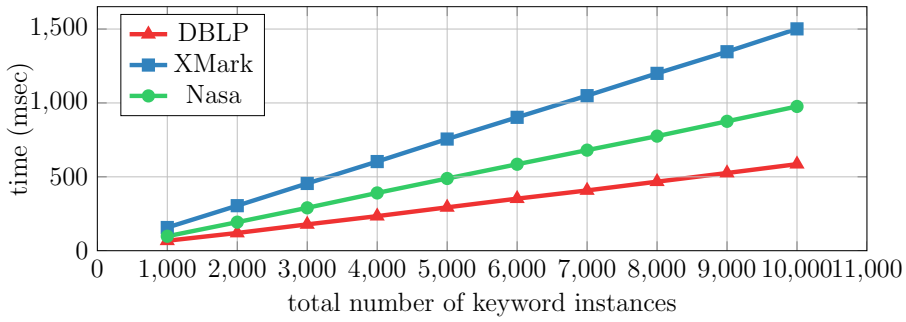
ρά μεγέθους, που λαμβάνει υπόψιν και τα μεγέθη των συνεκτικών όρων, είναι πολύ κοντά στην ταξινόμηση αναφοράς που προκύπτει από τη βαθμολόγηση των ειδικών. Οι περισσότερες τιμές για το μέγεθος MAP είναι ελαφρώς χαμηλότερες του 100%. Αυτό σημαίνει πως ένας μικρός αριθμός από μη σχετικούς LCA με την ερώτηση κατατάσσονται ψηλότερα από κάποιους σωστούς. Παρ΄ όλ΄ αυτά οι πολύ υψηλές τιμές για το NDCG, υποδεικνύουν ότι ακόμα κι αυτά τα μη επιθυμητά αποτελέσματα δε βρίσκονται ψηλά στη γενική κατάταξη, το οποίο σημαίνει ότι ο χρήστης μπορεί να μη βρεθεί ποτέ στη θέση να τα εξετάσει.

5.3.3 Επίδοση αλγορίθμου CohesiveLCA

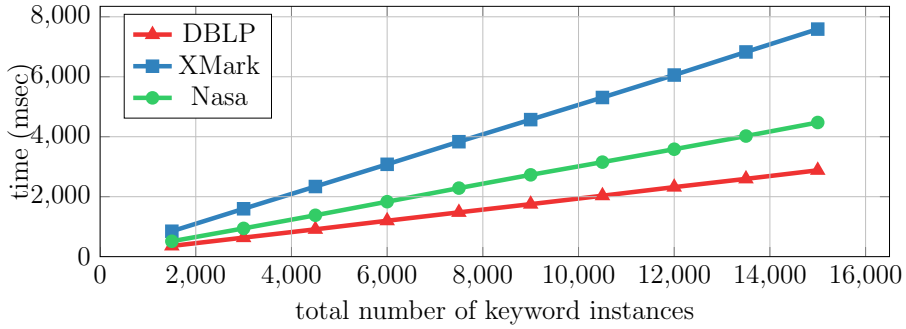
Η αποδοτικότητα του αλγορίθμου CohesiveLCA εξετάστηκε μελετώντας: α) την επίδοσή του καθώς αυξάνεται το μέγεθος του συνόλου δεδομένων, β) την επίδοσή του καθώς αυτξάνεται η πληθικότητα των συνεκτικών όρων που περιέχει η ερώτηση και γ) τη βελτίωση στην αποδοτικότητά του σε σχέση με προηγούμενες προσεγγίσεις. Χρησιμοποιήθηκαν ομάδες ερωτήσεων με 10, 15 και 20 λέξεις-κλειδιά που τέθηκαν στα σύνολα δεδομένων DBLP, XMark και NASA.

Για κάθε μέγεθος ερώτησης, σχηματίσαμε δέκα διαφορετικά πρότυπα συνεκτικών σχέσεων. Κάθε τέτοιο πρότυπο ορίζει διαφορετικό αριθμό από συνεκτικούς όρους, με διάφορες πληθυκότητες και ποικίλους βαθμούς εμφώλευσης. Για παράδειγμα, ένα πρότυπο συνεκτικών σχέσεων για μια οποιαδήποτε ερώτηση 10 λέξεων-κλειδιών είναι το `(xx((xxxx)(xxxx)))`, το οποίο μπορεί να εφαρμοστεί ως μάσκα σε οποιαδήποτε ερώτηση. Αυτά τα πρότυπα εφαρμόστηκαν στη συλλογή των ερωτήσεων πολλαπλασιάζοντας τον αριθμό των διακριτών ερωτήσεων που εκτελέστηκαν. Οι λέξεις-κλειδιά επιλέχθηκαν τυχαία ανάμεσα στις πιο συχνές σε κάθε σύνολο δεδομένων, ώστε να δοκιμαστεί ο αλγόριθμος σε συνθήκες αποτίμησης μεγάλου αριθμού στιγμιοτύπων και παραγωγής πολλών αποτελεσμάτων. Συγκεκριμένα, για κάθε πρότυπο, ορίστηκαν 10 διαφορετικές ερωτήσεις και υπολογίστηκε ο μέσος χρόνος εκτέλεσής τους. Συνολικά, δοκμιάστηκαν 100 ερωτήσεις ανά σύνολο δεδομένων. Για κάθε ερώτηση, διενεργήθηκαν πειράματα αλλάζοντας το μέγεθος της εισόδου, δηλ. των ανεστραμμένων λιστών των λέξεων-κλειδιών από 100 έως 1000 στιγμιότυπα ανά λέξη-κλειδί με βήμα 100 στιγμιότυπα.

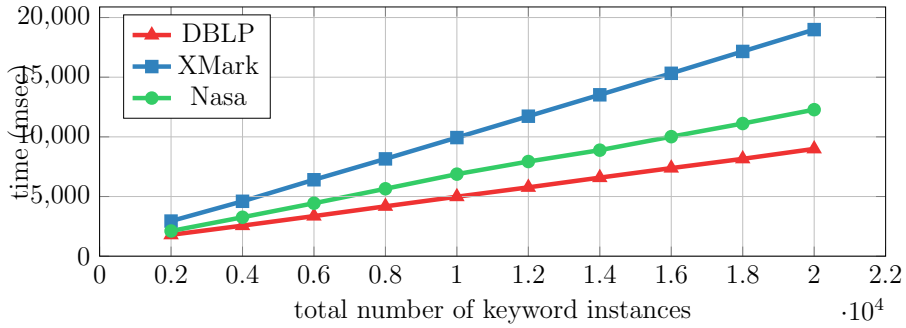
Αποδοτικότητα αλγορίθμου Η Εικόνα 5.5 δείχνει πώς κλιμακώνεται ο χρόνος εκτέλεσης του CohesiveLCA όταν το συνολικό μήκος των ανεστραμμένων λιστών των λέξεων-κλειδιών μεγαλώνει. Κάθε διάγραμμα αντιστοιχεί σε ένα διαφορετικό μέγεθος ερώτησης (10, 15 ή 20 λέξεις-κλειδιά) και παρουσιάζει την επίδοση του CohesiveLCA σε όλα τα σύνολα δεδομένων. Κάθε καμπύλη αντιστοιχεί σε διαφορετικό σύνολο δε-



(α′) ερωτήσεις 10 λέξεων-κλειδιών



(β′) ερωτήσεις 15 λέξεων-κειδιών



(γ′) ερωτήσεις 20 λέξεων-κλειδιών

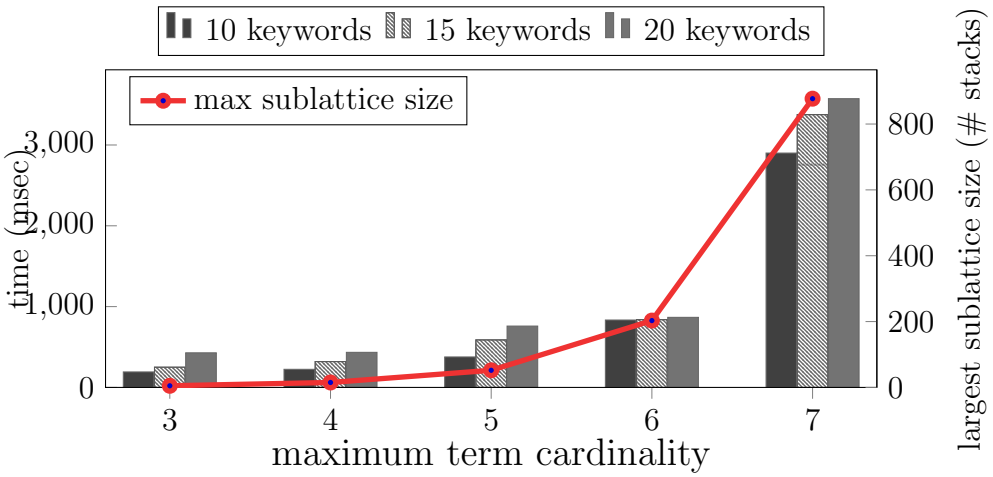
Σχήμα 5.5: Επίδοση CohesiveLCA για ερωτήσεις 10, 15 και 20 λέξεων-κλειδιών με μεταβλητό αριθμό στιγμιοτύπων

δομένων και κάθε σημείο της καμπύλης αναπαριστά το μέσο χρόνο εκτέλεσης των 100 ερωτήσεων που ακολουθούν τα 10 διαφορετικά πρότυπα συνεκτικών σχέσεων για το δεδομένο μήκος ερώτησης. Δεδομένου ότι οι λέξεις-κλειδιά είναι τυχαία επιλεγμένες ανάμεσα στο συχνότερες των συλλογών δεδομένων, οι καμπύλες απεικονίζουν την κλιμάκωση της επίδοσης του αλγορίθμου σε σχέση με το μέγεθος της αντίστοιχης βάσης δεδομένων.

Όλεοι καμπύλες δείχνουν καθαρά τη γραμμική συμπεριφορά του αλγορίθμου CohesiceLCAsz σε σχέση με το μέγεθος της ειδόσου. Για την ακρίβεια, αυτή η συμπεριφορά ακολουθείται και για καθεμιά από τις 100 ερωτήσεις που συνεισφέρουν στη μέση τιμή μιας καμπύλης. Σε όλες τις περιπτώσεις, οι χρόνοι αποτίμησης των ερωτήσεων επιβεβαιώνουν την εξάρτηση της επίδοσης του αλγορίθμου από το μέγιστο βάθος του συνόλου δεδομένων: η αποτίμηση ερωτήσεων στο DBLP (μέγιστο βάθος 5) είναι πάντα ταχύτερη απ ό,τι στο NASA (μέγιστο βάθος 7), και στο NASA η αποτίμηση είναι γρηγορότερη απ΄ ό,τι στο XMark (μέγιστο βάθος 11).

Είναι αξισημείωτο το γεγονός ότι ο αλγόριθμος CohesiveLCA επιτρέπει την αποτίμηση ερωτήσεων σε πρακτικό χρόνο, ακόμα και για πολλές λέξεις-κλειδιά σε μεγάλα και σύνθετα σύνολα δεδομένων. Για παράδειγμα, μια ερώτηση με 20 λέξεις-κλειδιά και 20.000 στιγμιότυπα χρειάζεται 20σες για να υπολογιστεί στο XMark. Σ′ αυτήν την παρατήρηση, πρέπει κανείς να λάβει επίσης υπόψιν ότι αυτοί οι χρόνοι επιτυγχάνονται από ένα πρωτότυπο σύστημα χωρίς τις προσαρμογές και βελτιστοποιήσεις ενός εμπορικού συστήματος αναζήτησης. Στο πλαίσιο της μελέτης της σχετικής βοβλιογραφίας, δεν υπάρχει άλλη προσέγγιση που εκτελεί ερωτήσεις με τόσο μεγάλους αριθμούς λέξεων-κλειδιών.

Πληθικότητα συνεκτικών όρων και επίδοση. Όπως φάνηκε στην ανάλυση του αλγορίθμου CohesiveLCA στην Ενότητα 5.2.3 ο καθοριστικός παράγων στην επίδοση του αλγορίθμου είναι η μέγιστη πληθικότητα συνεκτικού όρου σε μια ερώτηση. Ο αριθμός αυτός καθορίζει το μέγεθος του μεγαλύτερου δικτυωτού συνεκτικού όρου που τελικά αποτελεί μέρος του δικτυωτού του αλγορίθμου (Εικόνα 5.3). Αυτή η εξάρτηση επιβεβαιώνεται στο διάγραμμα της Εικόνας 5.6. Χρησιμοποιήθηκαν ερωτήσεις 10, 15 και 20 λέξεων-κλειδιών με συνολικό αριθμό 6000 στιγμιοτύπων, οι οποίες εκτελέστηκαν στο DBLP. Ο άξονας των x αναπαριστά τη μέγιστη πληθικότητα των συνεκτικών όρων μιας ερώτησης. Ο χρόνος εκτέλεσης που απεικονίζεται από τις μπάρες (αριστερός άξονας των y) είναι ο μέσος όρος των ερωτήσεων του αντίστοιχου μήκους ερώτησης, για τις οποίες ο μέγιστος συνεκτικός όρος έχει τη συγκεκριμένη πληθικότητα του άξονα x. Η καμπύλη αναπαριστά το μέγεθος του δικτυωτού μιας ερώτησης με τόσες λέξεις-κλειδιά όση και η πληθικότητα του άξονα των x. Το μέγεθος του δικτυωτού προσδιορίζεται από τον αριθμό των στοιβών που περιέχει.



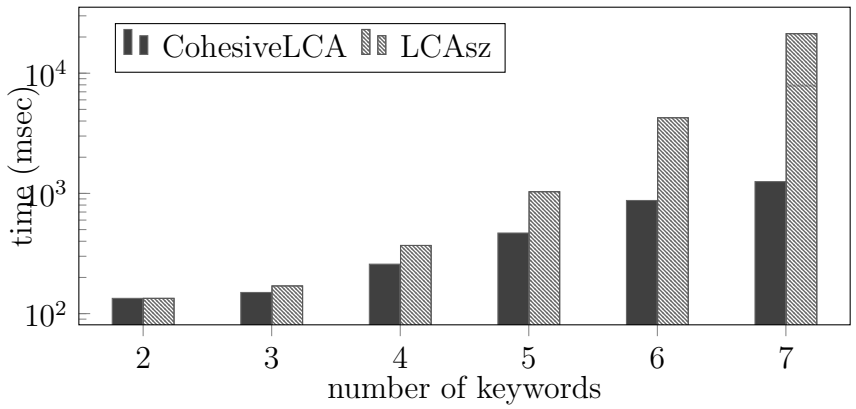
Σχήμα 5.6: Επίδοση CohesiveLCA για ερωτήσεις 6000 στιγμιοτύπων λέξεων-κλειδιών και διαφορετικές πληθυκότητες συνεκτικών όρων στο σύνολο δεδομένων DBLP

Είναι ενδιαφέρουσα η παρατήρηση ότι ο χρόνος εκτέλεσης εξαρτάται πρωτίστως από τη μέγιστη πληθικότητα συνεκτικού όρου μιας ερώτησης και πολύ λιγότερο από το συνολικό αριθμό λέξεων-κλειδιών. Για παράδειγμα, μια ερώτηση 20 λέξεων-κλειδιών με μέγιστη πληθικότητα συνεκτικού όρου 6, αποτιμάται πολύ γρηγορότερα από μια ερώτηση 10 λέξεων-κλειδιών, αλλά με μέγιστη πληθικότητα συνεκτικού όρου 7. Αυτή η παρατήρηση, υποδεικνύει ότι όσο το μέγιστο μήκος ενός συνεκτικού όρου παραμένει σε χαμηλά επίπεδα, ο αλγόριθμος CohesiveLCA μπορεί πρακτικά να αποτιμήσει ερωτήσεις

με πολύ μεγάλο συνολικό αριθμό λέξεων-κλειδιών.

Βελτίωση επίδοση λόγω συνεκτικών σχέσεων. Σ΄ αυτήν την παράγραφο ο αλγόριθμος CohesiveLCA συγκρίνεται με αντίστοιχους αλγορίθμους ως προς την επίδοση, οι οποίοι υπολογίζουν το σύνολο των LCA για μια ερώτηση σε δενδρικές δομές δεδομένων και τους ταξινομούν με βάση το μέγεθος. Δεν έχει νόημα η σύγκριση με άλλους αλγορίθμους καθώς υπολογίζουν ασύγκριτα σύνολα LCA ως αποτελέσματα (δηλ. δεν περιέχονται τα αποτελέσματα της μιας προσέγγισης από το σύνολο αποτελεσμάτων της άλλης), ενώ επιπλέον δεν παρέχουν εγγενή τρόπο ταξινόμησης των αποτελεσμάτων. Στα πειράματα οι ερωτήσεις εκτελέστηκαν στο σύνολο δεδομένων DBLP. Τα αποτελέσματα στα άλλα σύνολα δεδομένων είναι αντίστοιχα.

Δύο αλγόριθμοι μπορούν να υπολογίσουν το πλήρες σύνολο των LCA με τα μεγέθη τους: ο αλγόριθμος LCAsz (βλ. 3.3.1) και ο αλγόριθμος SAOne [21]. Ο αλγόριθμος SAOne είναι μια πιο αποδοτική παραλλαγή του αλγορίθμου SA που προτείνεται στην ίδια δουλειά. Ο SA υπολογίζει όλους τους LCA μαζί με μια σύνοψη των υποδέντρων (GDMCT) των στιγμιοτύπων των λέξεων-κλειδιών που είναι απόγονοί τους. Η σύνοψη αυτή επιτρέπει τον υπολογισμό μεγεθών. Η βελτίωση του SAOne σε σχέση με την επίδοση του SA έγκειται στο γεγονός ότι πρώτος υπολογίζει μόνο LCA χωρίς να απαιτείται ο υπολογισμός όλων των δυνατών GDMCT.



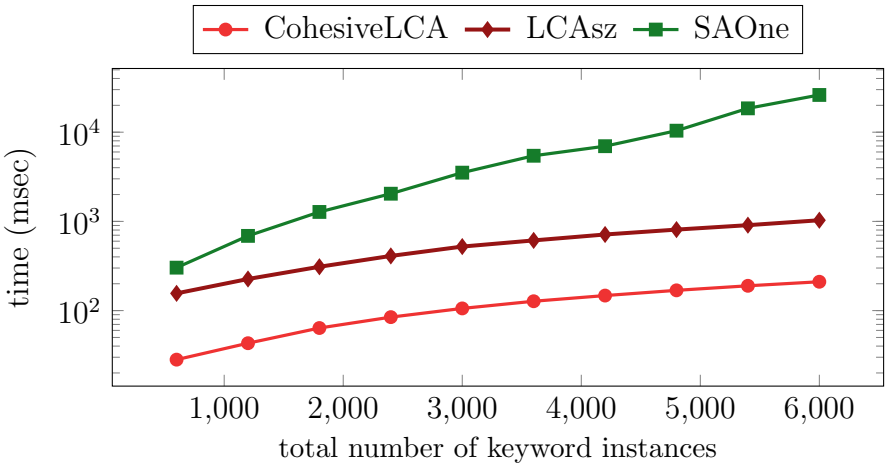
Σχήμα 5.7: Βελτίωση CohesiveLCA έναντι LCAsz για ερωτήσεις διαφορετικών αριθμών λέξεων-κλειδιών

Στην Εικόνα 5.7 συγκρίνονται οι χρόνοι εκτέλεσης του LCAsz και του CohesiveLCA στην αποτίμηση ερωτήσεων στο DBLP για διαφορετικούς αριθμούς λέξεων-κλειδιών. Οι χρόνοι εκτέλεσης του LCAsz είναι ο μέσος όρος από 10 τυχαίες ερωτήσεις αποτελούμενες από συχνές λέξεις-κλειδιά. Ποικίλα πρότυπα συνεκτικών σχέσεων ορίστηκαν για τον CohesiveLCA (ο αριθμός τους εξαρτάται από το συνολικό αριθμό λέξεων-κλειδιών μιας ερώτησης), και για καθένα απ΄ αυτά δημιουργήθηκαν 10 ερωτήσεις με τυχαία επιλογή από τις συχνότερες λέξεις-κλειδιά. Οι χρόνοι εκτέλεσης του CohesiveLCA προκύπτουν από το μέσο όρο εκτέλεσης όλων των ερωτήσεων με ίδιο αριθμό λέξεων-κλειδιών. Σε όλες τις περιπτώσεις, λήφθηκαν 1000 στιγμιότυπα ανά λέξη-κλειδί μέσω περικοπής των ανεστραμμένων λιστών των συχνών λέξεων-κλειδιών.

Όπως είναι φανερό στην Εικόνα 5.7, ο CohesiveLCA ξεπερνά τον LCAsz. Η βελτίωση στην επίδοση αγγίζει τη μία τάξη μεγέθους για 6 λέξεις-κλειδιά και αυξάνεται ακόμα περισσότερο για 7. Επιπροσθέτως, ο CohesiveLCA κλιμακώνεται πιο ομαλά από τον LCAsz, όπως εξηγήθηκε και παραπάνω, καθώς η επίδοσή του εξαρτάται από το μέγιστο πληθικό αριθμό των συνεκτικών όρων μιας ερώτησης σε αντίθεση με το

συνολικό αριθμό λέξεων-κλειδιών που καθορίζουν την επίδοση του LCAsz.

Η Εικόνα 5.8 παρουσιάζει τη σύγκριση των χρόνων αποτίμησης των CohesiveLCA, LCAsz και SAOne για ερωτήσεις 6 λέξεων-κλειδιών όταν ο αριθμός των στιγμιοτύπων τους μεταβάλλεται. Οι μετρήσεις είναι μέσες τιμές χρόνων εκτέλεσης για πολλαπλές τυχαίες ερωτήσεις και πρότυπα συνεκτικών σχέσεων (για τον CohesveLCA), όπως και στα προηγούμενα πειράματα. Όπως είναι φανερό, ο CohesiveLCA ξεκάθαρα υπερτερεί των άλλων δύο αλγορίθμων. Ο LCAsz, με τη σειρά του, ξεπερνά με άνεση τον SAOne. Ο SAOne, εξάλλου, κλιμακώνεται πολύ χειρότερα από τους άλλους δύο.



Σχήμα 5.8: Σύγκριση κλιμάκωσης CohesiveLCA με άλλες προσεγγίσεις για ερωτήσεις 6 λέξεων-κλειδιών

Κεφάλαιο 6

Σύνοψη και μελλοντικές επεκτάσεις

6.1 Σύνοψη

Στην παρούσα διατριβή εισαγάγαμε τη σημασιολογία του μεγέθους χαμηλότερου κοινού προγόνου (LCA size) ως μέτρου σχετικότητα των αποτελεσμάτων ερωτήσεων με λέξεις-κλειδιά σε δενδρικά δεδομένα. Για την αποτίμηση ερωτήσεων με αυτή τη σημασιολογία, παρουσιάσαμε έναν πρωτότυπο αλγόριθμο που επιστρέφει το σύνολο των αποτελεσμάτων ταξινομημένα με βάση το μέγεθος LCA size. Ο αλγόριθμος LCAsz εκμεταλλεύεται ένα δικτυωτό από στοίβες, η καθεμιά από τις οποίες αντιστοιχεί σε μια διαμέριση των λέξεων-κλειδιών μιας ερώτησης. Εξαιτίας αυτής της δομής, επιδεικνύει γραμμική συμπεριφορά στην επίδοσή του σε σχέση με το μέγεθος της εισόδου για δεδομένο αριθμό από λέξεις-κλειδιά. Παρουσιάσαμε επίσης μια παραλλαγή του LCAsz, η οποία χειρίζεται με ειδικό τρόπο σπάνιες λέξεις-κλειδιά επιταχύνοντας ακόμα περισσότερο την αποτίμηση των ερωτήσεων. Τα πειραματικά μας αποτελέσματα δείχνουν πως ο LCAsz υπερέχει έναντι παλαιότερων προσεγγίσεων, αποτιμά σε πρακτικό χρόνο ερωτήσεις με πολλές λέξεις-κλειδιά και σε πολύ μεγάλα σύνολα δεδομένων και κλιμακώνεται πολύ ομαλά, καθώς οι παράμετροι των ερωτήσεων και των συνόλων δεδομένων αλλάζουν.

Προτείναμε, επίσης, μια νέα προσέγγιση υπολογισμού κορυφαίων αποτελεσμάτων ερωτήσεων με λέξεις-κλειδιά. Με βάση την έννοια του μεγέθους LCA, ορίσαμε μια κλιμακωτή ταξινόμηση αποτελεσμάτων. Κατ΄ επέκταση, προτείναμε προσεγγίσεις επιστροφής τόσο top-k αποτελεσμάτων, όσο και top-k-size. Η σημασιολογία TLCA, με επιστροφή top-k-size αποτελεσμάτων, αποδεικνύεται μια πρωτότυπη ιδέα που είναι πολύ αποτελεσματική πειραματικά σε σχέση με παλαιότερες σημασιολογίες φιλτραρίσματος. Τα πλεονεκτήματά της προκύπτουν από το γεγονός ότι δε στηρίζεται στον εκ των πρωτέρων αποκλεισμό μέρους των αποτελεσμάτων και από τη διευκόλυνση που προσφέρεται στους χρήστες, μέσω της κλιμακωτής ταξινόμησης, για προσδιορισμό της παραμέτρου k με διαισθητικό τρόπο.

Εισαγάγαμε μια νέα μεθοδολογία ιεραρχικής συσταδοποίησης, η οποία ομαδοποιεί αποτελέσματα παρόμοιας δομής και σημασιολογίας. Η μέθοδος αυτή βασίζεται στον ορισμό ομομορφισμών των προτύπων των αποτελεσμάτων μιας ερώτησης. Στο σύστημά μας, οι χρήστες περιηγούνται στις συστάδες των αποτελεσμάτων, οι οποίες είναι οργανωμένες σε διαφορετικά επίπεδα λεπτομέρειας σε ιεραρχία, όπου οι διαφορετικές συστάδες παρουσιάζονται ταξινομημένες. Κατ΄ αυτόν τον τρόπο διευκολύνεται η ανα-

ζήτησή των χρηστών, καθώς καταλήγουν με λίγα βήματα στα αποτελέσματα μέγιστου ενδιαφέροντος. Η πειραματική μας μελέτη κετέδειξε πως ο αλγόριθμος υπολογισμού των προτύπων και των συστάδων τους είναι γρήγορος και κλιμακώνεται ομαλά, ενώ η μεθοδολογία μας ξεπερνά σε αποτελεσματικότητα ανάλογες προσεγγίσεις.

Τέλος, προτείναμε μια νέα γλώσσα ερωτήσεων με λέξεις-κλειδιά που αξιοποιεί σχέσεις συνεκτικότητας. Οι σχέσεις αυτές βοηθούν το χρήστη να αποσαφηνίσει την πρόθεση αναζήτησής του. Έτσι, το σύστημα αναζήτησης απελευθερώνεται από την ανάγκη να "μαντέψει" τη σημασία της ερώτησης του χρήστη και μπορεί πιο εύστοχα να επιστρέψει τα σωστά αποτελέσματα. Ο αποδοτικός αλγόριθμος που αναπτύξαμε για να αποτιμά τις ερωτήσεις αυτές, αποδεικνύεται στην πειραματική μας ανάλυση ότι μπορεί να αποκρίνεται σε πρακτικό χρόνο για ερωτήσεις με πολλές λέξεις-κλειδιά και σε μεγάλα σύνολα δεδομένων. Πειραματικά, επίσης, δείξαμε ότι η γλώσσα που προτάθηκε, ενώ είναι απλή στη φύση της και δε θυσιάζει την άνεση που προσφέρουν οι ερωτήσεις με λέξεις-κλειδιά στους απλούς χρήστες, προσθέτει βαθμούς εκφραστικότητας, καθώς η αποτελεσματικότητά της ξεπερνά με διαφορά προηγούμενες προσεγγίσεις φιλτραρίσματος αποτελεσμάτων.

6.2 Μελλοντικές επεκτάσεις

Η δουλειά που παρουσιάστηκε σ΄ αυτήν τη διατριβή αφορά έρευνα που έχει γίνει σε δενδρικά δεδομένα κατά κύριο λόγο. Παρ΄ όλ΄ αυτά, έχει ήδη δημοσιευθεί δουλειά προς την κατεύθυνση υιοθέτησης των συνεκτικών σχέσεων λέξεων-κλειδιών σε γράφους και συγκεκριμένα σε δεδομένα μορφής RDF. Σε σχέση με τις ερωτήσεις με συνεκτικές σχέσεις λέξεων-κλειδιών, μελετάμε εξάλλου το αντίστροφο πρόβλημα: αν ο χρήστης υποβάλει μια απλή ερώτηση με λέξεις-κλειδιά, ποιες είναι οι διαφορετικές σημασίες που μπορεί η ερώτηση αυτή να έχει στο συγκεκριμένο σύνολο δεδομένων; Τις ερμηνείες της ερώτησής του, ο χρήστης μπορεί να τις εξετάσει μέσα σε μια ιεραρχία από συνεκτικές μορφές της ερώτησής του.

Την τρέχουσα περίοδο, βρίσκεται, επίσης, υπό διερεύνηση η δημιουργία ενός νέου μοντέλου ταξινόμησης των αποτελεσμάτων αναζήτησης με λέξεις-κλειδιά, η οποία θα λαμβάνει υπόψιν τη συσχέτιση μεταξύ των όρων αναζήτησης. Στη βιβλιογραφία έχει ξαναμελετηθεί το συγκεκριμένο ζήτημα, αλλά από μια σκοπιά που δε λαμβάνει υπόψιν τις λέξεις-κλειδιά της ερώτησης. Το διανυσματικό μοντέλο που βασίζεται σε σύνολα (set-based vector model), το οποίο έχει εφαρμοστεί με επιτυχία στο χώρο του κλασικού IR επιδεικνύει κάποιες πολύ ενδιαφέρουσες ιδιότητες. Η υιοθέτησή του και ο συνδυαμός του μοντέλου με τη σημασιολογία μεγέθους χαμηλότερου κοινού προγόνου, χωρίς την ανάγκη στατιστικής ανάλυσης του συνόλου δεδομένων εκ των πρωτέρων, είναι το αντικείμενο αυτής της δουλειάς.

Ταυτόχρονα, η μελέτη της ταξινόμησης και συσταδοποίησης δενδρικών προτύπων με βάση τους ομομορφισμούς μονοπατιών και προτύπων, ανοίγει το δρόμο για νέες επεκτάσεις του συστήματός μας. Στόχος είναι, εκτός από την ομαδοποίηση, η διαφοροποίηση των αποτελεσμάτων με βάση τα δενδρικά πρότυπα. Η διαφοροποίηση (diversification) των αποτελεσμάτων είναι μία πολύ αποτελεσματική προσέγγιση εποπτείας πολυάριθμων αποτελεσμάτων αναζήτησης, ιδιαίτερα αν δεν υπάρχει δυνατότητα κατάταξής τους σε απόλυτη σειρά, ώστε να μπορεί κανείς ν΄ αναζητήσει μόνο τα κορυφαία αποτελέσματα. Μια τέτοια προσέγγιση δίνει μια αφαιρετική εικόνα των διαφορετικών τύπων αποτελεσμάτων που απαντούν στην αναζήτηση του χρήστη, καθώς οι ομάδες αποτελεσμάτων είναι σημαντικά λιγότερες σε σχέση με το πλήθος των επιμέρους α-

ποτελεσμάτων. Στις επιτυχημένες περιπτώσεις διαφοροποίησης, τ΄ αντιπροσωπευτικά αποτελέσματα των ομάδων δίνουν μια σαφή αίσθηση της διάκρισης των διαφορετικών ειδών αποτελεσμάτων, που έχει ανακαλυφθεί.

Ο αλγόριθμος LCAsz και τα παράγωγά του έχει φανεί στην πειραματική ανάλυση πως επιδεικνύουν ταχύτατους χρόνους αποτίμησης ακόμα και σε μεγάλα σύνολα δεδομένων. Η αδυναμία τους είναι το μέγεθος του δικτυωτού (lattice) που επιβαρύνει την αποτίμηση για ερωτήσεις πολλών λέξεων - κλειδιών. Στις περιπτώσεις που το μέγεθος του δικτυωτού μπορεί ν΄ απομειωθεί, όπως στην αποτίμηση των ερωτήσεων συνεκτικών λέξεων - κλειδιών ή των ερωτήσεων με σπάνιους όρους, υπερνικάται η πολυπλοκότητα που προκύπτει απ΄ αυτό το χαρακτηριστικό. Κατά συνέπεια, στα μελλοντικά σχέδια είναι να μελετηθεί η υιοθέτηση λογικής MapReduce στο σχεδιαμό του βασικού αλγορίθμου. Μια τέτοια προσαρμογή βασίζεται στην ιδέα, ότι η επεξεργασία στις στοίβες που αντιστοιχούν σ΄ ένα επίπεδο του δικτυωτού μπορεί να πραγματοποιείται παράλληλα. Αυτή η επέκταση του αλγορίθμου είναι δυνατόν να καταστήσει τον αλγόριθμο ικανό να διαχειριστεί πραγματικά big data, ανοίγοντας νέα πεδία διερεύνσης.

Bibliography

[1] C. Aksoy, A. Dimitriou, and D. Theodoratos. Reasoning with Patterns to Effectively Answer XML Keyword Queries. *VLDBJ*, 2015, doi:10.1007/s00778-015-0384-3.

[2] C. Aksoy, A. Dimitriou, D. Theodoratos, and X. Wu. XReason: A Semantic Approach that Reasons with Patterns to Answer XML Keyword Queries. In *DASFAA*, pages 299–314, 2013.

[3] S. Amer-Yahia and M. Lalmas. XML Search: Languages, INEX and Scoring. *SIGMOD Record*, 35(4):16–23, 2006.

[4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search*. Pearson Education Ltd., England, 2011.

[5] Z. Bao, T. W. Ling, B. Chen, and J. Lu. Effective XML Keyword Search with Relevance Oriented Ranking. In *ICDE*, pages 517–528, 2009.

[6] Z. Bao, J. Lu, T. W. Ling, and B. Chen. Towards an Effective XML Keyword Search. *IEEE Trans. Knowl. Data Eng.*, 22(8):1077–1092, 2010.

[7] C. Botev and J. Shanmugasundaram. Context-sensitive keyword search and ranking for XML. In *Proceedings of the Eight International Workshop on the Web & Databases (WebDB 2005), Baltimore, Maryland, USA, Collocated mith ACM SIGMOD/PODS 2005, June 16-17, 2005*, pages 115–120, 2005.

[8] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30(1-7):107–117, 1998.

[9] C. Carpineto, S. Osinski, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3), 2009.

[10] O. C. L. Center. Dewey Decimal Classification, 2006.

[11] L. J. Chen and Y. Papakonstantinou. Supporting top-K Keyword Search in XML Databases. In *ICDE*, pages 689–700, 2010.

[12] S. Cohen, Y. Kanza, and Y. Sagiv. Generating Relations from XML Documents. In *ICDT*, pages 282–296, 2003.

[13] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSEarch: A Semantic Search Engine for XML. In *VLDB*, pages 45–56, 2003.

[14] T. Dalamagas, T. Cheng, K. Winkel, and T. K. Sellis. A methodology for clustering XML documents by structure. *Inf. Syst.*, 31(3):187–228, 2006.

[15] E. Demidova, X. Zhou, and W. Nejdl. Iq$^{\mathrm{p}}$: Incremental query construction, a probabilistic approach. In *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, pages 349–352, 2010.

[16] E. Demidova, X. Zhou, and W. Nejdl. A probabilistic scheme for keyword-based incremental query construction. *IEEE Trans. Knowl. Data Eng.*, 24(3):426–439, 2012.

[17] A. Dimitriou and D. Theodoratos. Efficient keyword search on large tree structured datasets. In *KEYS*, pages 63–74, 2012.

[18] A. Dimitriou, D. Theodoratos, and T. Sellis. Top-k-size keyword search on tree structured data. *Inf. Syst.*, 47:178–193, 2015.

[19] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: Ranked Keyword Search over XML Documents. In *SIGMOD Conference*, pages 16–27, 2003.

[20] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, pages 9–16, 2006.

[21] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastava. Keyword Proximity Search in XML Trees. *IEEE TKDE*, 18(4):525–539, 2006.

[22] V. Hristidis, Y. Papakonstantinou, and A. Balmin. Keyword proximity search on xml graphs. In *ICDE*, pages 367–378, 2003.

[23] L. Kong, R. Gilleron, and A. Lemay. Retrieving meaningful relaxed tightest fragments for XML keyword search. In *EDBT*, pages 815–826, 2009.

[24] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 658–665, 2004.

[25] M. Ley. DBLP (Digital Bibliography & Library Project) http://www.informatik.uni-trier.de/ ley/db/, 2000.

[26] G. Li, J. Feng, J. Wang, and L. Zhou. Effective Keyword Search for Valuable LCAs over XML documents. In *CIKM*, pages 31–40, 2007.

[27] G. Li, C. Li, J. Feng, and L. Zhou. SAIL: Structure-aware Indexing for Effective and Progressive top-k Keyword Search over XML Documents. *Inf. Sci.*, 179(21):3745–3762, 2009.

[28] J. Li, C. Liu, R. Zhou, and W. Wang. Suggestion of Promising Result Types for XML Keyword Search. In *EDBT*, pages 561–572, 2010.

[29] J. Li, C. Liu, R. Zhou, and W. Wang. Top-k Keyword Search over Probabilistic XML Data. In *ICDE*, pages 673–684, 2011.

[30] J. Li and J. Wang. Xqsuggest: An interactive XML keyword search system. In *Database and Expert Systems Applications, 20th International Conference, DEXA 2009, Linz, Austria, August 31 - September 4, 2009. Proceedings*, pages 340–347, 2009.

[31] Y. Li, C. Yu, and H. V. Jagadish. Schema-Free XQuery. In *VLDB*, pages 72–83, 2004.

[32] W. Lian, D. W. Cheung, N. Mamoulis, and S. Yiu. An efficient and scalable algorithm for clustering XML documents by structure. *IEEE Trans. Knowl. Data Eng.*, 16(1):82–96, 2004.

[33] J. Liu, J. T. Wang, J. Hu, and B. Tian. A method for aligning RNA secondary structures and its application to RNA motif detection. *BMC Bioinformatics*, 6:89, 2005.

[34] X. Liu, C. Wan, and L. Chen. Returning Clustered Results for Keyword Search on XML Documents. *IEEE TKDE*, 23(12):1811–1825, 2011.

[35] Z. Liu and Y. Chen. Identifying meaningful return information for XML keyword search. In *SIGMOD Conference*, pages 329–340, 2007.

[36] Z. Liu and Y. Chen. Answering keyword queries on XML using materialized views. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 1501–1503, 2008.

[37] Z. Liu and Y. Chen. Reasoning and Identifying Relevant Matches for XML Keyword Search. *PVLDB*, 1(1):921–932, 2008.

[38] Z. Liu and Y. Chen. Return specification inference and result clustering for keyword search on XML. *ACM Trans. Database Syst.*, 35(2), 2010.

[39] Z. Liu and Y. Chen. Processing Keyword Search on XML: a Survey. *WWW*, 14(5-6):671–707, 2011.

[40] Y. Lu, W. Wang, J. Li, and C. Liu. Xclean: Providing valid spelling suggestions for XML keyword queries. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, pages 661–672, 2011.

[41] NASA. NASA XML project http://www.cs.washington.edu/research/xmldatasets-/www/repository.html, 2001.

[42] K. Nguyen and J. Cao. Top-k Answers for XML Keyword Queries. *WWW*, 15(5-6):485–515, 2012.

[43] K. Q. Pu and X. Yu. Keyword query cleaning. *PVLDB*, 1(1):909–920, 2008.

[44] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[45] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[46] A. Schmidt, M. L. Kersten, and M. Windhouwer. Querying XML Documents Made Easy: Nearest Concept Queries. In *ICDE*, pages 321–329, 2001.

[47] F. Shao, L. Guo, C. Botev, A. Bhaskar, M. Chettiar, F. Yang, and J. Shanmugasundaram. Efficient keyword search over virtual XML views. *VLDB J.*, 18(2):543–570, 2009.

[48] C. Sun, C. Y. Chan, and A. K. Goenka. Multiway SLCA-based Keyword Search in XML Data. In *WWW*, pages 1043–1052, 2007.

[49] A. Termehchy and M. Winslett. Using Structural Information in XML Keyword Search Effectively. *ACM Trans. Database Syst.*, 36(1):4, 2011.

[50] XMark. An XML Benchmark Project http://www.xml-benchmark.org, 2001.

[51] Y. Xu and Y. Papakonstantinou. Efficient Keyword Search for Smallest LCAs in XML Databases. In *SIGMOD Conference*, pages 527–538, 2005.

[52] Y. Xu and Y. Papakonstantinou. Efficient LCA based keyword search in XML data. In *EDBT*, pages 535–546, 2008.

[53] M. J. Zaki. Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Trans. Knowl. Data Eng.*, 17(8):1021–1035, 2005.

[54] J. Zhou, Z. Bao, W. Wang, T. W. Ling, Z. Chen, X. Lin, and J. Guo. Fast SLCA and ELCA Computation for XML Keyword Queries Based on Set Intersection. In *ICDE*, pages 905–916, 2012.

[55] J. Zhou, Z. Bao, W. Wang, J. Zhao, and X. Meng. Efficient query processing for XML keyword queries based on the idlist index. *VLDB J.*, 23(1):25–50, 2014.

[56] J. Zhou, X. Zhao, W. Wang, Z. Chen, and J. X. Yu. Top-down keyword query processing on XML data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2225–2230, 2013.

[57] R. Zhou, C. Liu, and J. Li. Fast ELCA Computation for Keyword Queries on XML Data. In *EDBT*, pages 549–560, 2010.

Παράρτημα Α΄

Γλωσσάρι

Ελληνικός όρος	**Αγγλικός όρος**
ακριβεια	precision
ανάκτηση πληροφορίας	information retrieval (IR)
ανεστραμμένη λίστα λέξης-κλειδιού	keyword inverted list
αποκλειστικός χαμηλότερος κοινός πρόγονος	exclusive LCA (ELCA)
ελάχιστο συνδετικό υποδέντρο	minimum connecting tree (MCT)
ελάχιστος χαμηλότερος κοινός πρόγονος	smallest LCA (SLCA)
ευρετήριο	index
γλώσσα ερωτήσεων SPARQL	SPARQL
γλώσσα ερωτήσεων SQL	SQL
γλώσσα ερωτήσεων XQuery	XQuery
λέξη κλειδί	keyword
πληρότητα	recall
πολύτιμος χαμηλότερος κοινός πρόγονος	valuable LCA (VLCA)
σημαντικός χαμηλότερος κοινός πρόγονος	meaningful LCA (MLCA)
στιγμιότυπο δέντρου	instance tree, IT
συμπαγής χαμηλότερος κοινός πρόγονος	tight LCA (TLCA)
χαμηλότερος κοινός πρόγονος	lowest common ancestor (LCA)
χρόνος εντοπισμού	reach time

Παράρτημα Β′

Βιογραφικό Σημείωμα

Στοιχεία Επικοινωνίας

Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
Ηρώων Πολυτεχνείου 9. Ζωγράφου
157 80 Αθήνα, Ελλάδα

Τηλέφωνο: (+30) 210 772 3415
Fax: (+30) 210 772 1866
Ηλεκτρονικό ταχυδρομείο (e-mail): angela@dblab.ece.ntua.gr
Προσωπική Σελίδα: http://www.dblab.ece.ntua.gr/~angela/

Σπουδές

- 1996 - 2001: Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, ΕΜΠ

Ερευνητικά Ενδιαφέροντα

- Βάσεις δεδομένων, διαχείριση δεδομένων
- Ημιδομημένα δεδομένα, XML, RDF, OWL
- Αναζήτηση λέξεων - κλειδιών
- Αλγόριθμοι, πολυπλοκότητα
- Σημασιολογικός ιστός
- Ανάκτηση πληροφορίας (Information Retrieval)

Επιστημονικές Δημοσιεύσεις

- Περιοδικά
  - Aggeliki Dimitriou, Dimitri Theodoratos, Timos Sellis. Top-k-size Keyword Search on Tree Structured Data. Information Systems (IS), Elsevier, Volume 47, January 2015, Pages 178-193
  - Cem Aksoy, Aggeliki Dimitriou, Dimitri Theodoratos. Reasoning with Patterns to Effectively Answer XML Keyword Queries. International Journal on Very Large Databases (VLDBJ), Springer, 2015, Vol.24, Issue 3, pages 441-465
- Περιοδικά υπό κρίση ή υποβολή
  - Cem Aksoy, Aggeliki Dimitriou, Ananya Dass, Dimitri Theodoratos. Clustering Query Result for Effective Keyword Search on Tree Data. Transactions on Knowledge and Data Engineering (TKDE), IEEE (*2nd revision*)
  - Aggeliki Dimitriou, Ananya Dass, Dimitri Theodoratos, Yannis Vassiliou. Fishing Cohesiveness in the Sea of Keywords. (*under submission*)
- Συνέδρια
  - Aggeliki Dimitriou, Ananya Dass, Dimitri Theodoratos, Yannis Vassiliou. Cohesive Keyword Search on Trre Data. Proc. of the 19th Intl. Conference on Extending Database Technology (EDBT), 2016, pages 137-148
  - Ananya Dass, Cem Aksoy, Aggeliki Dimitriou, Dimitri Theodoratos, Xiaoying Wu. Diversifying the Results of Keyword Queries on Linked Data. Proc. of the 20th Intl. Conference on Web information System Engineering (WISE'16), 2016, Springer, LNCS, pages 8
  - Ananya Dass, Aggeliki Dimitriou, Cem Aksoy, Dimitri Theodoratos. Incorporating Cohesiveness into Keyword Search on Linked Data. Proc. of the 19th Intl. Conference on Web information System Engineering (WISE'15), 2015, Springer, LNCS, pages 47-62
  - Ananya Dass, Cem Aksoy, Aggeliki Dimitriou, Dimitri Theodoratos. Keyword Pattern Graph Relaxation for Selective Result Space Expansion on Linked Data. Proc. of the 15th Intl. Conference on Web Engineering (ICWE'15), 2015, Springer, LNCS, 18 pages
  - Vasiliki Pouli, Stella Kafetzoglou, Eirini Eleni Tsiropoulou, Aggeliki Dimitriou, Symeon Papavassiliou. Personalized Multimedia Content Retrieval through Relevance Feedback techniques for Enhanced User Experience. IEEE 13th International Conference on Telecommunications (CONTEL), July 2015
  - Ananya Dass, Cem Aksoy, Aggeliki Dimitriou, Dimitri Theodoratos. Exploiting Semantic Result Clustering to Support Keyword Search on Linked Data. Proc. of the 18th Intl. Conference on Web information System Engineering (WISE'14), 2014, Springer, LNCS, 16 pages

– Cem Aksoy, Aggeliki Dimitriou, Dimitri Theodoratos, Xiaoying Wu. XReason: A Semantic Approach that Reasons with Patterns to Answer XML Keyword Queries. Proc. of the 18th Intl. Conference on Database Systems for Advanced Applications (DASFAA'13), Wuhan, China, April 2013, Springer, LNCS 7825, pages 299-314

– Aggeliki Dimitriou, Dimitri Theodoratos. Efficient keyword search on large tree structured datasets. ACM KEYS 2012, Proc. of the Third International Workshop on Keyword Search on Structured Data, Pages 63-74

– Stratis Viglas, Theodore Dalamagas, Vassilis Christophides, Timos K. Sellis, Aggeliki Dimitriou. Application of the Peer-to-Peer Paradigm in Digital Libraries. DELOS Conference 2007: 318-327.

Συμμετοχή σε Επιτροπές

- Κριτής στο διεθνές περιοδικό VLDB Journal
- Εξωτερικός κριτής σε διεθνή περιοδικά και συνέδρια (ACM SIGMOD, CIKM, EDBT, DASFAA, WISE, DOLAP, DAWAK, BHI, BIBM)

Εργασιακή Εμπειρία

- 10/2000 ; σήμερα Κέντρο Δικτύων (ΚΕΔ) Εθνικού Μετσόβιου Πολυτεχνείου (Σχεδιασμός, ανάπτυξη και υποστήριξη ηλεκτρονικών υπηρεσιών - Υπεύθυνη πολυμέσων και τεχνολογιών τηλεδιάσκεψης/τηλεκπαίδευσης)
- 7/2000 - 8/2000 ΔΕΗ (πρακτική άσκηση)
- 10/1999 - 6/2000 Μαθήματα ηλεκτρονικών υπολογιστών σε τμήματα σχολείων δημοτικής εκπαίδευσης

Διδακτική Εμπειρία

- Συμμετοχή στην επίβλεψη των παρακάτω διπλωματικών εργασιών:
  - 'Αυτόματη Θεματική Κατηγοριοποίηση και Σημασιολογική Διεύρυνση Ερωτημάτων για Μηχανή Αναζήτησης με Οντολογίες', Αμαλία Κούρτη, 2008
  - 'GoNToggle: Έξυπνη μηχανή αναζήτησης με χρήση οντολογιών', Γεώργιος Γιαννόπουλος, 2006
- 4 - 11/2008 Εκπαίδευση εκπαιδευτικών πληροφορικής δευτεροβάθμιας εκπαίδευσης στα πλαίσια του έργου «Πρακτική Εκπαίδευση Εκπαιδευτικών Πληροφορικής»
- 9/2006 Διεξαγωγή σεμιναρίου για τεχνολογίες σύγχρονης και ασύγχρονης τηλεκπαίδευσης σε μέλη ΔΕΠ και υπαλλήλους του ΕΜΠ
- 2002 - 2003 Υποστήριξη στο προπτυχιακό μάθημα «Δομές Δεδομένων» της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ

- 10/2001 - 2/2002 Επιτήρηση εργαστηρίου του μαθήματος 1ου εξαμήνου του τμήματος Ηλ/γων Μηχ/κών & Μηχ/κών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου «Εισαγωγή στον Προγραμματισμό Η/Υ»

- 10/1999 - 6/2000 Μαθήματα ηλεκτρονικών υπολογιστών σε τμήματα ανηλίκων

- 9/1997 - 1/1998 Επιτήρηση εργαστηρίου του μαθήματος 1ου εξαμήνου του τμήματος Ηλ/γων Μηχ/κών & Μηχ/κών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου «Εισαγωγή στον Προγραμματισμό Η/Υ»

Συμμετοχή σε Έργα

- 10/2013 - σήμερα Σχεδιασμός και ανάπτυξη της πλατφόρμας ανοικτών μαθημάτων (Open eClass) του «Κεντρικού Μητρώου Ελληνικών Ανοικτών Μαθημάτων», για τους φορείς του Ακαδημαϊκού Διαδικτύου (GUnet)

- 12/2012 - σήμερα Εγκατάσταση, παραμετροποίηση και υποστήριξη του Συστήματος Αποτίμησης Ποιότητας Ευρυζωνικών Συνδέσεων, με στόχο την ενίσχυση της διαφάνειας και του υγιούς ανταγωνισμού στην αγορά τηλεπικοινωνιών της Κύπρου για το ΓΕΡΗΕΤ

- 11/2011 - 4/2015 Σχεδιασμός και ανάπτυξη συστήματος αποθήκευσης και αναζήτησης μεταδεδομένων τρισδιάστατης τηλεόρασης στο πλαίσιο του ευρωπαϊκού ερευνητικού έργου «FP7 - 3DTV Content Search (Multimodal 3DTV database design and implementation) (http://www.3dtvs-project.eu)» σε συνεργασία με την εταιρία Velti.

- 7/2009 - σήμερα Έρευνα και ανάπτυξη κατανεμημένου συστήματος μέτρησης ποιοτικών χαρακτηριστικών των ευρυζωνικών συνδέσεων και του Διαδικτύου, με στόχο την ενίσχυση της διαφάνειας και του υγιούς ανταγωνισμού στην αγορά τηλεπικοινωνιών για την ΕΕΤΤ (http://hyperiontest.gr)

- 2008 - 2011 Ενιαίο Διαδικτυακό Περιβάλλον που Παρέχει Υπηρεσίες στον Πολίτη και στις Επιχειρήσεις (LGAF)

- 5/2008 - 11/2008 Οργάνωση και Υλοποίηση Εκπαίδευσης

- 1/2007 - 10/2007 Ευρυζωνική Αναβάθμιση της Πρόσβασης Σχολείων Στο Πανελλήνιο Σχολικό Δίκτυο στην περιοχή ευθύνης ΕΜΠ

- 12/2006 - 6/2008 Μελέτη και Διαχείριση Υπηρεσιών Δικτύου Ευρυζωνικής Πρόσβασης για Φοιτητές και Σπουδαστές

- 7/2006 - 6/2011 Λειτουργία και Διαχείριση Υπηρεσιών Δικτύου για το Πανελλήνιο Σχολικό Δίκτυο

- 1/2006 - 9/2006 Τηλεματικές Υπηρεσίες και Υπηρεσίες Τηλεκπαίδευσης στο ΕΜΠ

- 7/2005 - 12/2006 Προηγμένες Τηλεματικές Υπηρεσίες για τη Δευτεροβάθμια Εκπαίδευση: α) Ανάπτυξη και Υποστήριξη προηγμένων Τηλεαμτικών Υπηρεσιών β) Υποστήριξη πληροφοριακών συστημάτων για τις εκπαιδευτικές μονάδες Δευτεροβάθμιας Εκπαίδευσης Β΄Α, Δ΄Α Αθήνας και Πειραιά.

- 1/2004 - 6/2005 Ενίσχυση και παρακολούθηση λειτουργίας των ΚΕΠΛΗΝΕΤ
- 9/2003 - 12/2003 Οργάνωση Τεχνικής Στήριξης / Τεχνικός Συντονισμός
- 8/2002 - 9/2002 Μελέτη του Εθνικού Δικτύου Δημόσιας Διοίκησης ΣΥΖΕΥΞΙΣ
- 3/2002 - 12/2002 Προηγμένες Τηλεματικές Υπηρεσίες για τους φορείς του Ακαδημαϊκού Διαδικτύου (Πακέτο εργασίας: Συντονισμένη Ανάπτυξη Προηγμένων Τηλεματικών Υπηρεσιών ; Υπηρεσία Ασύγχρονης Τηλεκπαίδευσης - eClass)
- 1/2002 - 6/2002 Διαχείριση Δικτύου Δεδομένων για το ΚΕΔ του Ε.Μ.Π.
- 10/2001 - 12/2001 Μελέτη & Διαχείριση Δικτύου Κορμού για το ΕΔΕΤ
- 9/2001 - 10/2002 Διαχείριση Δικτύου Τηλεκπαίδευσης
- 1/2001 - 6/2001 Μελέτη & Ανάπτυξη Δικτύου Edunet
- 10/2000 - 12/2004 Μελέτη και Διαχείριση Δικτύου Δεδομένων για το ΚΕΔ του Ε.Μ.Π.
- 9/2000 - 08/2001 Διαχείριση Δικτύου Τηλεκπαίδευσης

Συνεισφορά ανοικτού κώδικα

- Σύστημα Αποτίμησης Ποιότητας Ευρυζωνικών Συνδέσεων (http://github.com/spebs/spebs)
- Πλατφόρμα φιλοξενίας μαθημάτων Open eClass (http://hg.gunet.gr/openeclass/)
- Επεκτάσεις πλατφόρμας φιλοξενίας μαθησθακού υλικού Επινοώ https://github.com/eellak/wp-moodle-lessons-api, https://github.com/eellak/cim-sso-wp-moodle-fb)

Ξένες γλώσσες

- Αγγλικά : Proficiency (University of Cambridge)
- Γερμανικά : Mittelstufe (Goethe Institut)
- Ιταλικά : CELI 4 (Università di Perugia)