6

Ο Αντεστραμμένος Κατάλογος

Περιεχόμενα Κεφαλαίου

6.1 Εισαγωγή

Ο *αντεστραμμένος κατάλογος* (inverted index) είναι η πιο διαδεδομένη μορφή καταλόγου για την οργάνωση των όρων μίας συλλογής εγγράφων. Σε προηγούμενα κεφάλαια αναφέραμε συνοπτικά τα βασικά του χαρακτηριστικά. Στο κεφάλαιο αυτό μελετούμε τη δομή του αντεστραμμένου καταλόγου σε μεγαλύτερο βάθος, δίνοντας έμφαση σε τρεις βασικούς άξονες: (α) στους διαφορετικούς τύπους αντεστραμμένων καταλόγων που έχουν προταθεί στη βιβλιογραφία, (β) στον τρόπο επεξεργασίας διαφόρων τύπων ερωτημάτων, και (γ) στη συμπίεση των λιστών εμφάνισης που καταλαμβάνουν συνήθως μεγάλο χώρο. Σχετικά με τον άξονα (β) θα εστιάσουμε στο διανυσματικό μοντέλο ανάκτησης και θα εξετάσουμε εναλλακτικές μορφές επεξεργασίας ερωτημάτων.

Η χρήση του αντεστραμμένου καταλόγου δεν είναι ο μοναδικός τρόπος να οργανώσουμε μία συλλογή εγγράφων. Στη βιβλιογραφία έχουν προταθεί και άλλες μέθοδοι, οι πιο γνωστές από τις οποίες βασίζονται στις *υπογραφές* (signatures). Εναλλακτικές μεθόδους καταλόγων μελετούμε σε ξεχωριστό κεφάλαιο. Η έμφαση στο κεφάλαιο αυτό δίνεται στον αντεστραμμένο κατάλογο ο οποίος συνήθως είναι πιο αποδοτικός τόσο ως προς το χώρο που καταλαμβάνει όσο και ως προς την ταχύτητα επεξεργασίας των ερωτημάτων.

6.2 Η Δομή του Αντεστραμμένου Καταλόγου

Η χρήση καταλόγων στοχεύει στην αποδοτική επεξεργασία των ερωτημάτων. Η χρήση καταλόγων αποτελεί το βασικότερο μηχανισμό βελτίωσης της απόδοσης ενός Συστήματος Βάσεων Δεδομένων. Για παράδειγμα, το B-δένδρο και ο κατακερματισμός (hashing) είναι δύο από τις σημαντικότερες μεθόδους καταλόγων που υποστηρίζονται και επιτρέπουν την αποφυγή εξέτασης μεγάλου τμήματος της βάσης για τον προσδιορισμό της απάντησης σε ένα ερώτημα SQL. Στην περίπτωση ενός Συστήματος Ανάκτησης Πληροφορίας ο κατάλογος είναι απαραίτητος ώστε να αποφευχθεί η εξέταση όλων των εγγράφων της συλλογής για τον προσδιορισμό των περισσότερο σχετικών εγγράφων ως προς ένα ερώτημα. Η απουσία καταλόγου συνεπάγεται την εξέταση όλων των εγγράφων της συλλογής για τον προσδιορισμό των σχετικών εγγράφων. Είναι προφανές ότι μία τέτοια λύση δεν είναι καθόλου αποδοτική και επομένως η χρήση του καταλόγου κρίνεται απαραίτητη.

Ο αντεστραμμένος κατάλογος είναι η πιο διαδεδομένη μορφή καταλόγου, λόγω της απλότητας αλλά και της πολύ καλής συμπεριφοράς ως προς την απόδοση (ταχύτητα επεξεργασίας) σχετικά με εναλλακτικές μορφές οργάνωσης. Ο

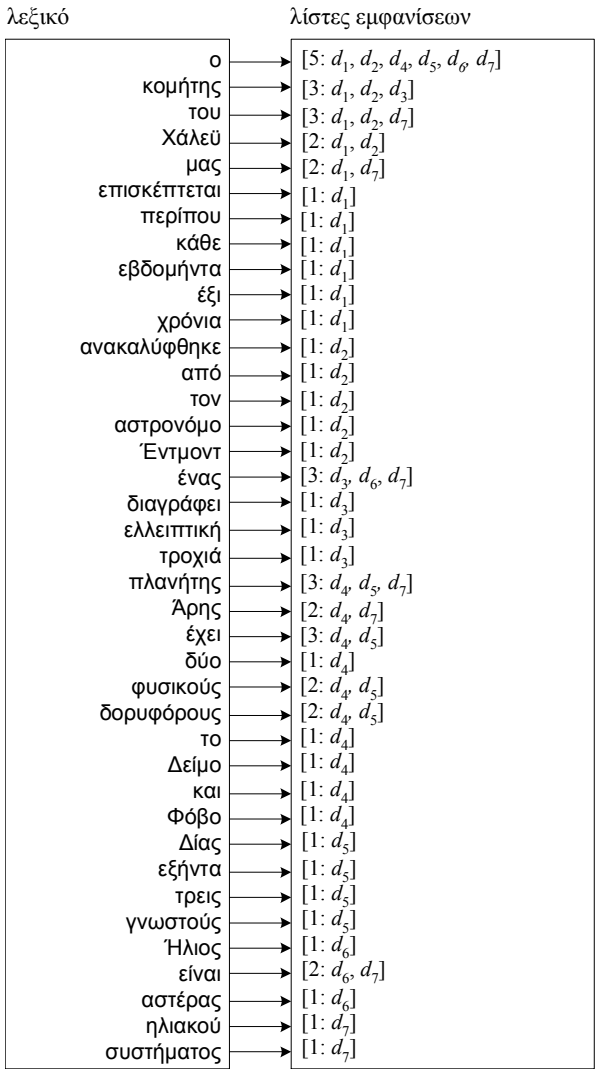
d_1: Ο κομήτης του Χάλλεϋ μας επισκέπτεται περίπου κάθε εβδομήντα έξι χρόνια.
d_2: Ο κομήτης του Χάλλεϋ ανακαλύφθηκε από τον αστρονόμο Έντμοντ Χάλλεϋ.
d_3: Ένας κομήτης διαγράφει ελλειπτική τροχιά.
d_4: Ο πλανήτης Άρης έχει δύο φυσικούς δορυφόρους, το Δείμο και το Φόβο.
d_5: Ο πλανήτης Δίας έχει εξήντα τρεις γνωστούς φυσικούς δορυφόρους.
d_6: Ο Ήλιος είναι ένας αστέρας.
d_7: Ο Άρης είναι ένας πλανήτης του ηλιακού μας συστήματος.

Σχήμα 6.1: *Συλλογή εγγράφων.*

όρος *αντεστραμμένος* χησιμοποιείται για να δηλώσει ότι αντί να κρατούμε πληροφορία σχετικά με το ποιοί όροι βρίσκονται σε ένα έγγραφο, καταγράφουμε το σύνολο των εγγράφων που περιέχουν ένα συγκεκριμένο όρο.

Ένας αντεστραμμένος κατάλογος αποτελείται από δύο βασικά τμήματα: (α) το *λεξικό* (lexicon) και (β) τις *λίστες εμφανίσεων* (posting lists). Σε κάθε όρο αντιστοιχεί μία λίστα εμφανίσεων η οποία καταγράφει πληροφορίες σχετικά με την εμφάνιση του όρου στα έγγραφα. Στην πιο απλή της μορφή μία λίστα εμφανίσεων περιέχει μόνο τους κωδικούς αριθμούς των εγγράφων που εμφανίζεται ο όρος. Ωστόσο, σε πολλές περιπτώσεις καταχωρείται και η θέση εμφάνισης του όρου στο έγγραφο (positional inverted index). Ο δεύτερος τρόπος επιτρέπει την αποδοτική αναζήτηση φράσεων αλλά απαιτεί περισσότερο χώρο για την αποθήκευση του καταλόγου. Επομένως, ανάλογα με το επίπεδο λεπτομέρειας που χρησιμοποιείται για την καταγραφή των εμφανίσεων ενός όρου μπορούν να προκύψουν πολλοί διαφορετικοί αντεστραμμένοι κατάλογοι. Τονίζεται ωστόσο ότι όσο μικρότερο το επίπεδο λεπτομέρειας τόσο λιγότερος χώρος απαιτείται για την αποθήκευση του καταλόγου αλλά αυξάνει ο χρόνος επεξεργασίας ορισμένων ερωτημάτων.

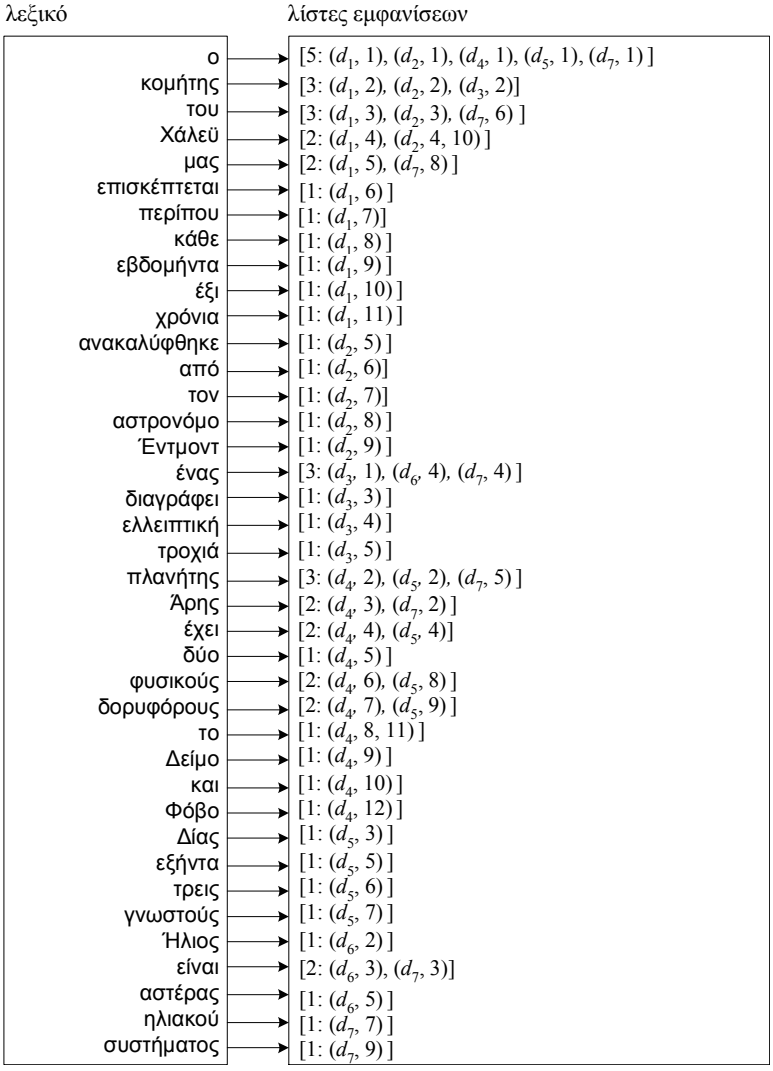
Στο Σχήμα 6.2 δίνεται η δομή του αντεστραμμένου καταλόγου για όλους τους όρους που εμφανίζονται στη συλλογή εγγράφων του Σχήματος 6.1. Η κάθε λίστα εμφανίσεων αποτελείται από έναν ακέραιο αριθμό που δηλώνει τον αριθμό των εγγράφων που περιέχουν τον όρο ακολουθούμενο από ένα σύνολο κωδικών εγγράφων. Για παράδειγμα, η λίστα εμφανίσεων του όρου **κομήτης** είναι [4: d_1, d_2, d_3, d_6] που σημαίνει ότι ο όρος εμφανίζεται σε τέσσερα έγγραφα τα οποία είναι τα d_1, d_2, d_3 και d_6. Εφόσον στις λίστες εμφανίσεων αναφέρεται μόνο ο κωδικός αριθμός του εγγράφου, ο αντεστραμμένος κατάλογος του Σχήματος 6.2 αναφέρεται σε επίπεδο εγγράφου. Αντίθετα, ο κατάλογος του Σχήματος 6.3 έχει καλύτερο επίπεδο λεπτομέρειας διότι για κάθε όρο καταγράφει όχι μόνο σε ποια έγγραφα βρίσκεται ο όρος αλλά και σε ποιες θέσεις μέσα στο κάθε έγγραφο. Για



Σχήμα 6.2: *Αντεστραμμένος κατάλογος επιπέδου εγγράφων.*

παράδειγμα, η λίστα εμφανίσεων του όρου **κομήτης** είναι [4: $(d_1, 2)$, $(d_2, 2)$, $(d_3, 2)$, $(d_6, 2)$] που σημαίνει ότι ο όρος βρίσκεται στη δεύτερη θέση του εγγράφου d_1, στη δεύτερη θέση του εγγράφου d_2, στη δεύτερη θέση του εγγράφου d_3 και στη δεύτερη θέση του εγγράφου d_6. Εναλλακτικά, θα μπορούσαμε να καταγράψουμε τη θέση του πρώτου χαρακτήρα του κάθε όρου μέσα στο έγγραφο, έτσι ώστε η μετακίνηση στις θέσεις που εμφανίζεται ο όρος να γίνεται με μεγαλύτερη ευκολία.

Είναι προφανές ότι ο δεύτερος κατάλογος καταλαμβάνει περισσότερο χώρο



Σχήμα 6.3: *Αντεστραμμένος κατάλογος επιπέδου όρων.*

απ' ότι ο πρώτος. Υπάρχουν δύο λόγοι για τους οποίους συμβαίνει αυτό:

- εκτός από τον κωδικό αριθμό του εγγράφου αποθηκεύεται και η θέση του όρου μέσα στο έγγραφο, και
- σε περίπτωση που ένας όρος εμφανίζεται περισσότερες φορές σε ένα έγγραφο καταγράφονται όλες οι εμφανίσεις του (για παράδειγμα, δείτε τη λίστα εμφανίσεων του όρου Χάλλεϋ).

Σε μία ιδανική κατάσταση, τόσο το λεξικό όσο και οι λίστες εμφανίσεων βρίσκονται στην κύρια μνήμη. Ωστόσο, μία τέτοια υπόθεση δεν είναι ρεαλιστική διότι οι λίστες εμφάνισης καταλαμβάνουν σημαντικό χώρο [2, 21, 75] και επομένως συνήθως αποθηκεύονται στη δευτερεύουσα μνήμη. Για παράδειγμα, για τη συλλογή εγγράφων NewsWire [75] συνολικού μεγέθους 1 GBytes, το μέγεθος του αντεστραμμένου καταλόγου καταλαμβάνει περίπου 430 MBytes, περίπου δηλαδή το 40% του συνολικού μεγέθους της συλλογής. Αντίθετα, το λεξικό καταλαμβάνει σημαντικά λιγότερο χώρο και είναι εφικτή η μόνιμη αποθήκευσή του στην κύρια μνήμη του συστήματος. Σύμφωνα με το νόμο του Heap [31] εάν έχουμε μία συλλογή εγγράφων με συνολικά n λέξεις, τότε ο αριθμός των μοναδικών όρων V (μέγεθος λεξικού) δίνεται από τον τύπο $V = K \cdot n^{\beta}$, όπου η σταθερά K λαμβάνει τιμές μεταξύ 10 και 100 ενώ η σταθερά β λαμβάνει θετικές τιμές μικρότερες της μονάδας. Πειραματικές μετρήσεις με συλλογές εγγράφων από το TREC-2 [1, 2] έχουν δείξει ότι η τιμή του β είναι μεταξύ 0.4 kai 0.6. Φυσικά, οι τιμές των σταθερών K και β εξαρτώνται άμεσα από τη συλλογή εγγράφων. Με βάση τις παραπάνω παρατηρήσεις, έχει διεξαχθεί σημαντική έρευνα στο χώρο με στόχο τη μείωση του χώρου που καταλαμβάνει ο αντεστραμμένος κατάλογος και κυρίως οι λίστες εμφανίσεων. Μία από τις τεχνικές που έχουν προταθεί είναι η συμπίεση των λιστών, που εξετάζεται αργότερα στο κεφάλαιο αυτό.

6.3 Χρήση του Καταλόγου στην Επεξεργασία Ερωτημάτων

Κατά τη μελέτη του Λογικού μοντέλου ανάκτησης (Κεφάλαιο 3) αναφερθήκαμε σε τρόπους επεξεργασίας ερωτημάτων χρησιμοποιώντας τον αντεστραμμένο κατάλογο. Εδώ θα προχωρήσουμε σε μεγαλύτερο βάθος και θα μελετήσουμε μεθόδους επεξεργασίας ερωτημάτων για την περίπτωση του Διανυσματικού μοντέλου ανάκτησης που παρουσιάζει ιδιαίτερο ενδιαφέρον λόγω της μεγάλης αποδοχής του. Τονίζεται ότι θα αντιμετωπίσουμε αρκετές δυσκολίες καθώς με βάση το Διανυσματικό μοντέλο τα έγγραφα θα πρέπει να βαθμολογηθούν ως προς τη σχετικότητά τους με το εκάστοτε ερώτημα.

Θεωρούμε ότι ο βαθμός ομοιότητας μεταξύ q και d δίνεται από το βασικό τύπο της συνάρτησης συνημιτόνου χρησιμοποιώντας συγκεκριμένους σχηματισμούς για τον προσδιορισμό των ποσοτήτων που δίνονται στους Πίνακες 4.3 έως 4.7 του Κεφαλαίου 4. Θεωρούμε λοιπόν ότι ο βαθμός ομοιότητας υπολογίζεται ως εξής:

$$S(q,d) = \frac{1}{L_q \cdot L_d} \cdot \sum_{t \in \mathcal{T}_{q,d}} (1 + \ln(f_{t,d})) \cdot \ln\left(1 + \frac{N}{n_t}\right) \tag{6.1}$$

όπου N το πλήθος των εγγράφων της συλλογής, $L_q = |\vec{q}|$, $L_d = |\vec{d}|$, n_t είναι το πλήθος των εγγράφων που περιέχουν τον όρο t, $\mathcal{T}_{q,d}$ το σύνολο των κοινών όρων του q και d, και $f_{t,d}$ η συχνότητα εμφάνισης του όρου t στο έγγραφο d.

Μία απλή προσέγγιση για τον προσδιορισμό των σχετικών ως προς το q εγγράφων είναι να θεωρήσουμε ότι στην απάντηση ανήκουν όλα τα έγγραφα που εμφανίζουν βαθμό ομοιότητας μεγαλύτερο από το μηδέν. Αυτό εμπεριέχει τον κίνδυνο το πλήθος των εγγράφων της απάντησης να είναι υπερβολικά μεγάλο, οπότε η διαχείριση της απάντησης από το χρήστη γίνεται δυσκολότερη. Ένας άλλος τρόπος προσδιορισμού της απάντησης είναι να θεωρήσουμε ότι μας ενδιαφέρουν τα k έγγραφα που εμφανίζουν το μεγαλύτερο βαθμό ομοιότητας. Με τον τρόπο αυτό πετυχαίνουμε λιγότερα έγγραφα στην απάντηση και επομένως μικρότερη κατανάλωση πόρων για την επεξεργασία του ερωτήματος. Όταν εντοπιστούν τα k έγγραφα με το μεγαλύτερο βαθμό η διαδικασία επεξεργασίας τερματίζεται.

Αν υποθέσουμε ότι δεν έχουμε στη διάθεσή μας κάποιον κατάλογο, τότε ο μοναδικός τρόπος επεξεργασίας του ερωτήματος είναι η ακολουθιακή εξέταση όλων των εγγράφων, ο υπολογισμός της ομοιότητας με το ερώτημα και η αποθήκευση των k μεγαλύτερων τιμών που έχουν υπολογιστεί (εξαντλητική μέθοδος). Όταν εξαντληθούν τα έγγραφα, επιστρέφονται στο χρήστη τα k έγγραφα που έχουν το μεγαλύτερο βαθμό ομοιότητας. Είναι προφανές, ότι το κόστος επεξεργασίας στην περίπτωση αυτή θα είναι πολύ σημαντικό και μάλλον απαγορευτικό για την χρήση του συστήματος, ειδικά όταν το πλήθος των εγγράφων της συλλογής είναι μεγάλο. Επιπλέον, η απουσία καταλόγου σημαίνει ότι δεν υπάρχει τρόπος να διαπιστώσουμε αν ένας όρος έμφανίζεται σε ένα έγγραφο αν δε σαρώσουμε το έγγραφο από την αρχή ως το τέλος.

Στο Σχήμα 6.4 δίνεται ο εξαντλητικός αλγόριθμος υπολογισμού των k εγγράφων με το μεγαλύτερο βαθμό ομοιότητας ως προς το ερώτημα q. Σε κάθε έγγραφο d αντιστοιχεί η μεταβλητή $score_d$ που καταγράφει τον τρέχοντα βαθμό του εγγράφου. Η τιμή $score_d$ αυξάνεται όταν συναντήσουμε στο έγγραφο έναν όρο που υπάρχει στο ερώτημα. Ένας απλός τρόπος καταγραφής των τιμών $score_d$ είναι με τη χρήση ενός μονοδιάστατου πίνακα. Ο απλός αυτός αλγόριθμος επεξεργασίας μπορεί να χρησιμοποιηθεί μόνο σε μικρές συλλογές εγγράφων διότι το κόστος επεξεργασίας αυξάνεται σημαντικά.

Στη συνέχεια θα μελετήσουμε την επεξεργασία ενός ερωτήματος q με στόχο τον προσδιορισμό των k εγγράφων της συλλογής με το μεγαλύτερο βαθμό ομοιό-

Αλγόριθμος Top-k-exhaustive ($\mathcal{D}$, q, k)
$\mathcal{D}$: συλλογή εγγράφων
q: ερώτημα
k: πλήθος εγγράφων της απάντησης

1. για κάθε όρο $t \in \mathcal{T}_q$ υπολογισμός της ποσότητας $idf_t = \ln(1 + N/n_t)$
2. για κάθε έγγραφο $d \in \mathcal{D}$
   2.1. αρχικοποίηση $score_d = 0$
   2.2. για κάθε όρο $t \in \mathcal{T}_q$
   υπολογισμός της ποσότητας $tf_{t,d} = 1 + \ln(f_{t,d})$
   ενημέρωση $score_d = score_d + tf_{t,d} \cdot idf_t$
   2.3. υπολογισμός της ποσότητας L_d
   2.4. ενημέρωση $score_d = score_d / L_d$
3. επιστροφή των k εγγράφων με τη μεγαλύτερη τιμή $score_d$

Σχήμα 6.4: *Εξαντλητικός αλγόριθμος υπολογισμού των k ομοιοτέρων εγγράφων.*

τητας με το ερώτημα χρησιμοποιώτας τον αντεστραμμένο κατάλογο. Η αξία του καταλόγου έγκειται στο γεγονός ότι τμήματα της συλλογής που δεν είναι δυνατό να συμμετέχουν στην απάντηση δεν εξετάζονται, με αποτέλεσμα να εξοικονομείται πολύτιμος χρόνος. Αρχικά πρέπει να βεβαιωθούμε ότι ο αντεστραμμένος κατάλογος περιέχει όλες τις απαραίτητες πληροφορίες για τον υπολογισμό της ομοιότητας μεταξύ ενός ερωτήματος q και ενός εγγράφου d_j. Εστιάζουμε στον αντεστραμμένο κατάλογο επιπέδου εγγράφων, διότι δε μας ενδιαφέρουν οι ακριβείς θέσεις των όρων στα έγγραφα. Ωστόσο, ο κατάλογος του Σχήματος 6.2 δεν μπορεί να χρησιμοποιηθεί απευθείας διότι δεν περιέχει όλες τις απαραίτητες πληροφορίες για τον υπολογισμό του βαθμού ομοιότητας μεταξύ q και d. Πιο συγκεκριμένα, ενώ η ποσότητα n_t είναι διαθέσιμη για κάθε όρο t, η συχνότητα εμφάνισης των όρων στα έγγραφα δεν καταγράφεται. Με την καταγραφή της συχνότητας εμφάνισης προκύπτει ο κατάλογος του Σχήματος 6.5. Παρατηρήστε τη διαφορά από τον κατάλογο του Σχήματος 6.3 όπου μας ενδιαφέρει η θέση της κάθε εμφάνισης ενός όρου σε ένα έγγραφο. Αντιθέτως, στο Σχήμα 6.5 καταγράφεται για κάθε έγγραφο το πλήθος των εμφανίσεων ενός όρου στο έγγραφο (η ποσότητα $f_{t,d}$). Με βάση τη συνάρτηση υπολογισμού της ομοιότητας 6.1 οι ποσότητες L_q και L_d πρέπει να είναι γνωστές. Η τιμή L_q υπολογίζεται για κάθε νέο ερώτημα ενώ η τιμή L_d υπολογίζεται μία φορά για κάθε έγγραφο (κατά την εισαγωγή του εγγράφου στη συλλογή) και αποθηκεύεται. Σε περίπτωση ενημέρωσης του περιεχομένου του εγγράφου, ανανεώνεται και η τιμή L_d. Στο Σχήμα 6.6 οι

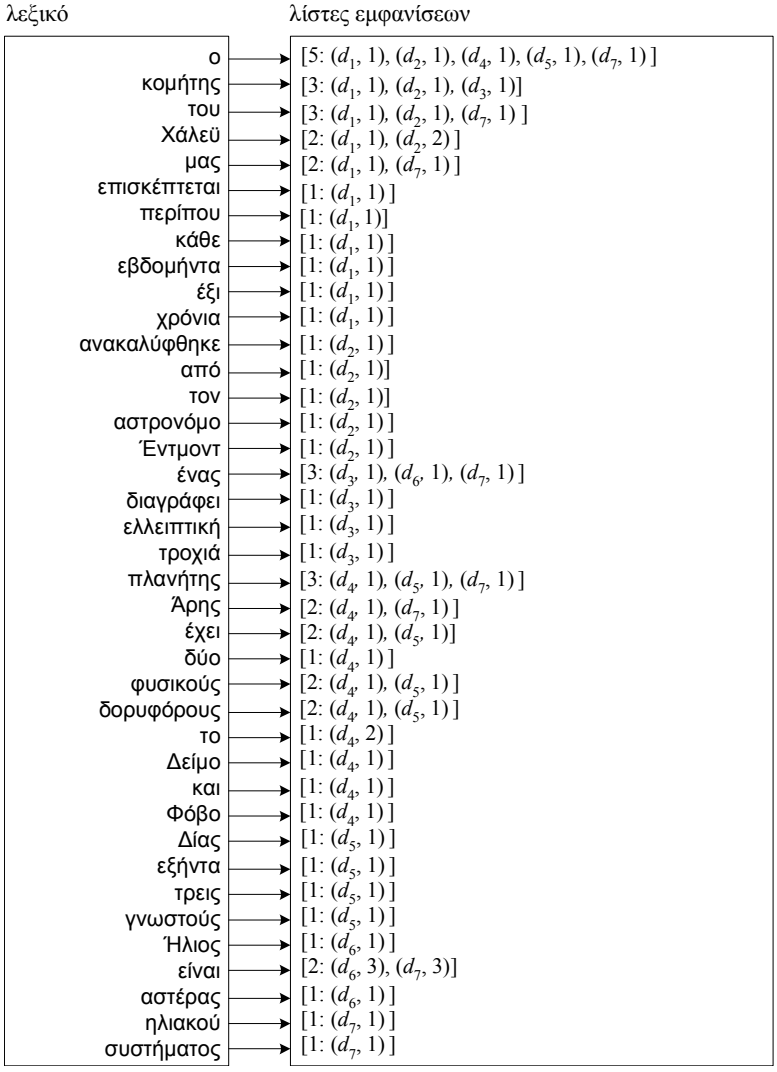
τιμές L_d εμφανίζονται με τη μορφή ενός μονοδιάστατου πίνακα (διανύσματος) του οποίου η j-οστή θέση αντιστοιχεί στην τιμή L_{d_j}.

Εφόσον όλες οι πληροφορίες για τον υπολογισμό του βαθμού ομοιότητας είναι διαθέσιμες, θα πρέπει να προσδιοριστεί και ο τρόπος επεξεργασίας του ερωτήματος με τη χρήση του καταλόγου. Μία πρώτη παρατήρηση είναι ότι δε χρειάζεται να χρησιμοποιηθεί ο όρος L_q για τον υπολογισμό της ομοιότητας καθώς έχει την ίδια επίδραση σε όλα τα έγγραφα. Μία δεύτερη παρατήρηση είναι ότι δε χρειάζεται να επεξεργαστούμε όλα τα έγγραφα της συλλογής, αλλά μόνο τα έγγραφα που περιέχουν όρους κοινούς με το ερώτημα. Από τις λίστες εμφανίσεων είναι γνωστό το υποσύνολο των εγγράφων που περιέχουν ένα συγκεκριμένο όρο. Κάθε φορά που συναντούμε ένα έγγραφο που περιέχει έναν όρο του ερωτήματος, ο όρος αυτός συνεισφέρει στην τιμή της συνάρτησης ομοιότητας. Αρκεί επομένως να αθροίσουμε τις συνεισφορές όλων των όρων του ερωτήματος για τα έγγραφα που περιέχουν έστω και έναν εκ των όρων του ερωτήματος. Η υλοποίηση αυτής της μεθόδου απαιτεί την ύπαρξη ενός συνόλου από *συσσωρευτές* (accumulators). Σε κάθε έγγραφο d που περιέχει έστω και έναν όρο του ερωτήματος αντιστοιχεί ένας συσσωρευτής Σ_d που αρχικοποιείται με μηδέν. Στη συνέχεια, όταν το έγγραφο d συναντηθεί στη λίστα εμφανίσεων ενός όρου t που υπάρχει στο ερώτημα, υπολογίζεται η συνεισφορά του όρου αυτού και το αποτέλεσμα προστίθεται στο συσσωρευτή. Όταν εξαντληθούν οι όροι του ερωτήματος, από τις τιμές των συσσωρευτών επιλέγονται οι k μεγαλύτερες που αντιστοιχούν στα k έγγραφα με την υψηλότερη βαθμολογία. Είναι προφανές ότι έγγραφα που δεν περιέχουν κανέναν εκ των όρων του ερωτήματος βαθμολογούνται με μηδέν και επομένως αγνοούνται.

Ο νέος αλγόριθμος επεξεργασίας που χρησιμοποιεί τον αντεστραμμένο κατάλογο δίνεται στο Σχήμα 6.7, ενώ μία πιο παραστατική επεξήγηση της διαδικασίας επεξεργασίας παρουσιάζεται στο Σχήμα 6.8.

Παράδειγμα 6.1

Ας εξετάσουμε τον τρόπο εύρεσης των k=2 ομοιότερων εγγράφων με βάση τον αλγόριθμο του Σχήματος 6.7 για τον αντεστραμμένο κατάλογο του Σχήματος 6.5. Θα θεωρήσουμε ότι το ερώτημα q αποτελείται από τους όρους t_1 = κόμητης και t_2 = Χάλεϋ. Αρχικά θα επεξεργαστούμε τον όρο κομήτης και στη συνέχεια τον όρο Χάλεϋ. Από τη λίστα εμφανίσεων του όρου κομήτης παρατηρούμε ότι ο όρος βρίσκεται σε τέσσερα έγγραφα. Επομένως, μπορούμε να υπολογίσουμε την ποσότητα idf του όρου που είναι $idf_{t_1} = \ln(1 + N/n_{t_1}) = \ln(2.75) = 1.011$. Δημιουργούνται τέσσερις νέοι συσσωρευτές για τα έγγραφα που περιέχουν τον όρο αυτό, δηλαδή d_1, d_2, d_3 και d_6. Οι τιμές των συσσωρευτών έχουν ως εξής: $\Sigma_{d_1} = tf_{t_1,d_1} \cdot idf_{t_1} = 1.011$ και ομοίως $\Sigma_{d_2} = 1.011$, $\Sigma_{d_3} = 1.011$, $\Sigma_{d_6} = 1.011$.



Σχήμα 6.5: *Αντεστραμμένος κατάλογος επιπέδου εγγράφων με συχνότητες εμφάνισης.*

d_1	d_2	d_3	d_4	d_5	d_6	d_7
2.646	3.296	2.236	3.586	3	2.236	3

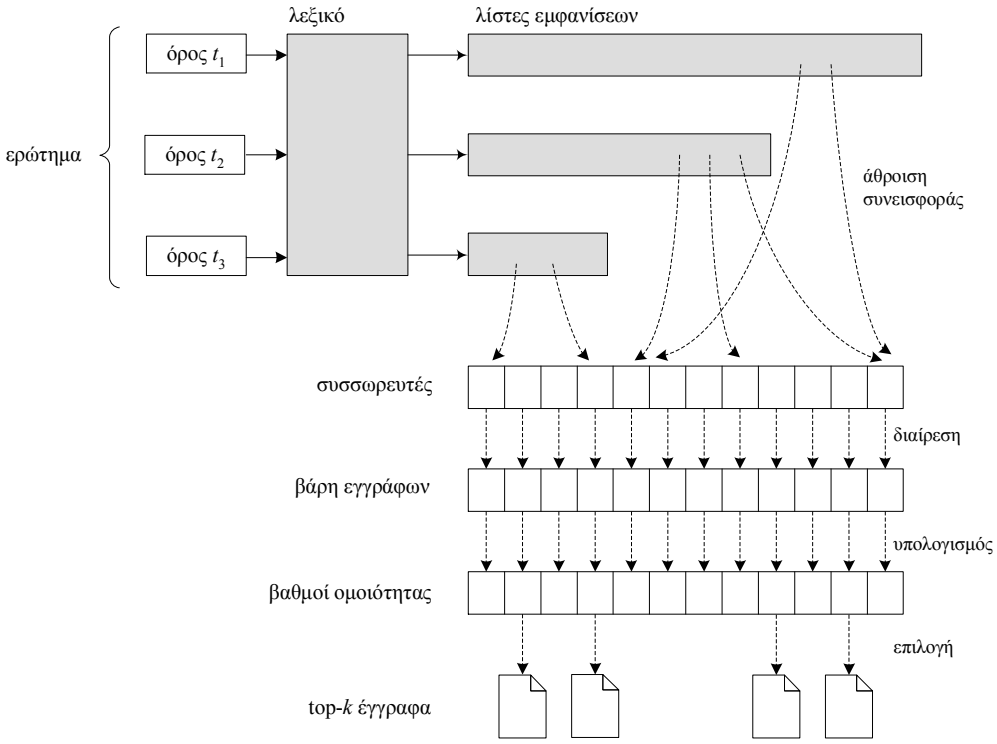
Σχήμα 6.6: *Οι τιμές L_d.*

Αλγόριθμος Top-k-inverted ($\mathcal{D}$, q, k)
$\mathcal{D}$: συλλογή εγγράφων
q: ερώτημα
k: πλήθος εγγράφων της απάντησης

1. αρχικοποίηση $\Sigma = \emptyset$ (σύνολο σωσσωρευτών)
2. για κάθε όρο $t \in \mathcal{T}_q$
   2.1. αναζήτηση του t στο λεξικό
   2.2. ανάγνωση της τιμής n_t
   2.3. υπολογισμός της ποσότητας $idf_t = \ln(1 + N/n_t)$
   2.4. για κάθε ζεύγος $(d, f_{t,d})$ της λίστας εμφανίσεων του t
      2.4.1. αν δεν υπάρχει ο συσσωρευτής Σ_d τότε δημιουργείται
      2.4.2. υπολογισμός της ποσότητας $tf_{t,d} = 1 + \ln(f_{t,d})$
      2.4.3. ενημέρωση του συσσωρευτή $\Sigma_d = \Sigma_d + tf_{t,d} \cdot idf_t$
3. για κάθε συσσωρευτή Σ_d ενημέρωση $\Sigma_d = \Sigma_d / L_d$
4. επιλογή των k μεγαλύτερων συσσωσρευτών και επιστροφή των αποτελεσμάτων

Σχήμα 6.7: *Αλγόριθμος υπολογισμού των k ομοιοτέρων εγγράφων με χρήση αντεστραμμένου καταλόγου.*

Στη συνέχεια εξετάζουμε τον όρο Χάλεϋ ο οποίος εμφανίζεται σε δύο έγγραφα τα d_1 και d_2. Επειδή ήδη υπάρχουν οι συσσωρευτές για τα έγγραφα αυτά, δε χρειάζεται να δημιουργήσουμε νέους. Ωστόσο, θα πρέπει να ενημερωθούν οι τιμές των συσσωρευτών. Ο όρος Χάλευ εμφανίζεται σε δύο έγγραφα, οπότε $n_{t_2} = 2$ και επομένως $idf_{t_2} = 1.504$. Οι μόνοι συσσωρευτές που πρέπει να ενημερωθούν είναι οι Σ_{d_1} και Σ_{d_2}. Άρα έχουμε, $\Sigma_{d_1} = \Sigma_{d_1} + TF_{t_2,d_1} \cdot idf_{t_2} = 1.011 + 1 \cdot 1.504 = 2.515$ και $\Sigma_{d_2} = \Sigma_{d_2} + tf_{t_2,d_2} \cdot idf_{t_2} = 1.011 + 1.693 \cdot 1.504 = 3.557$. Στη συνέχεια διαιρούμε τον κάθε συσσωρευτή με την τιμή L_d και έχουμε τις τελικές τιμές: $\Sigma_{d_1} = 2.515/2.646 = 0.95$, $\Sigma_{d_2} = 3.557/3.296 = 1.079$, $\Sigma_{d_3} = 1.011/2.236 = 0.452$ και $\Sigma_{d_6} = 1.011/3.141 = 0.322$. Είναι προφανές ότι οι συσσωρευτές με τις δύο μεγαλύτερες τιμές είναι οι Σ_{d_2} και Σ_{d_1} και επομένως τα δύο περισσότερο σχετικά έγγραφα ως προς το ερώτημα είναι τα d_2 και d_1. Σε αντίθεση με τον εξαντλητικό αλγόριθμο, που εξετάζει όλα τα έγγραφα της συλλογής, ο αλγόριθμος που χρησιμοποιεί αντεστραμμένο κατάλογο χρειάστηκε να εξετάσει μόνο τέσσερα από τα επτά έγγραφα για να εντοπίσει τα δύο πιο σχετικά ως προς το ερώτημα. Επιπλέον, όλες οι πληροφορίες για τον υπολογισμό του βαθμού ομοιότητας βρίσκονται στη δομή του αντεστραμμένου καταλόγου επομένως δεν απαιτείται η χρήση τεχνικών αναζήτησης των όρων μέσα στα έγγραφα. □



Σχήμα 6.8: *Η διαδικασία επεξεργασίας ερωτήματος top-k.*

6.4 Θέματα Υλοποίησης

Στη συνέχεια εξετάζουμε μερικά ζητήματα υλοποίησης του αντεστραμμένου καταλόγου τα οποία πρέπει να αντιμετωπιστούν με στόχο την αύξηση της απόδοσης του καταλόγου τόσο σε θέματα κατασκευής και συντήρησης όσο και σε θέματα που αφορούν στον απαιτούμενο χώρο για την αποθήκευσή του.

6.4.1 Δημιουργία Καταλόγου

Έως τώρα, θεωρήσαμε δεδομένη την ύπαρξη του καταλόγου και μελετήσαμε μεθόδους επεξεργασίας ερωτημάτων. Ωστόσο, η δημιουργία του καταλόγου αποτελεί ενδιαφέρον πρόβλημα που απαιτεί αποδοτική λύση. Εδώ θα εξετάσουμε την περίπτωση της δημιουργίας του αντεστραμμένου καταλόγου για μία δεδομένη συλλογή εγγράφων. Η διαδικασία αυτή καλείται και *αντιστροφή* (inversion) της συλλογής. Η αντιστροφή χρησιμοποιείται σε περιπτώσεις όπου τα περισσότερα έγγραφα της συλλογής είναι γνωστά. Περαιτέρω αλλαγές στον κατάλογο είναι

Αλγόριθμος BuildInvertedIndex-InMemory ($\mathcal{D}$)
$\mathcal{D}$: συλλογή εγγράφων

1. πρώτη ανάγνωση της συλλογής $\mathcal{D}$, όπου γιά κάθε όρο t
   1.1. υπολογίζεται το πλήθος των εγγράφων που τον περιέχουν (n_t)
   1.2. υπολογίζεται ένα άνω όριο (u_t) για το μήκος της λίστας εμφανίσεων
2. δεσμεύεται στη μνήμη χώρος μεγέθους $\sum u_t$
3. για κάθε t δημιουργείται ένα δείκτης p_t που δείχνει στη λίστα εμφανίσεων
4. δεύτερη ανάγνωση της συλλογής $\mathcal{D}$
   4.1. για κάθε όρο t και κάθε έγγραφο d
      4.1.1 υπολογίζεται η συχνότητα εμφάνισης $f_{t,d}$
      4.1.2 αποθηκεύεται ο κωδικός του εγγράφου και ο αριθμός $f_{t,d}$ στη θέση p_t
      4.1.3 αποθηκεύονται οι θέσεις του όρου t στο έγγραφο d
      4.1.4 μετακινείται ο δείκτης p_t μία θέση δεξιά
5. ανάγνωση του καταλόγου, συμπίεση και αποθήκευση στο δίσκο

Σχήμα 6.9: *Αλγόριθμος αντιστροφής στην κύρια μνήμη.*

εφικτές χρησιμοποιώντας τις λειτουργίες συντήρησης (εισαγωγή νέου εγγράφου, διαγραφή υπάρχοντος εγγράφου).

Η πρώτη τεχνική αντιστροφής που θα μελετήσουμε βασίζεται αποκλειστικά στην κύρια μνήμη (RAM) του συστήματος. Επομένως, για πολύ μεγάλες συλλογές εγγράφων, αυτή η μέθοδος αντιστροφής μπορεί να είναι προβληματική στην περίπτωση που η κύρια μνήμη δεν είναι αρκετή για την αποθήκευση του λεξικού και των λιστών εμφανίσεων των όρων. Ο αλγόριθμος αντιστροφής στην κύρια μνήμη διαβάζει δύο φορές τη συλλογή των εγγράφων $\mathcal{D}$. Κατά την πρώτη ανάγνωση υπολογίζονται οι συχνότητες εμφάνισης των όρων στα έγγραφα, ενώ κατά το δεύτερο πέρασμα τοποθετούνται οι δείκτες των λιστών εμφανίσεων στις κατάλληλες θέσεις, χρησιμοποιώντας τη δυνατότητα της τυχαίας προσπέλασης που προσφέρει η κύρια μνήμη. Τα βασικά βήματα του αλγορίθμου δίνονται στο Σχήμα 6.9. Τονίζεται, ότι σε περίπτωση που δεν απαιτείται η καταχώρηση των θέσεων του κάθε όρου στα έγγραφα, τότε μπορούμε να παραλείψουμε το βήμα 4.1.3., δημιουργώντας έτσι έναν αντεστραμμένο κατάλογο επιπέδου εγγράφου (ένας κατάλογος αυτής της μορφής παρουσιάζεται στο Σχήμα 6.5).

Τα δύο βασικά προβλήματα της προηγούμενης μεθόδου είναι τα ακόλουθα: (α) απαιτούνται δύο αναγνώσεις της συλλογής εγγράφων, ενώ θα πρέπει να αναζητούμε κάθε φορά τους όρους στα έγγραφα και (β) για μεγάλες συλλογές εγγράφων το μέγεθος του καταλόγου ενδέχεται να δημιουργήσει προβλήματα

Αλγόριθμος BuildInvertedIndex-Sorting ($\mathcal{D}$)
$\mathcal{D}$: συλλογή εγγράφων

1. για κάθε έγγραφο $d \in \mathcal{D}$
   1.1. για κάθε όρο $t \in d$ εύρεση της ποσότητας $f_{t,d}$
   1.2. δημιουργία της εγγραφής (t, d, $f_{t,d}$)
2. για κάθε τμήμα του ενδιάμεσου αρχείου
   2.1. ανάγνωση του τμήματος στην κύρια μνήμη
   2.2. ταξινόμηση του τμήματος
   2.3. εγγραφή του ταξινομημένου τμήματος στο δίσκο
3. συγχώνευση των ταξινομημένων τμημάτων
4. ανάγνωση του ταξινομημένου αρχείου και σταδιακή κατασκευή του καταλόγου

Σχήμα 6.10: *Αλγόριθμος αντιστροφής με ταξινόμηση.*

επάρκειας μνήμης. Τα προβλήματα της αντιστροφής εξ' ολοκλήρου στην κύρια μνήμη μπορούν να επιλυθούν χρησιμοποιώντας εναλλακτικές μεθόδους αντιστροφής. Μία από τις μεθόδους αυτές στηρίζεται στην ταξινόμηση. Σύμφωνα με τη μέθοδο αυτή, πραγματοποιείται μία μόνο ανάγνωση της συλλογής, κατά την οποία δημιουργείται ένα ενδιάμεσο αρχείο στο δίσκο που περιέχει εγγραφές της μορφής (t, d, $f_{t,d}$). Αρχικά, το αρχείο είναι ταξινομημένο ως προς τους κωδικούς των εγγράφων, αφού επεξεργαζόμαστε τα αρχεία σειριακά. Στη συνέχεια, το ενδιάμεσο αρχείο ταξινομείται ως προς τους όρους και για κάθε όρο t υπολογίζεται το πλήθος των εγγράφων n_t που τον περιέχουν. Η ταξινόμηση μπορεί να πραγματοποιηθεί χρησιμοποιώντας τη μέθοδο ταξινόμησης με συγχώνευση. Το ενδιάμεσο αρχείο χωρίζεται σε τμήματα, στη συνέχεια κάθε τμήμα ταξινομείται στην κύρια μνήμη και τέλος εγγράφεται στο δίσκο. Ακολουθεί η διαδικασία της συγχώνευσης, αποτέλεσμα της οποίας είναι το ταξινομημένο ενδιάμεσο αρχείο. Τέλος, κατασκευάζεται ο αντεστραμμένος κατάλογος. Με προσεκτικό σχεδιασμό, η μέθοδος αυτή μπορεί να λειτουργήσει χωρίς να δημιουργηθεί θέμα επάρκειας της κύριας μνήμης.

Η τρίτη μέθοδος που εξετάζεται στηρίζεται στη συγχώνευση. Πραγματοποιείται και πάλι μία μόνο ανάγνωση της συλλογής εγγράφων, κατά την οποία δημιουργείται ο αντεστραμμένος κατάλογος στην κύρια μνήμη. Όταν η ελεύθερη εξαντληθεί, τότε το τμήμα του καταλόγου που έχει δημιουργηθεί μαζί με το τμήμα του λεξικού αποθηκεύονται στο δίσκο, ενώ διαγράφονται τα δεδομένα από την κύρια μνήμη. Επαναλαμβάνεται η ίδια διαδικασία μέχρι να ολοκληρωθεί η ανάγνωση και η επεξεργασία της συλλογής. Στη συνέχεια, τα τμήματα που

Αλγόριθμος BuildInvertedIndex-Merging ($\mathcal{D}$)
$\mathcal{D}$: συλλογή εγγράφων

1. μέχρι να εξαντληθεί η ελεύθερη μνήμη
   1.1. ανάγνωση του επόμενου εγγράφου $d \in \mathcal{D}$
   1.2. ενημέρωση του καταλόγου για τους όρους $t \in d$
2. εγγραφή του τμήματος του καταλόγου στο δίσκο
3. αν υπάρχουν και άλλα αρχεία, επανάληψη από το βήμα 1.
4. συγχώνευση των τμημάτων καταλόγου που έχουν δημιουργηθεί

Σχήμα 6.11: *Αλγόριθμος αντιστροφής με συγχώνευση.*

έχουν αποθηκευθεί στο δίσκο συγχωνεύονται ώστε να παραχθεί ο τελικός συνολικός αντεστραμμένος κατάλογος. Σύμφωνα με πειραματικές μελέτες, η μέθοδος της αντιστροφής με συγχώνευση είναι πολύ αποδοτική ακόμη και για μεγάλες συλλογές εγγράφων. Βέβαια, με τη χρήση τεχνικών συμπίεσης μπορούμε να πετύχουμε ακόμη καλύτερα αποτελέσματα τόσο ως προς το χώρο που απαιτείται για την αποθήκευση του καταλόγου όσο και ως προς τον απαιτούμενο χρόνο κατασκευής.

6.4.2 Συντήρηση Καταλόγου

Στις προηγούμενες παραγράφους μελετήθηκε το πρόβλημα της κατασκευής ενός αντεστραμμένου καταλόγου για μία δεδομένη συλλογή εγγράφων. Εάν δεν υπάρχουν αλλαγές στη συλλογή (δεν εισάγονται ούτε διαγράφονται έγγραφα) τότε η μέθοδος της αντιστροφής εκτελείται μία μόνο φορά. Ωστόσο, σε πολλές περιπτώσεις επιβάλλεται η υποστήριξη της ενημέρωσης του καταλόγου λόγω της εισαγωγής νέων εγγράφων (ή της διαγραφής παλαιών). Είναι προφανές, ότι σε μία τέτοια περίπτωση θα πρέπει ο κατάλογος να τροποποιηθεί ώστε να ανταποκρίνεται πλήρως στη συλλογή εγγράφων. Δυστυχώς, η άμεση ενημέρωση του καταλόγου με στόχο την προσθήκη ενός νέου εγγράφου είναι αρκετά χρονοβόρα, ιδιαίτερα για πολύ μεγάλα έγγραφα. Για το λόγο αυτό, έχουν προταθεί εναλλακτικές μέθοδοι ενημέρωσης του καταλόγου με σκοπό τη μείωση του κόστους ενημέρωσης. Οι βασικότερες από αυτές είναι:

Αναδόμηση καταλόγου. Σύμφωνα με την τεχνική αυτή, συλλέγονται νέα έγγραφα προς εισαγωγή και στη συνέχεια πραγματοποιείται μία εκ νέου δημιουργία του καταλόγου, ενώ ο προηγούμενος κατάλογος διαγράφεται.

Ενημέρωση με συγχώνευση. Για τα νέα έγγραφα δημιουργείται ένας προσωρινός κατάλογος που διατηρείται στην κύρια μνήμη του συστήματος. Όταν εξαντληθεί ο ελεύθερος χώρος στη μνήμη, τότε εφαρμόζεται η τεχνική της συγχώνευσης κατά την οποία συγχωνεύονται τα περιεχόμενα του υπάρχοντος καταλόγου που βρίσκονται στο δίσκο με τα περιεχόμενα του προσωρινού καταλόγου της κύριας μνήμης.

Σταδιακή ενημέρωση. Σύμφωνα με τη μέθοδο αυτή, ο κατάλογος ενημερώνεται σταδιακά για κάθε όρο όταν αυτό είναι εφικτό. Για την ενημέρωση της λίστας εμφανίσεων ενός όρου t η λίστα διαβάζεται από το δίσκο στην κύρια μνήμη, ενημερώνεται και στη συνέχεια αποθηκεύεται πάλι στο δίσκο. Η λειτουργία αυτή πρέπει να καθυστερεί όσο γίνεται περισσότερο ώστε να αποφεύγεται κατά το δυνατόν η πολλαπλή ανάγνωση και αποθήκευση της ίδιας λίστας. Ένας εύκολος τρόπος να γίνει αυτό είναι να συλλέγονται τα δεδομένα στην κύρια μνήμη, και όταν το μέγεθος των δεδομένων ξεπεράσει κάποιο όριο τότε να πραγματοποιείται η ενημέρωση των λιστών εμφάνισης. Εναλλακτικά, μία λίστα μπορεί να ενημερωθεί όταν απαιτηθεί η ανάγνωσή της κατά τη διαδικασία επεξεργασίας κάποιου ερωτήματος. Οι τεχνικές αυτές είναι λιγότερο αποδοτικές σε σχέση με την ενημέρωση με συγχώνευση.

Για μετρίου μεγέθους συλλογές εγγράφων η μέθοδος της ενημέρωσης με συγχώνευση έχει την καλύτερη απόδοση. Η μέθοδος της σταδιακής ενημέρωσης χαρακτηρίζεται από δυσκολία στην αποκατάσταση του καταλόγου μετά από κάποιο σφάλμα διότι θα πρέπει να γνωρίζουμε ποιες λίστες έχουν μεταβληθεί και ποιες όχι. Απαιτείται δηλαδή μία μορφή αρχείου ημερολογίου. Η μέθοδος της αναδόμησης χαρακτηρίζεται από απλότητα, διότι δεν απαιτεί κάποια εξειδικευμένη διαδικασία ενημέρωσης. Το κόστος της αναδόμησης για συλλογές που δεν χαρακηρίζονται από συχνή μεταβολή των εγγράφων αναμένεται να είναι είναι μεγαλύτερο σε σχέση με αυτό των άλλων μεθόδων. Ωστόσο, σε περιπτώσεις όπου οι αλλαγές είναι συχνές και μπορεί να αφορούν ακόμη και την προσθήκη ή μεταβολή εγγράφου σε οποιοδήποτε σημείο του εγγράφου (φαινόμενο που συναντούμε πολύ συχνά σε έγγραφα του παγκόσμιου ιστού) η επιλογή της αναδόμησης είναι συνήθως η μόνη αποδοτική λύση.

6.4.3 Τεχνικές Συμπίεσης Καταλόγου

Έχει ήδη αναφερθεί ότι το μέγεθος που καταλαμβάνουν οι λίστες εμφανίσεων ενός αντεστραμμένου καταλόγου είναι πολύ σημαντικό και πολλές φορές ξεπερνά το μέγεθος της συλλογής εγγράφων. Για τη μείωση του χώρου της δομής

χρησιμοποιούνται *τεχνικές συμπίεσης* (compression techniques). Με τη συμπίεση της δομής πετυχαίνουμε δύο σημαντικούς στόχους: (α) εξοικονομείται πολύτιμος χώρος στην κύρια μνήμη και (β) μειώνεται ο αριθμός των προσπελάσεων στη δευτερεύουσα μνήμη. Στη συνέχεια εξετάζονται μερικές από τις μεθόδους συμπίεσης που έχουν προταθεί στη βιβλιογραφία, εστιάζοντας για λόγους απλότητας στον αντεστραμμένο κατάλογο επιπέδου εγγράφων.

Υπενθυμίζεται ότι μία λίστα εμφανίσεων έχει τη μορφή $\langle n_t; d_{i_1}, d_{i_2}, ..., d_{i_{nt}} \rangle$, όπου n_t είναι το πλήθος των εγγράφων που περιέχουν τον όρο t και d_{i_j} είναι ο κωδικός αριθμός ενός εγγράφου. Συνήθως, οι κωδικοί των εγγράφων σε μία λίστα εμφανίσεων αποθηκεύονται κατά αύξουσα διάταξη. Άρα, η λίστα των εγγράφων μπορεί να αναπαρασταθεί με μία *ακολουθία διαφορών* (gap sequence) . Αυτό σημαίνει ότι αντί να αποθηκευτούν απευθείας οι κωδικοί των εγγράφων, αποθηκεύεται ο κωδικός του πρώτου και στη συνέχεια οι διαφορές μεταξύ συνεχόμενων κωδικών. Η ακολουθία διαφορών μπορεί να συμπιεστεί πιο εύκολα από την αρχική λίστα εμφανίσεων.

Παράδειγμα 6.2

Έστω ότι η λίστα εμφανίσεων ενός όρου έχει τη μορφή ⟨7; 134, 45, 130, 20, 10, 100, 120⟩. Αν ταξινομήσουμε τη λίστα παίρνουμε ⟨7; 10, 20, 45, 100, 120, 130, 134⟩. Για την αναπαράσταση της λίστας με χρήση διαφορών διατηρείται ο πρώτος κωδικός εγγράφου (10) και στη συνέχεια παρατίθενται μόνο οι διαφορές των κωδικών από τον προηγούμενο. Εφόσον οι κωδικοί των εγγράφων είναι σε αύξουσα διάταξη, αφαιρώντας τον τρέχοντα κωδικό από τον προηγούμενο προκύπτει ένας θετικός ακέραιος αριθμός. Η λίστα διαφορών είναι η ⟨7; 10, 10, 25, 55, 20, 10, 4⟩. □

Είναι προφανές ότι αναμένονται μικρές διαφορές κωδικών για όρους που εμφανίζονται σε πολλά έγγραφα, ενώ αναμένονται μεγαλύτερες διαφορές για όρους που δεν είναι συχνοί. Αυτή η παρατήρηση οδηγεί στην ιδέα να χρησιμοποιηθούν *κωδικοί μεταβλητού μήκους*, οι οποίοι δίνουν λίγα δυαδικά ψηφία σε συχνά εμφανιζόμενους αριθμούς και περισσότερα δυαδικά ψηφία στην αντίθετη περίπτωση. Ένας τρόπος κωδικοποίησης είναι ο *μοναδιαίος κώδικας* (unary code) βάσει του οποίου ένας ακέραιος αριθμός x κωδικοποιείται χρησιμοποιώντας x-1 δυαδικά ψηφία με τιμή 1 και ένα επιπλέον με τιμή 0. Για παράδειγμα, ο αριθμός 5 κωδικοποιείται ως 11110.

Δύο άλλες μέθοδοι κωδικοποίησης προτάθηκαν το 1975 από τον Elias [13]. Η πρώτη μέθοδος καλείται Elias-γ και αναπαριστά έναν ακέραιο αριθμό x με τη σύνθεση δύο διαφορετικών τμημάτων. Το πρώτο τμήμα είναι ο μοναδιαίος

κώδικας του αριθμού $1+\lfloor \log x \rfloor$ και το δεύτερο τμήμα είναι ένας κωδικός που αποτελείται από $\lfloor \log x \rfloor$ δυαδικά ψηφία και αναπαριστά στο δυαδικό σύστημα τον αριθμό x-$2^{\log x}$. Η δεύτερη μέθοδος που προτάθηκε από τον Elias καλείται Elias-δ και έχει μία σημαντική διαφορά. Το πρώτο τμήμα του κώδικα είναι ο αριθμός των δυαδικών ψηφίων που υπάρχουν στον κώδικα Elias-γ του αριθμού x. Το δεύτερο τμήμα παράγεται όπως και προηγουμένως.

Παράδειγμα 6.3

Έστω ότι πρέπει να υπολογιστούν οι κώδικες Elias-γ και Elias-δ για τον ακέραιο αριθμό x=7. Για τον κώδικα Elias-γ απαιτείται ο μοναδιαίος κωδικός του αριθμού $1+\lfloor \log x \rfloor = 1+2 = 3$ που απαρτίζει το πρώτο τμήμα του κωδικού. Ο μοναδιαίος κωδικός για το 3 είναι ο 110. Το δεύτερο τμήμα αποτελείται από τον αριθμό x-$2^{\lfloor \log x \rfloor} = 7\text{-}2^2 = 3$ ο οποίος στο δυαδικό σύστημα είναι ο 11. Επομένως, ο κωδικός Elias-γ του αριθμού 7 είναι ο 11011. Το πρώτο τμήμα του κωδικού Elias-δ είναι ο αριθμός των δυαδικών ψηφίων που έχει ο αντίστοιχος κωδικός Elias-γ. Σύμφωνα με τα προηγούμενα, ο αριθμός των ψηφίων αυτών είναι 5, επομένως το πρώτο τμήμα του κωδικού Elias-δ θα είναι το 101. Το δεύτερο τμήμα του κωδικού είναι ίδιο με αυτό του Elias-γ. Άρα τελικά έχουμε ότι ο κωδικός Elias-δ του ακεραίου x=7 είναι ο 10111. □

Ένας άλλος τρόπος κωδικοποίησης προτάθηκε από τον Golomb [28] και χρησιμοποιεί την γεωμετρική κατανομή. Πιο συγκεκριμένα, η πιθανοτητα η διαφορά δύο κωδικών εγγράφων να είναι x δίνεται από τον ακόλουθο τύπο, όπου p είναι η πιθανότητα εμφάνισης ενός όρου σε ένα έγγραφο.

$$P(x) = (1-p)^{x-1} \cdot p$$

Στην ουσία ο παραπάνω τύπος αναφέρει ότι έχουμε x-1 συνεχόμενες αποτυχίες και μία επιτυχία. Αφού η πιθανότητα μίας αποτυχίας είναι 1-p, η πιθανότητα να έχουμε x-1 συνεχόμενες αποτυχίες είναι $(1-p)^{x-1}$. Για τον προσδιορισμό της πιθανότητας θα πρέπει να είναι γνωστή η τιμή p. Αν N είναι το πλήθος των εγγράφων της συλλογής και M το πλήθος των μοναδικών όρων που χρησιμοποιούνται για την αναπαράσταση των εγγράφων, τότε η ποσότητα p που δηλώνει την πιθανότητα ένα τυχαίο έγγραφο να περιέχει έναν τυχαίο όρο προσδιορίζεται ως εξής:

$$p = \frac{\text{πλήθος δεικτών καταλόγου}}{N \cdot M} \tag{6.2}$$

Η μέθοδος Golomb χρησιμοποιεί την παράμετρο b για να δημιουργήσει τους

κωδικούς. Εάν b=1 τότε η μέθοδος Golomb ταυτίζεται με την κωδικοποίηση μοναδιαίου κωδικού. Η κωδικοποίηση ενός ακεραίου x πραγματοποιείται με τον προσδιορισμό δύο ποσοτήτων, του πηλίκου (quotient) που συμβολίζεται με q και του υπολοίπου (remainder) που συμβολίζεται με r. Οι δύο αυτές ποσότητες προσδιορίζονται ως εξής:

$$q = \left\lfloor \frac{x-1}{b} \right\rfloor$$

$$r = x - 1 - q \cdot b$$

Ο κώδικας Golomb απαρτίζεται από τον αριθμό $q + 1$ στη μορφή μοναδιαίου κωδικού ακολουθούμενο από τον αριθμό r στη δυαδική αναπαράσταση. Ο αριθμός r απαιτεί $\lfloor \log b \rfloor$ δυαδικά ψηφία για την αναπαράστασή του σε περίπτωση που $r < 2^{\lfloor \log b \rfloor} - 1$ ή $\lceil \log b \rceil$ δυαδικά ψηφία διαφορετικά.

Παράδειγμα 6.4

Στο παράδειγμα αυτό εξετάζεται ο τρόπος προσδιορισμού του κωδικού Golomb για τον ακέραιο x=7 χρησιμοποιώντας διαφορετικές τιμές για την παράμετρο b (3, 4 ή 5). Για διευκόλυνση στην παρουσίαση θα υποθέσουμε ότι η μέθοδος Golomb είναι μία συνάρτηση που λαμβάνει δύο παραμέτρους, τον αριθμό x που θέλουμε να κωδικοποιήσουμε και την τιμή για την παράμετρο b. Η συνάρτηση συμβολίζεται με Golomb(x,b). Έστω ότι b=3. Με βάση τον τύπο προσδιορισμού της ποσότητας q έχουμε $q = \lfloor (7-1)/3 \rfloor = 2$. Επομένως, ο αριθμός q+1 είναι ο 3. Επομένως, το πρώτο τμήμα του κωδικού Golomb θα αποτελείται από τον αριθμό 3 σε μορφή μοναδιαίου κωδικού, δηλαδή 110. Στην περίπτωση αυτή έχουμε r=0, οπότε το υπόλοιπο θα αποτελείται από ένα δυαδικό ψηφία ίσο με 0. Άρα, Golomb(7,3)=1100. Για την περίπτωση b=4 έχουμε $q = \lfloor (7-1)/4 \rfloor = 1$, άρα το πρώτο τμήμα του Golomb(7,4) θα είναι ο αριθμός 1+1 = 2 στη μορφή μοναδιαίου κωδικού, δηλαδή 10. Για το υπόλοιπο έχουμε $r = 7-1-1\cdot 4 = 2$. Η αναπαράσταση του αριθμού 2 στο δυαδικό σύστημα είναι 10. Επομένως, Golomb(7,4) = 1010. Τέλος, αν b=5 με παρόμοιους υπολογισμούς καταλήγουμε ότι Golomb(7,5) = 1001. □

Σύμφωνα με τους Gallager και Van Voorhis [27] η κωδικοποίηση Golomb κατασκευάζει βέλτιστους κωδικούς για την γεωμετρική κατανομή χωρίς να υπάρχουν κοινά προθέματα (prefix-free) εάν η παράμετρος b επιλεγεί ώστε να ικανοποιεί την παρακάτω ανισότητα:

$$\boxed{(1-p)^b + (1-p)^{b+1} \leq 1 < (1-p)^{b-1} + (1-p)^b} \tag{6.3}$$

Με επίλυση της παραπάνω σχέσης για την περίπτωση της ισότητας προκύπτει η ακόλουθη σχέση που συνδέει τα p και b:

$$b = \left\lceil \frac{\log(2-p)}{-\log(1-p)} \right\rceil$$

Θεωρώντας ότι η ποσότητα p είναι πολύ μικρότερη της μονάδας (κάτι που μπορεί να φανεί από την εξίσωση 6.2), προκύπτει ο ακόλουθος προσεγγιστικός τύπος για την παράμετρο b:

$$b \approx 0.69 \cdot \frac{N \cdot M}{\text{πλήθος δεικτών καταλόγου}}$$

Έως τώρα θεωρήσαμε ότι η μέθοδος Golomb λειτουργεί με τον ίδιο τρόπο για κάθε λίστα εμφανίσεων. Ωστόσο, μπορεί να επιτευχθεί καλύτερη απόδοση στη συμπίεση του αντεστραμμένου καταλόγου εάν η κάθε λίστα εμφανίσεων συμπιεστεί ξεχωριστά, με διαφορετικές τιμές παραμέτρων. Αυτό είναι απολύτως λογικό, σκεπτόμενοι ότι το πλήθος των εγγράφων όπου εμφανίζεται ένας όρος είναι διαφορετικό. Για κάθε όρο t απαιτείται η γνώση της ποσότητας n_t που συμβολίζει το πλήθος των εγγράφων που περιέχουν τον όρο t. Αν εστιάζουμε στη λίστα εμφανίσεων ενός όρου t, τότε προφανώς το πλήθος των όρων είναι 1 ενώ αντικαθιστούμε το πλήθος των δεικτών του καταλόγου με το πλήθος των δεικτών της λίστας τού όρου t. Αν συμβολίσουμε με b_t την τιμή της παραμέτρου b για τη λίστα εμφανίσεων του όρου t τότε έχουμε:

$$\boxed{b_t \approx 0.69 \cdot \frac{N}{n_t}} \tag{6.4}$$

6.5 Σύνοψη και Περαιτέρω Μελέτη

Ο αντεστραμμένος κατάλογος αποτελεί την πιο διαδεδομένη μέθοδο οργάνωσης μίας συλλογής εγγράφων. Αποτελείται από δύο βασικά τμήματα, το λεξικό όρων και τις λίστες εμφανίσεων. Στην απλούστερη μορφή της, κάθε λίστα εμφανίσεων καταγράφει τους κωδικούς των εγγράφων που περιέχουν τον όρο. Ωστόσο, στις λίστες εμφανίσεων μπορούν να καταγραφούν και άλλες πληροφορορίες όπως το πλήθος των εμφανίσεων του όρου σε ένα έγγραφο ή το πλήθος των εγγράφων που περιέχουν έναν όρο.

Με τη χρήση του αντεστραμμένου καταλόγου είναι δυνατή η αποδοτική επεξεργασία ερωτημάτων χωρίς να απαιτείται η εξέταση του συνόλου των εγγράφων της συλλογής. Η επεξεργασία εστιάζει στα έγγραφα που περιέχουν όρους που υπάρχουν στο ερώτημα και στη συνέχεια σταδιακά ενημερώνει το βαθμό ομοιότητας του κάθε εγγράφου με το ερώτημα. Στο κεφάλαιο αυτό, δόθηκε έμφαση στην επεξεργασία ερωτημάτων σύμφωνα με το διανυσματικό μοντέλο ανάκτησης.

Η κατασκευή του αντεστραμμένου καταλόγου είναι μία πολύ σημαντική διαδικασία. Εάν δίνεται μία συλλογή εγγράφων, η κατασκευή του καταλόγου προϋποθέτει τον προσδιορισμό των εγγράφων που περιέχουν τον κάθε όρο. Σε περίπτωση που το μέγεθος της συλλογής είναι μικρό, ενδέχεται ο κατάλογος να μπορεί να αποθηκευτεί ολόκληρος στην κύρια μνήμη. Στην περίπτωση αυτή μπορεί να χρησιμοποιηθεί η μέθοδος αντιστροφής στην κύρια μνήμη. Διαφορετικά, θα πρέπει η αντιστροφή να πραγματοποιηθεί με τη βοήθεια της δευτερεύουσας μνήμης.

Ένα άλλο σημαντικό θέμα που συζητήθηκε είναι η συμπίεση του καταλόγου. Η συμπίεση οδηγεί σε μικρότερο μέγεθος επομένως απαιτείται λιγότερος χώρος για την αποθήκευσή του, με αποτέλεσμα να μειώνεται το κόστος προσπέλασης στη δευτερεύουσα μνήμη. Μελετήθηκαν μερικές τεχνικές συμπίεσης που βασίζονται σε κωδικούς μεταβλητού μήκους. Από αυτές, η μέθοδος Golomb δίνει τα καλύτερα αποτελέσματα σύμφωνα με πειραματικές μελέτες.

Λόγω της πολύ καλής απόδοσης του αντεστραμμένου καταλόγου, έχει μελετηθεί διεξοδικά στη βιβλιογραφία. Ο ενδιαφερόμενος αναγνώστης που θέλει να εμβαθύνει περισσότερο μπορεί να συμβουλευτεί το άρθρο των Zobel και Moffat [75] που αναλύει τα βασικά θέματα γύρω από τον αντεστραμμένο κατάλογο, καθώς επίσης και τα βιβλία [3, 25, 73] που περιγράφουν τον κατάλογο με πολύ μεγάλη λεπτομέρεια. Ένα πολύ ενδιαφέρον άρθρο σχετικά με τις απλουστεύσεις που μπορούμε να εφαρμόσουμε στη συνάρτηση προσδιορισμού της ομοιότητας για να πετύχουμε αποδοτικότερη επεξεργασία είναι το [36].

Σχετικά με τα θέματα συμπίεσης του καταλόγου, παραπέμπουμε τον αναγνώστη στο άρθρο [67] όπου περιγράφονται διάφορες μέθοδοι συμπίεσης, και στα άρθρα [4, 61] όπου περιγράφονται μέθοδοι προεπεξεργασίας των κωδικών των εγγράφων με στόχο την καλύτερη συμπίεση του καταλόγου. Επίσης, στο άρθρο [59] μελετάται η συμπίεση καταλόγων που αναφέρουν και τις θέσεις των όρων μέσα στα έγγραφα.

6.6 Ασκήσεις

6.1 Να περιγράψετε συνοπτικά τη δομή του αντεστραμμένου καταλόγου.

6.2 Να δώσετε τα διαφορετικά ήδη αντεστραμμένου καταλόγου.

6.3 Να περιγράψετε συνοπτικά τη διαδικασία προσδιορισμού των k ομοιότερων εγγράφων με χρήση του αντεστραμμένου καταλόγου.

6.4 Ποιους τρόπους κατασκευής ενός αντεστραμμένου καταλόγου γνωρίζετε; Να περιγράψετε συνοπτικά τον καθένα.

6.5 Πως μπορούμε να υποστηρίξουμε εισαγωγές νέων εγγράφων σε έναν αντεστραμμένο κατάλογο;

6.6 Ποια τα πλεονεκτήματα της συμπίεσης του καταλόγου;

6.7 Ποιες μεθόδους συμπίεσης αντεστραμμένου καταλόγου γνωρίζετε;

6.8 Αν αλλάξει η σειρά επεξεργασίας των εγγράφων τότε θα αλλάξει και η μορφή του αντεστραμμένου καταλόγου. Συμφωνείτε με την πρόταση αυτή; Να δικαιολογήσετε την απάντησή σας.

6.9 Να κατασκευάσετε ένα πρόγραμμα δημιουργίας αντεστραμμένου καταλόγου επιπέδου εγγράφων με τη χρήση αντιστροφής στην κύρια μνήμη, για τη συλλογή εγγράφων ISI.

6.10 Χρησιμοποιώντας τον κατάλογο που κατασκευάσατε στην Άσκηση A1 να υποστηρίξετε τη λειτουργία επεξεργασίας ερωτημάτων χρησιμοποιώντας ως συνάρτηση βαθμολόγησης τον τύπο 6.1. Σχολιάστε την επίδοση της μεθόδου σε σχέση με την εξαντλητική επεξεργασία των εγγράφων.

6.11 Να περιγράψετε την κατασκευή του αντεστραμμένου καταλόγου χρησιμοποιώντας τη μέθοδο αντιστροφής με ταξινόμηση, δίνοντας ένα παράδειγμα.

6.12 Για τους ακέραιους αριθμούς 10, 11, 12, 13, 14 και 15 να παραθέσετε τους μοναδιαίους κωδικούς, τους κωδικούς Elias-γ και Elias-δ.

6.13 Δίνεται η λίστα εμφανίσεων $\langle 2, 5, 8, 14, 16, 18, 22, 44, 66, 80\rangle$. Γνωρίζουμε επίσης ότι ο συνολικός αριθμός των εγγράφων είναι N=25. Να προσδιορίσετε τη βέλτιση τιμή για την παράμετρο b και να κωδικοποιήσετε τη λίστα εμφάνισης με τη μέθοδο Golomb.

6.14 Χρησιμοποιώντας τη συλλογή CACM, να κατασκευάσετε έναν αντεστραμμένο κατάλογο επιπέδου εγγράφων στην κύρια μνήμη και να καταγράψετε το χώρο που καταλαμβάνουν οι λίστες εμφανίσεων. Στη συνέχεια, να εφαρμόσετε τις μεθόδους συμπίεσης Elias-γ, Elias-δ και Golomb. Για τη μέθοδο Golomb να θεωρήσετε τις περιπτώσεις όπου (α) η παράμετρος b είναι κοινή

για όλες τις λίστες εμφανίσεων και (β) η τιμή της παραμέτρου διαφέρει. Να συγκριθεί το μέγεθος του αρχικού καταλόγου με αυτό του συμπιεσμένου και να σχολιαστούν τα αποτελέσματα.