ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ

κατηγοριοποίηση

Christopher D. Manning
Prabhakar Raghavan
Hinrich Schütze

αναζήτηση

ακρίβεια

συλλογή

σύνδεσμοι

spam

ανάκληση

Εισαγωγή στην ανάκτηση πληροφοριών

ερώτημα

svm

συσταδοποίηση

ευρετήριο

xml

ιστός

κατάταξη

γλωσσικό μοντέλο

Επιστημονική επιμέλεια ελληνικής έκδοσης:
Ιωάννης Ρεφανίδης, *Πανεπιστήμιο Μακεδονίας*

ΚΛΕΙΔΑΡΙΘΜΟΣ

Περιεχόμενα

1 *Ανάκτηση Boole*

ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ

Η έννοια του όρου *ανάκτηση πληροφοριών* (information retrieval, IR) είναι ιδιαίτερα ευρεία. Ακόμα και όταν απλώς βγάζουμε την πιστωτική μας κάρτα από το πορτοφόλι για να πληκτρολογήσουμε τον αριθμό της έχουμε μια μορφή ανάκτησης πληροφοριών. Όμως, ως πεδίο ακαδημαϊκής μελέτης, η *ανάκτηση πληροφοριών* μπορεί να οριστεί ως εξής:

> Ανάκτηση πληροφοριών (ΑΠ) είναι η εύρεση υλικού (συνήθως εγγράφων) αδόμητης φύσης (συνήθως κειμένου) μέσα σε μεγάλες συλλογές (που βρίσκονται συνήθως αποθηκευμένες σε υπολογιστές), το οποίο ικανοποιεί μια ανάγκη πληροφόρησης.

Ορισμένη κατ' αυτόν τον τρόπο, η ανάκτηση πληροφοριών ήταν μια δραστηριότητα που αφορούσε λίγους μόνο ανθρώπους: βιβλιοθηκονόμους, βοηθούς νομικών επαγγελμάτων, και άλλους επαγγελματίες της αναζήτησης. Ο κόσμος μας όμως έχει πλέον αλλάξει και εκατοντάδες εκατομμύρια άνθρωποι ασχολούνται καθημερινά με την ανάκτηση πληροφοριών, όποτε χρησιμοποιούν μια μηχανή αναζήτησης ή ψάχνουν στην ηλεκτρονική τους αλληλογραφία.[1] Η ανάκτηση πληροφοριών εξελίσσεται με ταχείς ρυθμούς στην επικρατέστερη μορφή προσπέλασης δεδομένων, ξεπερνώντας την παραδοσιακή αναζήτηση σε βάσεις δεδομένων (το είδος της αναζήτησης που πραγματοποιείται στην πράξη όταν κάποιος υπάλληλος σας λέει: «Λυπάμαι, μπορώ να βρω τα στοιχεία της παραγγελίας σας μόνο αν μου δώσετε τον αριθμό παραγγελίας»).

Η ανάκτηση πληροφοριών καλύπτει και άλλους τύπους προβλημάτων πληροφόρησης και αναζήτησης δεδομένων, πέρα από αυτά που αναφέρονται στον παραπάνω βασικό ορισμό. Ο όρος «αδόμητα δεδομένα» περιγράφει δεδομένα τα οποία δεν έχουν ξεκάθαρη, σημασιολογικά εμφανή δομή, εύχρηστη για τον υπολογιστή. Είναι το αντίθετο των δομημένων δεδομένων, των οποίων αντιπροσωπευτικό παράδειγμα είναι οι σχεσιακές βάσεις δεδομένων, όπως αυτές που χρησιμοποιούν συνήθως οι εταιρείες για να παρακολουθούν τα αποθέματα των προϊόντων τους και τα στοιχεία του προσωπικού τους. Στην πραγματικότητα,

[1] Στη σημερινή φρασεολογία, η λέξη «αναζήτηση» τείνει να αντικαταστήσει την «ανάκτηση (πληροφοριών)»· ο όρος «αναζήτηση» είναι αρκετά διφορούμενος, ωστόσο στο βιβλίο θα χρησιμοποιούμε τους δύο όρους ως συνώνυμους.

δεν υπάρχουν σχεδόν καθόλου δεδομένα που να είναι πραγματικά «αδόμητα». Αυτό είναι οπωσδήποτε αληθές για όλα τα δεδομένα κειμένου, αν λάβουμε υπόψη την υποκείμενη γλωσσολογική δομή των ανθρώπινων γλωσσών. Ακόμα όμως και όταν ως δομή εννοούμε την εμφανή, ρητά καθορισμένη δομή, τα περισσότερα κείμενα έχουν τέτοια στοιχεία δόμησης, όπως επικεφαλίδες, παραγράφους και υποσημειώσεις, στοιχεία που αντιπροσωπεύονται συνήθως στα έγγραφα με συγκεκριμένη σήμανση (όπως στον κώδικα των ιστοσελίδων). Η ανάκτηση πληροφοριών χρησιμοποιείται και για «ημιδομημένες» αναζητήσεις, για παράδειγμα, όταν θέλουμε να βρούμε κάποιο έγγραφο που να περιλαμβάνει τη λέξη Java στον τίτλο του και τη λέξη νημάτωση στο σώμα κειμένου του.

Το πεδίο της ΑΠ καλύπτει επίσης θέματα σχετικά με την υποστήριξη των χρηστών στη «φυλλομέτρηση» και το φιλτράρισμα συλλογών εγγράφων, ή στην παραπέρα επεξεργασία ενός συνόλου ανακτημένων εγγράφων. Δεδομένου ενός συνόλου εγγράφων, συσταδοποίηση (clustering) είναι η εργασία της κατάταξης των εγγράφων στις κατάλληλες ομάδες, ανάλογα με το περιεχόμενό τους. Μοιάζει με την τακτοποίηση βιβλίων σε ένα ράφι, ανάλογα με το θέμα τους. Δεδομένου ενός συνόλου θεμάτων, τις τρέχουσες ανάγκες πληροφόρησης, ή άλλες κατηγορίες (όπως η καταλληλότητα των συγγραμμάτων για διάφορες ηλικιακές ομάδες), ταξινόμηση (classification) είναι η επιλογή των κλάσεων (τάξεων ή κατηγοριών), αν υπάρχουν, στις οποίες ανήκει κάθε ένα από τα έγγραφα ενός συνόλου εγγράφων. Συχνά την προσεγγίζουμε ταξινομώντας πρώτα μερικά έγγραφα χειρωνακτικά, ελπίζοντας ότι στη συνέχεια θα μπορούμε να ταξινομήσουμε τα νέα έγγραφα αυτομάτως.

Τα συστήματα ανάκτησης πληροφοριών διακρίνονται επίσης ανάλογα με την κλίμακα στην οποία λειτουργούν, και θα ήταν χρήσιμο να ξεχωρίσουμε τρεις σημαντικές κλίμακες. Στην *αναζήτηση στον Ιστό* το σύστημα πρέπει να παρέχει δυνατότητες αναζήτησης σε δισεκατομμύρια έγγραφα αποθηκευμένα σε εκατομμύρια υπολογιστές. Σημαντικά ζητήματα είναι η ανάγκη συγκέντρωσης των εγγράφων για τη δημιουργία ευρετηρίων, η δυνατότητα κατασκευής συστημάτων που να λειτουργούν αποδοτικά σε αυτή την τεράστια κλίμακα, και ο χειρισμός συγκεκριμένων γνωρισμάτων του Ιστού, όπως η αξιοποίηση του υπερ-κειμένου (hypertext) και η αποφυγή της παραπλάνησης από παρόχους ιστότοπων που «μαγειρεύουν» τα περιεχόμενα των σελίδων τους ώστε να παρουσιάζονται υψηλότερα στα αποτελέσματα των μηχανών αναζήτησης, δεδομένης πλέον της μεγάλης εμπορικής σημασίας του Ιστού. Εστιάζουμε σε όλα αυτά τα ζητήματα στα Κεφάλαια 19–21. Στο άλλο άκρο βρίσκεται η *προσωπική ανάκτηση πληροφοριών*. Τα τελευταία χρόνια, τα λειτουργικά συστήματα ευρείας κυκλοφορίας διαθέτουν ενσωματωμένες δυνατότητες ανάκτησης πληροφοριών — όπως το λογισμικό Spotlight στο Mac OS X της Apple ή η δυνατότητα άμεσης αναζήτησης (Instant Search) στα Windows Vista. Τα προγράμματα ηλεκτρονικού ταχυδρομείου, πέρα από την αναζήτηση, παρέχουν και δυνατότητες ταξινόμησης κειμένου: διαθέτουν τουλάχιστον κάποιο φίλτρο ανεπιθύμητης αλληλογραφίας (spam filter) ενώ συχνά περιλαμβάνουν και αυτόματα ή ρυθμιζόμενα από το χρήστη μέσα για την ταξινόμηση των μηνυμάτων και την άμεση τοποθέτησή τους σε συγκεκριμέ-

νους φακέλους. Στα χαρακτηριστικά ζητήματα αυτής της κλίμακας περιλαμβάνεται ο χειρισμός του μεγάλου εύρους διαφορετικών τύπων εγγράφων που μπορεί να συναντήσει κανείς σε έναν συνηθισμένο προσωπικό υπολογιστή, και η κατασκευή του συστήματος αναζήτησης ώστε να λειτουργεί απροβλημάτιστα και να είναι αρκετά ελαφρύ — κατά την εκκίνησή του, την επεξεργασία, αλλά και στη χρήση χώρου στο δίσκο — για να εκτελείται στον υπολογιστή χωρίς να ενοχλεί τον ιδιοκτήτη του. Ενδιάμεσα βρίσκεται ο χώρος της *αναζήτησης σε επίπεδο επιχείρησης, οργανισμού,* και *συγκεκριμένου τομέα*, όπου μπορεί να παρέχονται δυνατότητες ανάκτησης σε συλλογές που περιλαμβάνουν τα εσωτερικά έγγραφα κάποιας εταιρείας, μια βάση δεδομένων για πατέντες, ή άρθρα για την έρευνα στη βιοχημεία. Σε αυτή την περίπτωση, τα έγγραφα αποθηκεύονται συνήθως σε συγκεντρωτικά συστήματα αρχείων, με έναν ή λίγους υπολογιστές να είναι αφιερωμένοι αποκλειστικά στην παροχή δυνατοτήτων αναζήτησης στη συλλογή. Το βιβλίο περιλαμβάνει πολύτιμες τεχνικές για όλο αυτό το φάσμα, αλλά η κάλυψη κάποιων πλευρών της παράλληλης και της κατανεμημένης αναζήτησης σε συστήματα κλίμακας Ιστού είναι σχετικά μικρή, εξαιτίας του μικρού αριθμού δημοσιευμένων άρθρων για τις λεπτομέρειες τέτοιων συστημάτων. Ωστόσο, εκτός και αν απασχοληθούν σε κάποια από τις λίγες εταιρείες που παρέχουν αναζήτηση σε κλίμακα Ιστού, οι περισσότεροι τεχνολόγοι λογισμικού είναι πιθανότερο να ασχοληθούν με την αναζήτηση σε προσωπική κλίμακα ή σε κλίμακα επιχείρησης.

Σε αυτό το κεφάλαιο, ξεκινάμε με ένα πολύ απλό παράδειγμα προβλήματος ΑΠ και παρουσιάζουμε την ιδέα της μήτρας όρων-εγγράφων (Ενότητα 1.1) καθώς και την πρωταρχικής σημασίας δομή δεδομένων του αντεστραμμένου ευρετηρίου (Ενότητα 1.2). Στη συνέχεια εξετάζουμε το μοντέλο ανάκτησης Boole και τον τρόπο επεξεργασίας των ερωτημάτων Boole (Ενότητες 1.3 και 1.4).

1.1 Παράδειγμα προβλήματος ανάκτησης πληροφοριών

Ένα ογκώδες βιβλίο που πολλοί έχουν στη βιβλιοθήκη τους είναι το *Shakespeare's Collected Works* (Άπαντα του Σαίξπηρ). Ας υποθέσουμε ότι θέλετε να βρείτε ποια θεατρικά έργα του Σαίξπηρ περιέχουν τα ονόματα Βρούτος και Καίσαρ αλλά όχι Καλπουρνία (Brutus AND Caesar AND NOT Calpurnia). Ένας τρόπος να το καταφέρετε αυτό είναι να ξεκινήσετε από την αρχή και να διαβάσετε όλα τα έργα, σημειώνοντας όποιο έργο περιλαμβάνει τα ονόματα Brutus και Caesar αλλά αγνοώντας το αν περιλαμβάνει το Calpurnia. Η απλούστερη μορφή ανάκτησης εγγράφων είναι η πραγματοποίηση μιας τέτοιας γραμμικής σάρωσης μέσω του υπολογιστή. Αυτή η διαδικασία συχνά αναφέρεται και ως *grepping* σε κείμενο (grepping through text), από το όνομα της εντολής `grep` του Unix, η οποία χρησιμοποιείται για το συγκεκριμένο σκοπό. Το grepping είναι πολύ αποτελεσματικό, κυρίως χάρη στην ταχύτητα των σύγχρονων υπολογιστών, και συχνά παρέχει χρήσιμες δυνατότητες για αναζήτηση μοτίβων με χαρακτήρες μπαλαντέρ (wildcard) μέσω κανονικών εκφράσεων (regular expressions). Στους σύγχρονους υπολογιστές, για την εκτέλεση απλών ερωτημάτων σε συλλογές μέσου μεγέθους

GREPPING

(τα *Άπαντα του Σαίξπηρ* συνολικά περιέχουν κάτι λιγότερο από ένα εκατομμύριο λέξεις), πραγματικά δεν χρειάζεστε τίποτα παραπάνω.

Για πολλούς άλλους σκοπούς, όμως, χρειάζεστε πράγματι περισσότερα:

1. Για την ταχεία επεξεργασία μεγάλων συλλογών εγγράφων. Η ποσότητα των διαδικτυακά διαθέσιμων δεδομένων έχει αυξηθεί σχεδόν όσο και η ταχύτητα των υπολογιστών, και τώρα πλέον θέλουμε να πραγματοποιούμε αναζητήσεις σε συλλογές της τάξης των δισεκατομμυρίων ως τρισεκατομμυρίων λέξεων.
2. Για ισχυρότερες και πιο ευέλικτες δυνατότητες ταιριάσματος. Για παράδειγμα, δεν θα ήταν πρακτικό να εκτελέσετε το ερώτημα Romans NEAR countrymen με το `grep`, όταν το NEAR θα μπορούσε να σημαίνει «μέσα σε 5 λέξεις» ή «μέσα στην ίδια πρόταση».
3. Για να είναι εφικτή η καταταγμένη ανάκτηση (ranked retrieval, η ανάκτηση με κατάταξη των αποτελεσμάτων). Σε πολλές περιπτώσεις, θέλετε να έχετε τη βέλτιστη απάντηση σε μια ανάγκη πληροφόρησης, μεταξύ πολλών εγγράφων που περιέχουν συγκεκριμένες λέξεις.

ευρετηριο

μητρα συμπτωσης

οροσ

Για να αποφύγουμε τη γραμμική σάρωση των κειμένων για κάθε ερώτημα, μπορούμε να δημιουργήσουμε εκ των προτέρων *ευρετήρια* των εγγράφων. Ας παραμείνουμε στα *Άπαντα του Σαίξπηρ* για να παρουσιάσουμε τις βασικές αρχές του μοντέλου ανάκτησης Boole. Έστω ότι καταγράφουμε για κάθε έγγραφο — στην περίπτωσή μας, για κάθε θεατρικό έργο — αν περιλαμβάνει κάθε μία από το σύνολο των λέξεων που χρησιμοποίησε ο Σαίξπηρ (συνολικά, περίπου 32.000 διαφορετικές λέξεις). Το αποτέλεσμα είναι μια δυαδική *μήτρα σύμπτωσης* (incidence matrix) όρων-εγγράφων, όπως αυτή του Σχήματος 1.1. *Όροι* είναι οι ευρετηριασμένες μονάδες κειμένου (καλύπτονται περαιτέρω στην Ενότητα 2.2)· συνήθως είναι λέξεις και, προς το παρόν, μπορείτε να τους σκέφτεστε ως λέξεις, αλλά στην ορολογία της ανάκτησης πληροφοριών ονομάζονται όροι επειδή κάποιους από αυτούς, όπως τους E9 ή Χονγκ Κονγκ, συνήθως δεν τους θεωρούμε λέξεις. Τώρα, ανάλογα με το αν εξετάζουμε τις γραμμές ή τις στήλες της μήτρας, μπορούμε να ορίσουμε ένα διάνυσμα για κάθε όρο το οποίο θα δείχνει τα έγγραφα όπου εμφανίζεται ο όρος, ή ένα διάνυσμα για κάθε έγγραφο που θα δείχνει τους όρους που περιλαμβάνονται στο έγγραφο.[2]

Για να απαντήσουμε στο ερώτημα Brutus AND Caesar AND NOT Calpurnia, παίρνουμε τα διανύσματα Brutus, Caesar και Calpurnia, βρίσκουμε το συμπλήρωμα του τελευταίου, και εκτελούμε την πράξη AND στα δυαδικά ψηφία:

110100 AND 110111 AND 101111 = 100100

Επομένως, οι απαντήσεις στο ερώτημα είναι *Αντώνιος και Κλεοπάτρα* και *Άμλετ* (Σχήμα 1.2).

μοντελο ανακτησησ Boole

Το *μοντέλο ανάκτησης Boole* είναι ένα μοντέλο ανάκτησης πληροφοριών το οποίο μας επιτρέπει να υποβάλλουμε ερωτήματα που έχουν τη μορφή λογικών

[2] Η καθιερωμένη πρακτική είναι να χρησιμοποιούμε την ανάστροφη μήτρα για να παίρνουμε τους όρους ως διανύσματα-στήλες.

	Αντώνιος και Κλεοπάτρα	Ιούλιος Καίσαρ	Η Τρικυμία	Άμλετ	Οθέλος	Μάκβεθ	…
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
…							

Σχήμα 1.1 Μήτρα σύμπτωσης όρων-εγγράφων. Το στοιχείο (t, d) της μήτρας είναι 1 αν το θεατρικό έργο στη στήλη d περιλαμβάνει τη λέξη της γραμμής t, διαφορετικά είναι 0.

Αντώνιος και Κλεοπάτρα, Πράξη III, Σκηνή ii
Agrippa [Aside to Domitius Enobarbus]: Why, Enobarbus,
When Antony found Julius Caesar dead,
He cried almost to roaring· and he wept
When at Philippi he found Brutus slain.
Άμλετ, Πράξη III, Σκηνή ii
Lord Polonius: I did enact Julius Caesar: I was killed i' the
Capitol· Brutus killed me.

Σχήμα 1.2 Αποτελέσματα για το ερώτημα Brutus AND Caesar AND NOT Calpurnia στα θεατρικά έργα του Σαίξπηρ.

εκφράσεων όρων, δηλαδή ερωτήματα στα οποία οι όροι συνδυάζονται με τους τελεστές AND (και), OR (ή), και NOT (όχι). Το μοντέλο θεωρεί κάθε έγγραφο απλώς ως ένα σύνολο λέξεων.

Ας θεωρήσουμε τώρα ένα πιο ρεαλιστικό σενάριο, που θα το χρησιμοποιήσουμε και ως αφορμή για να παρουσιάσουμε μερικούς ακόμη όρους και τους αντίστοιχους συμβολισμούς. Υποθέστε ότι έχουμε $N = 1$ εκατομμύριο έγγραφα.

ΕΓΓΡΑΦΟ Με τον όρο *έγγραφο* (document) εννοούμε την όποια μονάδα έχουμε επιλέξει ως βάση για το σύστημα ανάκτησης. Θα μπορούσαν να είναι μεμονωμένα εταιρικά υπομνήματα ή κεφάλαια κάποιου βιβλίου (δείτε την Ενότητα 2.1.2 (σελίδα 43) για περαιτέρω ανάλυση). Αναφερόμαστε στο σύνολο των εγγράφων στα οποία

ΣΥΛΛΟΓΗ πραγματοποιούμε την ανάκτηση με τον όρο *συλλογή* (εγγράφων — document

ΣΩΜΑ ΕΓΓΡΑΦΩΝ collection). Συχνά καλείται και *σώμα εγγράφων* (corpus). Υποθέστε ότι κάθε έγγραφο περιλαμβάνει 1.000 περίπου λέξεις (2–3 σελίδες βιβλίου). Αν θεωρήσουμε ότι κάθε λέξη έχει μέγεθος 6 byte κατά μέσο όρο, συμπεριλαμβανομένων των κενών διαστημάτων και της στίξης, τότε αυτή η συλλογή εγγράφων έχει μέγεθος περίπου 6 gigabyte (GB). Τυπικά, θα μπορούσαν να υπάρχουν $M = 500.000$ διακριτοί όροι σε αυτά τα έγγραφα. Δεν υπάρχει τίποτα το ιδιαίτερο στους αριθμούς που επιλέξαμε, και θα μπορούσαν να διαφέρουν κατά μία τάξη μεγέθους ή και περισσότερο, αλλά μας δίνουν μια ιδέα για τις διαστάσεις του είδους των προβλημάτων που καλούμαστε να χειριστούμε. Εξετάζουμε και αιτιολογούμε αυτές τις εκτιμήσεις μεγεθών στην Ενότητα 5.1 (σελίδα 113).

περιστασιακη ανακτηση

Σκοπός μας είναι να αναπτύξουμε ένα σύστημα που θα καλύπτει την *περιστασιακή ανάκτηση* (ad hoc retrieval), η οποία είναι η πιο συνηθισμένη «αποστολή» της ΑΠ. Σε αυτήν, στόχος του συστήματος είναι να παρέχει έγγραφα της συλλογής συναφή με κάποια αυθαίρετη ανάγκη πληροφόρησης του χρήστη, η οποία γνωστοποιείται στο σύστημα μέσω ενός ερωτήματος που υποβάλλει ο χρήστης μία μόνο φορά. *Ανάγκη πληροφόρησης* (information need) είναι το θέμα για το οποίο ο χρήστης θέλει πράγματι να μάθει περισσότερα, και διαφέρει από το *ερώτημα* (query), δηλαδή από αυτό που ο χρήστης μεταβιβάζει στον υπολογιστή στην προσπάθειά του να δείξει στη μηχανή ποια είναι η ανάγκη πληροφόρησής του. Ένα έγγραφο είναι *συναφές* (relevant) όταν ο χρήστης θεωρεί ότι περιέχει πολύτιμες γι' αυτόν πληροφορίες, σε σχέση με την προσωπική του ανάγκη πληροφόρησης. Το παράδειγμα που δώσαμε παραπάνω ήταν μάλλον τεχνητό, με την έννοια ότι η ανάγκη πληροφόρησης οριζόταν με βάση συγκεκριμένες λέξεις ενώ, συνήθως, όταν οι χρήστες ενδιαφέρονται για θέματα όπως «διαρροές σωληνώσεων», θα ήθελαν να εντοπίζουν τα συναφή έγγραφα ανεξάρτητα από το αν περιέχουν ακριβώς αυτές τις λέξεις ή αν εκφράζουν την ίδια έννοια με άλλες λέξεις, όπως για παράδειγμα διάτρηση σωληνώσεων. Για να αξιολογήσουν την *αποτελεσματικότητα* (effectiveness) ενός συστήματος ΑΠ, δηλαδή την ποιότητα των αποτελεσμάτων του, οι χρήστες εξετάζουν συνήθως δύο βασικά στατιστικά στοιχεία για τα αποτελέσματα που επιστρέφει το σύστημα στα ερωτήματά τους:

ανάγκη πληροφόρησης

ερώτημα

συναφεια

αποτελεσματικοτητα

ακριβεια

Ακρίβεια (precision): ποιο ποσοστό των επιστρεφόμενων αποτελεσμάτων είναι συναφές προς την ανάγκη πληροφόρησης;

ανακληση

Ανάκληση (recall): ποιο ποσοστό των συναφών εγγράφων της συλλογής επιστρέφονται από το σύστημα;

Τα μέτρα που χρησιμοποιούνται για την εκτίμηση της συνάφειας και την αξιολόγηση των συστημάτων ανάκτησης, συμπεριλαμβανομένης της ακρίβειας και της ανάκλησης, τα αναλύουμε λεπτομερώς στο Κεφάλαιο 8.

Δεν μπορούμε πλέον να δημιουργήσουμε μια μήτρα όρων-εγγράφων με τόσο απλοϊκό τρόπο. Μια μήτρα διαστάσεων 500K × 1M περιέχει μισό τρισεκατομμύριο ψηφία 0 και 1 — πάρα πολλά για να χωρέσουν στη μνήμη ενός υπολογιστή. Η καίρια παρατήρηση όμως είναι ότι η μήτρα είναι εξαιρετικά αραιή, δηλαδή έχει ελάχιστες μη μηδενικές καταχωρίσεις. Αφού το μήκος κάθε εγγράφου είναι περίπου 1.000 λέξεις, η μήτρα δεν περιέχει περισσότερα από ένα δισεκατομμύριο 1, επομένως, τουλάχιστον το 99,8% των τιμών της είναι μηδενικά. Μια πολύ καλύτερη αναπαράσταση θα ήταν να καταγράφουμε μόνο ό,τι πραγματικά υπάρχει, δηλαδή, τις τιμές 1.

Αυτή η ιδέα αποτελεί το επίκεντρο μιας από τις κυριότερες έννοιες στην ανάκτηση πληροφοριών, του *αντεστραμμένου ευρετηρίου* (inverted index). Το όνομα είναι ουσιαστικά πλεοναστικό: ένα ευρετήριο πάντα αντιστοιχίζει τους όρους πίσω στα σημεία του εγγράφου όπου εμφανίζονται αυτοί οι όροι. Παρόλα αυτά,

αντεστραμμενο ευρετηριο

Σχήμα 1.3 Τα δύο μέρη ενός αντεστραμμένου ευρετηρίου. Το λεξικό συνήθως διατηρείται στη μνήμη μαζί με δείκτες προς κάθε λίστα καταχωρίσεων. Οι λίστες καταχωρίσεων βρίσκονται αποθηκευμένες στον δίσκο.

αντεστραμμένο ευρετήριο, ή μερικές φορές *αντεστραμμένο αρχείο*, είναι ο όρος που έχει καθιερωθεί στην ΑΠ. [3]

Η βασική ιδέα του αντεστραμμένου ευρετηρίου παρουσιάζεται στο Σχήμα 1.3.

ΛΕΞΙΚΟ Διατηρούμε ένα *λεξικό* όρων — το λεξικό συχνά καλείται και *λεξιλόγιο*, αλλά σε αυτό το βιβλίο χρησιμοποιούμε τον όρο *λεξικό* (dictionary) για τη δομή δεδομέ-

ΛΕΞΙΛΟΓΙΟ νων και τον όρο *λεξιλόγιο* (vocabulary) για το σύνολο των όρων. Στη συνέχεια, για κάθε όρο έχουμε μια λίστα στην οποία καταγράφονται τα έγγραφα που περιέχουν αυτόν τον όρο. Κάθε στοιχείο της λίστας — το οποίο δείχνει ότι ένας όρος βρίσκεται σε ένα έγγραφο (και μερικές φορές, όπως θα δούμε παρακάτω, και τις

ΚΑΤΑΧΩΡΙΣΗ θέσεις του στο έγγραφο) — ονομάζεται *καταχώριση* (posting).[4]

ΛΙΣΤΑ ΚΑΤΑΧΩΡΙΣΕΩΝ ΚΑΤΑΧΩΡΙΣΕΙΣ Η λίστα ονομάζεται *λίστα καταχωρίσεων* (postings list) ή αντεστραμμένη λίστα, ενώ όλες οι λίστες καταχωρίσεων μαζί καλούνται *καταχωρίσεις*. Το λεξικό στο Σχήμα 1.3 έχει διαταχθεί αλφαβητικά και κάθε λίστα καταχωρίσεων είναι διατεταγμένη κατά αναγνωριστικό εγγράφου (docID). Στην Ενότητα 1.3 εξηγούμε γιατί αυτό είναι χρήσιμο ενώ παρακάτω εξετάζουμε εναλλακτικές μεθόδους (Ενότητα 7.1.5).

1.2 Μια πρώτη ματιά στη δημιουργία αντεστραμμένων ευρετηρίων

Για να αξιοποιήσουμε κατά το χρόνο ανάκτησης τα οφέλη της επιτάχυνσης που προσφέρει η χρήση ευρετηρίου, πρέπει να έχουμε δημιουργήσει το ευρετήριο εκ των προτέρων. Τα βασικά βήματα είναι:

1. Συλλέγουμε τα έγγραφα που θα συμπεριληφθούν στο ευρετήριο:

[3] Μερικοί ερευνητές της ΑΠ προτιμούν τον όρο αντεστραμμένο αρχείο (inverted file), αλλά εκφράσεις όπως δημιουργία ευρετηρίου και συμπίεση ευρετηρίου είναι πολύ πιο συνηθισμένες από τις δημιουργία αντεστραμμένου αρχείου και συμπίεση αντεστραμμένου αρχείου. Για λόγους συνέπειας, χρησιμοποιούμε τον όρο (αντεστραμμένο) ευρετήριο σε όλο το βιβλίο.

[4] Σε ένα αντεστραμμένο ευρετήριο (χωρίς πληροφορίες θέσεων) μια καταχώριση είναι απλώς το αναγνωριστικό ενός εγγράφου (document ID ή docID), αλλά συσχετίζεται εγγενώς με έναν όρο μέσω της λίστας καταχωρίσεων στην οποία βρίσκεται· γι' αυτό μερικές φορές θεωρούμε ως καταχώριση και ένα ζεύγος της μορφής (όρος, docID).

Friends, Romans, countrymen.	So let it be with Caesar	…

2. Χωρίζουμε το κείμενο σε σύμβολα (tokens), μετατρέποντας κάθε έγγραφο σε μια λίστα συμβόλων:

Friends	Romans	countrymen	So	…

3. Προχωράμε σε γλωσσολογική προεπεξεργασία, παράγοντας μια λίστα κανονικοποιημένων συμβόλων τα οποία θα αποτελέσουν τους όρους του ευρετηρίου:

friend	roman	countryman	so	…

4. Δεικτοδοτούμε τα έγγραφα στα οποία εμφανίζεται κάθε όρος, δημιουργώντας ένα αντεστραμμένο ευρετήριο αποτελούμενο από ένα λεξικό και καταχωρίσεις.

Ορίζουμε και αναλύουμε τα πρώτα στάδια επεξεργασίας, δηλαδή τα βήματα 1–3, στην Ενότητα 2.2. Μέχρι τότε, μπορείτε να θεωρείτε τα *σύμβολα* και τα *κανονικοποιημένα σύμβολα* ως περίπου ισοδύναμα με *λέξεις*. Εδώ υποθέτουμε ότι τα πρώτα τρία βήματα έχουν ήδη πραγματοποιηθεί και εξετάζουμε την κατασκευή ενός απλού αντεστραμμένου ευρετηρίου ακολουθώντας μεθόδους ευρετηρίασης βάσει διάταξης (sort-based indexing).

Μέσα σε μια συλλογή εγγράφων, θεωρούμε ότι κάθε έγγραφο έχει έναν μοναδικό σειριακό αριθμό που είναι γνωστός ως αναγνωριστικό εγγράφου (document identifier ή *docID*). Κατά τη δημιουργία του ευρετηρίου, μπορούμε απλώς να αναθέτουμε διαδοχικούς ακεραίους σε κάθε έγγραφο την πρώτη φορά που το συναντάμε. Τα δεδομένα εισόδου (input) για τη δημιουργία του ευρετηρίου είναι μια λίστα με τα κανονικοποιημένα σύμβολα κάθε εγγράφου, την οποία μπορούμε να θεωρούμε και ως μια λίστα από ζεύγη όρων και αναγνωριστικών εγγράφων, όπως στο Σχήμα 1.4. Το κύριο βήμα στη δημιουργία του ευρετηρίου είναι η *διάταξη* αυτής της λίστας ώστε οι όροι να είναι σε αλφαβητική σειρά, όπως στη μεσαία στήλη του Σχήματος 1.4. Έπειτα, προχωράμε στη συγχώνευση των εμφανίσεων κάθε όρου στο ίδιο έγγραφο.[5] Στη συνέχεια, οι εμφανίσεις (τα «στιγμιότυπα») του ίδιου όρου ομαδοποιούνται και το αποτέλεσμα χωρίζεται στο *λεξικό* και τις *καταχωρίσεις*, όπως φαίνεται στη δεξιά στήλη του Σχήματος 1.4. Επειδή, γενικά, κάθε όρος εμφανίζεται σε πολλά έγγραφα, αυτή η μέθοδος οργάνωσης των δεδομένων βοηθά στη μείωση των απαιτήσεων του ευρετηρίου σε αποθηκευτικό χώρο. Στο λεξικό καταγράφονται επίσης κάποια στατιστικά στοιχεία, όπως το πλήθος των εγγράφων που περιέχουν κάθε όρο (η *συχνότητα εγγράφων* — document frequency — η οποία εδώ αντιπροσωπεύει και το μήκος κάθε λίστας καταχωρίσεων). Αυτή η πληροφορία δεν είναι απαραίτητη για μια βασική μηχανή αναζήτησης Boole, αλλά μας επιτρέπει να βελτιώσουμε την απόδοση της μηχανής αναζήτησης κατά το χρόνο επεξεργασίας του ερωτήματος, ενώ είναι και ένα στατιστικό στοιχείο που χρησιμοποιείται σε πολλά μοντέλα καταταγμένης αναζήτησης. Οι καταχωρίσεις είναι διατεταγμένες σε δεύτερο επίπεδο κατά αναγνωριστικό εγγράφου, πράγμα που αποτελεί τη βάση για την αποδοτική επεξεργασία

docID

διαταξη

συχνοτητα εγγραφων

[5] Οι χρήστες Unix ίσως προσέξουν ότι αυτά τα βήματα μοιάζουν με τη χρήση της ακολουθίας εντολών `sort` και `uniq`.

Έγγραφο 1
I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

όρος	**docID**
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Έγγραφο 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

όρος	**docID**
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

όρος	**συχν. εγγρ.**	→	**λίστες καταχωρίσεων**
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Σχήμα 1.4 Δημιουργία ευρετηρίου με διάταξη και ομαδοποίηση. Οι όροι κάθε εγγράφου, επισημασμένοι με τα αναγνωριστικά εγγράφων τους (*αριστερά*), διατάσσονται αλφαβητικά (*μέσο*). Στη συνέχεια, οι εμφανίσεις του κάθε όρου ομαδοποιούνται ανά λέξη και μετά ανά αναγνωριστικό εγγράφου (docID). Έπειτα, χωρίζονται οι όροι και τα αναγνωριστικά εγγράφων (*δεξιά*). Το λεξικό περιλαμβάνει τους όρους καθώς και ένα δείκτη προς τη λίστα καταχωρίσεων κάθε όρου. Συνήθως περιλαμβάνει και άλλες συνοπτικές πληροφορίες όπως, εδώ, τη συχνότητα εγγράφων για κάθε όρο. Χρησιμοποιούμε αυτές τις πληροφορίες για να βελτιώσουμε την απόδοση κατά το χρόνο επεξεργασίας των ερωτημάτων και, όπως θα δούμε στα επόμενα, για τον προσδιορισμό συντελεστών στάθμισης στα μοντέλα καταταγμένης ανάκτησης. Κάθε λίστα καταχωρίσεων περιέχει τη λίστα των εγγράφων στα οποία βρίσκεται ένας όρος, ενώ μπορεί να αποθηκεύει και άλλες πληροφορίες, όπως τη συχνότητα του όρου (τη συχνότητα κάθε όρου σε κάθε έγγραφο) ή τις θέσεις του όρου σε κάθε έγγραφο.

των ερωτημάτων αναζήτησης. Αυτή η δομή αντεστραμμένου ευρετηρίου είναι η απολύτως επικρατέστερη ως η πλέον αποδοτική για την υποστήριξη περιστασιακών αναζητήσεων κειμένου.

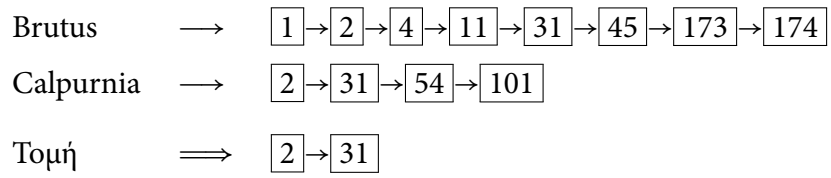
Στο προκύπτον ευρετήριο έχουμε να αντιμετωπίσουμε το αποθηκευτικό κόστος τόσο για το λεξικό όσο και για τις λίστες καταχωρίσεων. Οι τελευταίες είναι πολύ μεγαλύτερες, αλλά το λεξικό συνήθως διατηρείται στη μνήμη ενώ οι λίστες καταχωρίσεων αποθηκεύονται στο δίσκο, οπότε και τα δύο μεγέθη είναι σημαντικά. Στο Κεφάλαιο 5 εξετάζουμε πώς μπορούμε να βελτιστοποιήσουμε και τα δύο ώστε να έχουμε την καλύτερη δυνατή απόδοση, τόσο στις απαιτήσεις αποθήκευσης όσο και στην προσπέλαση. Ποια δομή δεδομένων είναι καταλληλότερη για τις λίστες καταχωρίσεων; Οι πίνακες σταθερού μήκους θα ήταν σπατάλη — κάποιες λέξεις εμφανίζονται σε πολλά έγγραφα ενώ άλλες σε πολύ λίγα. Για λίστες καταχωρίσεων που διατηρούνται στη μνήμη, δύο καλές εναλλακτικές λύσεις είναι οι απλά συνδεδεμένες λίστες (singly linked lists) ή οι πίνακες μεταβλητού μήκους (variable length arrays). Οι *απλά συνδεδεμένες λίστες* επιτρέπουν την «ανέξοδη» παρεμβολή εγγράφων στις λίστες καταχωρίσεων (έπειτα από ενημερώσεις, όπως μετά την επανασταχυολόγηση του Ιστού για ενημερωμένα έγγραφα), και μπορούν να επεκταθούν εύκολα σε πιο σύνθετες στρατηγικές ευρετηρίασης, όπως οι λίστες παράβλεψης (skip lists, Ενότητα 2.3), για τις οποίες απαιτούνται περισσότεροι δείκτες. Οι *πίνακες μεταβλητού μήκους* «κερδίζουν» σε απαιτήσεις χώρου, αποφεύγοντας την επιβάρυνση των δεικτών, αλλά και χρόνου, επειδή χρησιμοποιούν συνεχείς περιοχές μνήμης βελτιώνοντας την ταχύτητα επεξεργασίας στους σύγχρονους επεξεργαστές με ενσωματωμένη κρυφή μνήμη (cache). Στην πράξη, πρόσθετοι δείκτες μπορούν να κωδικοποιηθούν στις λίστες με τη μορφή σχετικών αποστάσεων (ή μετατοπίσεων, offset). Αν οι ενημερώσεις είναι σχετικά σπάνιες, οι πίνακες μεταβλητού μήκους είναι πιο συμπαγείς και επιτρέπουν ταχύτερη διάσχιση. Μπορούμε επίσης να χρησιμοποιήσουμε μια υβριδική μέθοδο, με μια συνδεδεμένη λίστα πινάκων σταθερού μήκους για κάθε όρο. Όταν οι λίστες καταχωρίσεων διατηρούνται στο δίσκο, αποθηκεύονται (πιθανώς συμπιεσμένες) ως συνεχόμενες ακολουθίες καταχωρίσεων χωρίς ενδιάμεσους δείκτες (όπως στο Σχήμα 1.3), ώστε να ελαχιστοποιηθεί το μέγεθός τους αλλά και ο αριθμός των απαιτούμενων αναζητήσεων στο δίσκο για την ανάγνωση μιας λίστας καταχωρίσεων και τη φόρτωσή της στη μνήμη.

Άσκηση 1.1 [⋆] Καταστρώστε το αντεστραμμένο ευρετήριο για την παρακάτω συλλογή εγγράφων. (Δείτε ως παράδειγμα το Σχήμα 1.3.)

Έγγραφο 1 new home sales top forecasts
Έγγραφο 2 home sales rise in july
Έγγραφο 3 increase in home sales in july
Έγγραφο 4 july new home sales rise

Άσκηση 1.2 [⋆] Δίνονται τα εξής έγγραφα:

Έγγραφο 1 breakthrough drug for schizophrenia
Έγγραφο 2 new schizophrenia drug



Σχήμα 1.5 Τομή των λιστών καταχωρίσεων για τα ονόματα Brutus και Calpurnia από το Σχήμα 1.3.

Έγγραφο 3 new approach for treatment of schizophrenia
Έγγραφο 4 new hopes for schizophrenia patients

α. Καταστρώστε τη μήτρα σύμπτωσης όρων-εγγράφων γι' αυτήν τη συλλογή εγγράφων.

β. Καταστρώστε το αντεστραμμένο ευρετήριο της συλλογής, στη μορφή του Σχήματος 1.3 (σελίδα 27).

Άσκηση 1.3 [⋆] Στη συλλογή εγγράφων της Άσκησης 1.2, ποια είναι τα αποτελέσματα που επιστρέφουν τα παρακάτω ερωτήματα;

α. schizophrenia AND drug

β. for AND NOT (drug OR approach)

1.3 Επεξεργασία ερωτημάτων Boole

Πώς επεξεργαζόμαστε ένα ερώτημα χρησιμοποιώντας ένα αντεστραμμένο ευρετήριο και το βασικό μοντέλο ανάκτησης Boole; Ας θεωρήσουμε την επεξεργασία του *απλού συζευκτικού ερωτήματος*:

ΑΠΛΑ ΣΥΖΕΥΚΤΙΚΑ ΕΡΩΤΗΜΑΤΑ

(1.1) Brutus AND Calpurnia

στο αντεστραμμένο ευρετήριο που φαίνεται εν μέρει στο Σχήμα 1.3 (σελίδα 27):

1. Εντοπίζουμε το όνομα Brutus στο λεξικό.
2. Βρίσκουμε τις καταχωρίσεις του.
3. Εντοπίζουμε το όνομα Calpurnia στο λεξικό.
4. Βρίσκουμε τις καταχωρίσεις του.
5. Βρίσκουμε την τομή των δύο λιστών καταχωρίσεων, όπως φαίνεται στο Σχήμα 1.5.

ΤΟΜΗ ΛΙΣΤΩΝ ΚΑΤΑΧΩΡΙΣΕΩΝ

Η πράξη της *τομής* είναι η πιο κρίσιμη: θέλουμε να υπολογίσουμε την τομή των λιστών καταχωρίσεων όσο πιο αποδοτικά γίνεται, ώστε να εντοπίσουμε γρήγορα τα έγγραφα που περιέχουν και τους δύο όρους. (Αυτή η πράξη ονομάζεται μερικές φορές και *συγχώνευση* λιστών καταχωρίσεων, ένα όχι και τόσο σαφές όνομα το οποίο παραπέμπει στη χρήση του όρου *αλγόριθμος συγχώνευσης* (merge algorithm) που χρησιμοποιείται στην ονομασία μιας γενικής οικογένειας αλγορίθμων οι οποίοι μπορούν να συνδυάζουν πολλές διατεταγμένες λίστες διατρέχοντάς τις παράλληλα μέσω δεικτών· εδώ συγχωνεύουμε τις λίστες με τη λογική πράξη AND.)

ΣΥΓΧΩΝΕΥΣΗ ΚΑΤΑΧΩΡΙΣΕΩΝ

```
INTERSECT(p1, p2)
 1  απάντηση ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(απάντηση, docID(p1))
 5             p1 ← επόμενη(p1)
 6             p2 ← επόμενη(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← επόμενη(p1)
 9                else p2 ← επόμενη(p2)
10  return απάντηση
```

Σχήμα 1.6 Αλγόριθμος για την εύρεση της τομής δύο λιστών καταχωρίσεων p_1 και p_2.

Υπάρχει μια απλή και αποτελεσματική μέθοδος για τον υπολογισμό της τομής λιστών καταχωρίσεων με χρήση του αλγορίθμου συγχώνευσης (δείτε το Σχήμα 1.6): διατηρούμε δείκτες στο εσωτερικό και των δύο λιστών και τις διατρέχουμε ταυτόχρονα, σε χρόνο γραμμικά ανάλογο προς το συνολικό πλήθος των καταχωρίσεών τους. Σε κάθε βήμα, συγκρίνουμε τα αναγνωριστικά εγγράφων (docID) στα οποία δείχνουν οι δύο δείκτες. Αν είναι τα ίδια, τοποθετούμε το αναγνωριστικό στη λίστα αποτελεσμάτων και αυξάνουμε και τους δύο δείκτες. Διαφορετικά, αυξάνουμε το δείκτη που δείχνει στο μικρότερο docID. Αν τα μήκη των λιστών καταχωρίσεων είναι x και y, για να ολοκληρωθεί ο υπολογισμός της τομής απαιτούνται $O(x + y)$ πράξεις. Τυπικά, η πολυπλοκότητα της εκτέλεσης του ερωτήματος είναι $\Theta(N)$, όπου N είναι το πλήθος των εγγράφων της συλλογής.[6] Οι μέθοδοι ευρετηρίασης που χρησιμοποιούμε μας αποφέρουν κέρδος κατά μία σταθερά, όχι κάποια διαφορά στην πολυπλοκότητα χρόνου Θ σε σχέση με τη γραμμική σάρωση· στην πράξη, όμως, η σταθερά αυτή είναι τεράστια. Για να χρησιμοποιήσουμε αυτόν τον αλγόριθμο, είναι κρίσιμο οι καταχωρίσεις να είναι διατεταγμένες με τον ίδιο ακριβώς τρόπο σε όλες τις λίστες. Η αριθμητική διάταξη κατά αναγνωριστικό εγγράφου (docID) είναι ένας απλός τρόπος για να πετύχουμε αυτόν τον σκοπό.

Μπορούμε να επεκτείνουμε την πράξη υπολογισμού τομής ώστε να επεξεργάζεται πιο περίπλοκα ερωτήματα, όπως:

(1.2) (Brutus OR Caesar) AND NOT Calpurnia

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΡΩΤΗΜΑΤΟΣ

Βελτιστοποίηση ερωτήματος (query optimization) είναι η διαδικασία με την οποία επιλέγουμε πώς θα οργανώσουμε την εργασία της απάντησης σε ένα ερώτημα, ώστε να ελαχιστοποιηθεί ο φόρτος του έργου που χρειάζεται να πραγματοποιηθεί από το σύστημα. Ιδιαίτερα σημαντική για τη βελτιστοποίηση ερωτημάτων

[6] Ο συμβολισμός $\Theta(\cdot)$ εκφράζει ένα ασυμπτωτικά σφιχτό φράγμα για την πολυπλοκότητα του αλγορίθμου. Συχνά γράφεται και ως $O(\cdot)$, αλλά αυτός ο συμβολισμός εκφράζει στην πραγματικότητα ένα ασυμπτωτικό άνω φράγμα που δεν είναι απαραίτητα σφιχτό. (Cormen κ.ά. 1990).

Boole είναι η σειρά προσπέλασης των λιστών καταχωρίσεων. Ποια είναι η βέλτιστη σειρά για την επεξεργασία του ερωτήματος; Ας θεωρήσουμε ένα ερώτημα *t* όρων που συνδυάζονται με τον τελεστή AND, για παράδειγμα:

(1.3) Brutus AND Caesar AND Calpurnia

Πρέπει να πάρουμε τις καταχωρίσεις καθενός από τους *t* όρους, και μετά να τις συνδυάσουμε με την πράξη AND. Η τυπική μέθοδος είναι να επεξεργαστούμε τους όρους κατά αύξουσα συχνότητα εγγράφων. Αν ξεκινήσουμε με την τομή των δύο μικρότερων λιστών καταχωρίσεων, τότε όλα τα ενδιάμεσα αποτελέσματα δεν θα πρέπει να είναι μεγαλύτερα από τη μικρότερη λίστα καταχωρίσεων, άρα είναι πιθανότερο να επιτύχουμε την ελαχιστοποίηση του απαραίτητου έργου. Επομένως, για τις λίστες καταχωρίσεων του Σχήματος 1.3 (σελίδα 27), εκτελούμε το παραπάνω ερώτημα ως εξής:

(1.4) (Calpurnia AND Brutus) AND Caesar

Αυτός είναι και ένας πρώτος καλός λόγος για να αποθηκεύουμε τις συχνότητες των όρων στο λεξικό· μας επιτρέπουν να πάρουμε την απόφαση για τη σειρά επεξεργασίας βασισμένοι σε δεδομένα που βρίσκονται στη μνήμη, προτού προσπελάσουμε οποιαδήποτε λίστα καταχωρίσεων.

Ας θεωρήσουμε τώρα τη βελτιστοποίηση γενικότερων ερωτημάτων, όπως:

(1.5) (madding OR crowd) AND (ignoble OR strife) AND (killed OR slain)

Όπως και προηγουμένως, παίρνουμε τις συχνότητες όλων των όρων ώστε, στη συνέχεια, να εκτιμήσουμε (συντηρητικά) το μέγεθος κάθε OR από το άθροισμα των συχνοτήτων των διαζευκτέων του. Μπορούμε έπειτα να επεξεργαστούμε το ερώτημα κατά αύξουσα σειρά μεγέθους κάθε διαζευκτικού όρου.

Για γενικά και αυθαίρετα ερωτήματα Boole, χρειάζεται να υπολογίσουμε και να αποθηκεύσουμε προσωρινά τις απαντήσεις των ενδιαμέσων εκφράσεων μιας περίπλοκης παράστασης. Όμως, σε αρκετές περιπτώσεις, είτε λόγω της φύσης της γλώσσας ερωτημάτων είτε απλώς επειδή είναι ο πιο συνηθισμένος τύπος ερωτημάτων που υποβάλλουν οι χρήστες, τα ερωτήματα είναι καθαρά συζευκτικά. Σε αυτή την περίπτωση, αντί να θεωρούμε τη συγχώνευση των λιστών καταχωρίσεων ως μια συνάρτηση με δύο εισόδους και μια διακριτή έξοδο, είναι αποδοτικότερο να υπολογίζουμε την τομή κάθε ανακτημένης λίστας καταχωρίσεων με το τρέχον ενδιάμεσο αποτέλεσμα στη μνήμη, παίρνοντας ως αρχική τιμή του ενδιάμεσου αποτελέσματος τη λίστα καταχωρίσεων του λιγότερου συχνού όρου. Αυτός ο αλγόριθμος παρουσιάζεται στο Σχήμα 1.7. Η πράξη της τομής είναι τότε ασύμμετρη: η λίστα ενδιάμεσων αποτελεσμάτων βρίσκεται στη μνήμη ενώ η λίστα με την οποία πρέπει να συνδυαστεί ώστε να υπολογιστεί η τομή τους διαβάζεται από το δίσκο. Επιπλέον, η λίστα ενδιάμεσων αποτελεσμάτων είναι πάντα τουλάχιστον τόσο μικρή όσο και η άλλη, ενώ σε πολλές περιπτώσεις είναι μικρότερη κατά αρκετές τάξεις μεγέθους. Η τομή των καταχωρίσεων θα μπορούσε πάλι να βρεθεί με τη χρήση του αλγορίθμου του Σχήματος 1.6, αλλά όταν τα μήκη των λιστών διαφέρουν πολύ μεταξύ τους έχουμε την ευκαιρία να χρησιμοποιήσουμε

```
INTERSECT(⟨t_1, ..., t_n⟩)
1  όροι ← SORTBYINCREASINGFREQUENCY(⟨t_1, ..., t_n⟩)
2  αποτέλεσμα ← καταχωρίσεις(πρώτος(όροι))
3  όροι ← υπόλοιποι(όροι)
4  while όροι ≠ NIL and αποτέλεσμα ≠ NIL
5  do αποτέλεσμα ← INTERSECT(αποτέλεσμα, καταχωρίσεις(πρώτος(όροι)))
6     όροι ← υπόλοιποι(όροι)
7  return αποτέλεσμα
```

Σχήμα 1.7 Αλγόριθμος για συζευκτικά ερωτήματα, ο οποίος επιστρέφει το σύνολο των εγγράφων που περιέχουν κάθε όρο της λίστας όρων εισόδου.

και άλλες τεχνικές. Η τομή μπορεί να υπολογιστεί επί τόπου, με τη μόνιμη τροποποίηση ή την επισήμανση των μη έγκυρων στοιχείων στη λίστα ενδιάμεσων αποτελεσμάτων. Διαφορετικά, η τομή μπορεί να υπολογιστεί με την εκτέλεση μιας ακολουθίας δυαδικών αναζητήσεων στις μεγάλες λίστες καταχωρίσεων για κάθε καταχώριση στη λίστα ενδιάμεσων αποτελεσμάτων. Άλλη μια δυνατότητα είναι η αποθήκευση των μακροσκελών λιστών καταχωρίσεων σε μορφή πίνακα κατακερματισμού (hashtable), ώστε ο έλεγχος για το αν ανήκει σε αυτές κάποιο στοιχείο των ενδιάμεσων αποτελεσμάτων να πραγματοποιείται σε σταθερό και όχι σε γραμμικό ή λογαριθμικό χρόνο. Τέτοιες εναλλακτικές τεχνικές, όμως, δύσκολα συνδυάζονται με τη συμπίεση των λιστών καταχωρίσεων, την οποία περιγράφουμε στο Κεφάλαιο 5. Επιπλέον, οι συνηθισμένες πράξεις υπολογισμού τομής των λιστών καταχωρίσεων εξακολουθούν να είναι απαραίτητες όταν δύο όροι ενός ερωτήματος είναι πολύ συνηθισμένοι.

? **Άσκηση 1.4** [⋆] Στα παρακάτω ερωτήματα, εξακολουθούμε να μπορούμε να υπολογίσουμε την τομή σε χρόνο $O(x + y)$, όπου x και y είναι τα μήκη των λιστών καταχωρίσεων για τα ονόματα Brutus και Caesar; Αν όχι, τότε τι μπορούμε να καταφέρουμε;

α. Brutus AND NOT Caesar

β. Brutus OR NOT Caesar

Άσκηση 1.5 [⋆] Επεκτείνετε τον αλγόριθμο συγχώνευσης καταχωρίσεων για αυθαίρετες παραστάσεις ερωτημάτων Boole. Ποια είναι η χρονική του πολυπλοκότητα; Για παράδειγμα, θεωρήστε:

γ. (Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

Μπορούμε πάντα να επιτυγχάνουμε τη συγχώνευση σε γραμμικό χρόνο; Γραμμικό ως προς τι; Μπορούμε να τα καταφέρουμε καλύτερα;

Άσκηση 1.6 [⋆⋆] Μπορούμε να αλλάζουμε τη διατύπωση των ερωτημάτων χρησιμοποιώντας τους επιμεριστικούς κανόνες για τους τελεστές AND και OR.

α. Δείξτε πώς μπορούμε να αναδιατυπώσουμε το ερώτημα της Άσκησης 1.5 σε κανονική διαζευκτική μορφή χρησιμοποιώντας τους επιμεριστικούς κανόνες.

β. Το ερώτημα που προέκυψε μπορούμε να το επεξεργαστούμε αποδοτικότερα από το αρχικό;

γ. Αυτό το συμπέρασμα αληθεύει γενικά ή εξαρτάται από τις λέξεις και τα περιεχόμενα της συλλογής εγγράφων;

Άσκηση 1.7 [⋆] Προτείνετε μια κατάλληλη αλληλουχία επεξεργασίας για το ερώτημα

δ. (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)

δεδομένων των παρακάτω μεγεθών για τις λίστες καταχωρίσεων:

Όρος	**Μέγεθος καταχωρίσεων**
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

Άσκηση 1.8 [⋆] Αν έχουμε το ερώτημα:

ε. friends AND romans AND (NOT countrymen)

πώς θα μπορούσαμε να χρησιμοποιήσουμε τη συχνότητα της λέξης countrymen για να εκτιμήσουμε τη βέλτιστη σειρά υπολογισμού του ερωτήματος; Συγκεκριμένα, προτείνετε έναν τρόπο χειρισμού της άρνησης για τον προσδιορισμό της σειράς επεξεργασίας του ερωτήματος.

Άσκηση 1.9 [⋆⋆] Στα συζευκτικά ερωτήματα, η επεξεργασία των λιστών καταχωρίσεων κατά σειρά μεγέθους είναι εγγυημένα η βέλτιστη; Εξηγήστε γιατί είναι, ή δώστε ένα παράδειγμα όπου δεν είναι έτσι.

Άσκηση 1.10 [⋆⋆] Γράψτε έναν αλγόριθμο συγχώνευσης καταχωρίσεων, όπως αυτός του Σχήματος 1.6 (σελίδα 32), για ένα ερώτημα *x* OR *y*.

Άσκηση 1.11 [⋆⋆] Πώς θα έπρεπε να γίνεται ο χειρισμός του ερωτήματος Boole *x* AND NOT *y*; Γιατί η απλοϊκή αποτίμηση (επεξεργασία) αυτού του ερωτήματος θα ήταν πολύ δαπανηρή; Γράψτε έναν αλγόριθμο συγχώνευσης καταχωρίσεων για την αποδοτική αποτίμηση του ερωτήματος.

1.4 Το επεκτεταμένο μοντέλο Boole σε σύγκριση με την καταταγμένη ανάκτηση

ΜΟΝΤΕΛΑ ΚΑΤΑΤΑΓΜΕΝΗΣ ΑΝΑΚΤΗΣΗΣ

ΕΡΩΤΗΜΑΤΑ ΕΛΕΥΘΕΡΟΥ ΚΕΙΜΕΝΟΥ

Το μοντέλο ανάκτησης Boole έρχεται σε αντίθεση με τα *μοντέλα καταταγμένης ανάκτησης* (ranked retrieval) όπως το διανυσματικό μοντέλο (Ενότητα 6.3), όπου οι χρήστες χρησιμοποιούν κυρίως *ερωτήματα ελεύθερου κειμένου*, δηλαδή, απλώς πληκτρολογούν μία ή περισσότερες λέξεις αντί να χρησιμοποιούν κάποια συγκεκριμένη γλώσσα με τελεστές για τη δημιουργία παραστάσεων ερωτημάτων, και το σύστημα αποφασίζει ποια έγγραφα ικανοποιούν καλύτερα το ερώτημα. Παρά τις δεκαετίες ακαδημαϊκών ερευνών γύρω από τα πλεονεκτήματα

της καταταγμένης ανάκτησης, τα συστήματα που υλοποιούσαν το μοντέλο ανάκτησης Boole αποτελούσαν την κύρια ή και τη μόνη επιλογή αναζήτησης που παρείχαν οι μεγάλοι εμπορικοί πάροχοι πληροφοριών για τρεις δεκαετίες, μέχρι τις αρχές της δεκαετίας του 1990 (δηλαδή, περίπου μέχρι την έλευση του Παγκόσμιου Ιστού). Ωστόσο, αυτά τα συστήματα δεν διέθεταν μόνο τις βασικές πράξεις Boole (AND, OR, και NOT) που παρουσιάσαμε ως αυτό το σημείο. Η εφαρμογή αυστηρών παραστάσεων Boole στους όρους των ερωτημάτων και η επιστροφή αποτελεσμάτων χωρίς κάποια κατάταξη είναι υπερβολικοί περιορισμοί για πολλές από τις ανάγκες πληροφόρησης των χρηστών, γι' αυτό και εκείνα τα συστήματα υλοποιούσαν επεκτεταμένα μοντέλα ανάκτησης Boole, ενσωματώνοντας πρόσθετους τελεστές, όπως τελεστές εγγύτητας όρων. Ένας *τελεστής εγγύτητας* (proximity operator) μας επιτρέπει να καθορίσουμε ότι δύο όροι του ερωτήματος πρέπει να βρίσκονται κοντά ο ένας στον άλλον σε κάποιο έγγραφο, όπου το «κοντά» (η εγγύτητα) μπορεί να ορίζεται ως περιορισμός του πλήθους των λέξεων που παρεμβάλλονται μεταξύ των όρων ή ως μια δομική μονάδα κειμένου, όπως μια πρόταση ή παράγραφος.

ΤΕΛΕΣΤΗΣ ΕΓΓΥΤΗΤΑΣ

Παράδειγμα 1.1: Εμπορική μηχανή αναζήτησης Boole: Westlaw. Η Westlaw (http://www.westlaw.com/) είναι ο μεγαλύτερος εμπορικός πάροχος υπηρεσιών αναζήτησης σε νομικά κείμενα (βάσει του πλήθους των καταγεγραμμένων συνδρομητών της), με περισσότερους από ένα εκατομμύριο συνδρομητές που πραγματοποιούν εκατομμύρια αναζητήσεις κάθε ημέρα, σε δεκάδες terabyte δεδομένων κειμένου. Η υπηρεσία αυτή ξεκίνησε το 1975. Το 2005, η αναζήτηση Boole (που ονομαζόταν *Terms and Connectors* — Όροι και συνδέσεις — στη Westlaw) εξακολουθούσε να είναι η προεπιλεγμένη και να χρησιμοποιείται από μεγάλο ποσοστό των χρηστών, παρόλο που η δυνατότητα υποβολής ερωτημάτων ελεύθερου κειμένου και καταταγμένης αναζήτησης (με επωνυμία *Natural Language* — Φυσική γλώσσα) είχε προστεθεί από το 1992. Ακολουθούν μερικά παραδείγματα ερωτημάτων Boole στη Westlaw:

Ανάγκη πληροφόρησης: Πληροφορίες για τις νομικές θεωρίες που αφορούν την απαγόρευση γνωστοποίησης (disclosure prevention) εμπορικών μυστικών (trade secrets) από υπαλλήλους (employees) οι οποίοι εργάζονταν παλαιότερα σε ανταγωνιστικές εταιρείες.
Ερώτημα: "trade secret" /s disclos! /s prevent /s employe!

Ανάγκη πληροφόρησης: Απαιτήσεις που πρέπει να ικανοποιούνται ώστε άτομα με αναπηρία (disabled) να έχουν πρόσβαση (access) στο χώρο εργασίας τους (work site, work place).
Ερώτημα: disab! /p access! /s work-site work-place (employment /3 place)

Ανάγκη πληροφόρησης: Δεδικασμένα σχετικά με τις ευθύνες (liabilities, responsibilities) του οικοδεσπότη (host) όταν οι καλεσμένοι του (guests) είναι σε κατάσταση μέθης (drunk, intoxicated).
Ερώτημα: host! /p (responsib! liab!) /p (intoxicat! drunk!) /p guest

Προσέξτε τα μακροσκελή, επακριβή ερωτήματα και τη χρήση τελεστών εγγύτητας, χαρακτηριστικά ασυνήθιστα και τα δύο στις αναζητήσεις σε περιβάλλον ιστού. Τα υποβαλλόμενα ερωτήματα έχουν κατά μέσο όρο μήκος δέκα λέξεων. Σε αντίθεση με τις συμβάσεις αναζήτησης στον Ιστό, ένα κενό διάστημα μεταξύ των λέξεων αντιπροσωπεύει διάζευξη (τον πλέον περιοριστικό τελεστή), το & σημαίνει AND και τα /s, /p, και /*k* σημαίνουν ταίριασμα στην ίδια πρόταση, στην ίδια παράγραφο ή εντός *k* λέξεων αντίστοιχα. Τα διπλά εισαγωγικά σημαίνουν *αναζήτηση φράσεων* (διαδοχικών λέξεων)· δείτε την Ενότητα 2.4 (σελίδα 63). Το θαυμαστικό (!) σηματοδοτεί μια αναζήτηση με χαρακτήρα μπαλαντέρ τέλους (δείτε την Ενότητα 3.2, σελίδα 76)· δηλαδή, ο όρος liab! εντοπίζει όλες τις λέξεις που αρχίζουν από liab. Επιπλέον, ο όρος work-site εντοπίζει οποιοδήποτε από τα *worksite*, *work-site*, ή *work site*· δείτε την Ενότητα 2.2.1. Τα περισσότερα εξειδικευμένα ερωτήματα αυτού του τύπου συνήθως είναι προσεκτικά ορισμένα και έχουν αναπτυχθεί βηματικά, με σταδιακές βελτιώσεις, ώσπου να επιστρέφουν τα αποτελέσματα που θεωρεί σωστά ο χρήστης.

Πολλοί χρήστες, ιδιαίτερα οι επαγγελματίες, προτιμούν τα μοντέλα ερωτημάτων Boole. Τα ερωτήματα Boole είναι ακριβή: ένα έγγραφο είτε ικανοποιεί το ερώτημα είτε όχι. Αυτό παρέχει στο χρήστη περισσότερο έλεγχο και διαφάνεια στα στοιχεία που ανακτώνται. Και μερικοί τομείς, όπως αυτός των νομικών θεμάτων, επιτρέπουν ένα αποτελεσματικό μέσο βαθμολόγησης των εγγράφων στο μοντέλο Boole: η Westlaw επιστρέφει έγγραφα σε αντίστροφη χρονολογική σειρά, κάτι που πρακτικά είναι ιδιαίτερα αποτελεσματικό. Το 2007, η πλειοψηφία των βιβλιοθηκονόμων νομικών βιβλιοθηκών εξακολουθούσαν να συνιστούν την Terms and Connectors για αναζητήσεις υψηλής ανάκλησης, ενώ και η πλειοψηφία των υπόλοιπων χρηστών θεωρούσαν ότι έχουν μεγαλύτερο έλεγχο με τη χρήση αυτού του μοντέλου. Όμως, αυτό δεν σημαίνει ότι τα ερωτήματα Boole θεωρούνται αποτελεσματικότερα από όλους όσοι χρησιμοποιούν την αναζήτηση για επαγγελματικούς λόγους. Πράγματι, πειραματιζόμενος με μια επιμέρους συλλογή της Westlaw, ο Turtle (1994) διαπίστωσε ότι τα ερωτήματα ελεύθερου κειμένου παρείχαν καλύτερα αποτελέσματα για την πλειοψηφία των αναγκών πληροφόρησης των πειραμάτων του, ακόμα και από τα ερωτήματα Boole που κατασκεύαζαν οι ίδιοι οι βιβλιοθηκονόμοι της Westlaw. Ένα γενικό πρόβλημα της αναζήτησης Boole είναι ότι η χρήση τελεστών AND συνήθως παράγει αναζητήσεις υψηλής ακρίβειας αλλά χαμηλής ανάκλησης ενώ η χρήση τελεστών OR οδηγεί σε αναζητήσεις χαμηλής ακρίβειας αλλά υψηλής ανάκλησης, και είναι δύσκολο ως και αδύνατο να βρεθεί μια ικανοποιητική μέση λύση.

Σε αυτό το κεφάλαιο εξετάσαμε τη δομή και την κατασκευή απλών αντεστραμμένων ευρετηρίων που αποτελούνται από ένα λεξικό και λίστες καταχωρίσεων. Παρουσιάσαμε το μοντέλο ανάκτησης Boole και εξετάσαμε πώς μπορούμε να πραγματοποιούμε αποδοτικές ανακτήσεις μέσω συγχωνεύσεων σε γραμμικό χρόνο και απλών βελτιστοποιήσεων των ερωτημάτων. Στα Κεφάλαια 2–7, εξετάζουμε λεπτομερώς άλλα μοντέλα ερωτημάτων, πλουσιότερα σε δυνατότητες,

καθώς και τις ενισχυμένες δομές ευρετηρίου που απαιτούνται για τον αποδοτικό χειρισμό τους. Εδώ αναφέρουμε απλώς μερικές από τις κυριότερες πρόσθετες δυνατότητες που θα θέλαμε να υλοποιήσουμε.

1. Θα θέλαμε να προσδιορίζουμε αποτελεσματικότερα του όρους του λεξικού και να παρέχουμε λειτουργίες ανάκτησης ανεκτικές σε ορθογραφικά σφάλματα και ακατάλληλες επιλογές λέξεων.
2. Συχνά είναι χρήσιμο να αναζητάμε σύνθετες λέξεις ή φράσεις που καταδεικνύουν έναν συγκεκριμένο όρο, π.χ. «λειτουργικό σύστημα». Όπως είδαμε και στα παραδείγματα για τη Westlaw, συχνά μας ενδιαφέρουν και ερωτήματα εγγύτητας, όπως το Gates NEAR Microsoft. Για να απαντήσουμε τέτοια ερωτήματα, το ευρετήριο πρέπει να επεκταθεί ώστε να περιλαμβάνει και δεδομένα εγγύτητας μεταξύ των όρων στα έγγραφα.
3. Το μοντέλο Boole καταγράφει μόνο την παρουσία ή την απουσία όρων, εμάς όμως συχνά μας ενδιαφέρει συσσώρευση πειστηρίων ώστε να δώσουμε μεγαλύτερη βαρύτητα στα έγγραφα που περιέχουν έναν όρο πολλές φορές παρά σε αυτά που περιέχουν τον όρο μόνο μία φορά. Για να το καταφέρουμε αυτό, χρειαζόμαστε πληροφορίες *συχνότητας όρων* στις λίστες καταχωρίσεων (πόσες φορές ένας όρος περιλαμβάνεται σε ένα έγγραφο).

ΣΥΧΝΟΤΗΤΑ ΟΡΟΥ

4. Τα ερωτήματα Boole απλώς ανακτούν το σύνολο των εγγράφων που ταιριάζουν στο ερώτημά μας, αυτό όμως που συνήθως θέλουμε είναι και μια αποτελεσματική μέθοδος διάταξης (ή *κατάταξης*) των αποτελεσμάτων. Αυτό απαιτεί κάποιο μηχανισμό για τον προσδιορισμό μιας βαθμολογίας για κάθε έγγραφο, η οποία θα εκφράζει πόσο καλά το συγκεκριμένο έγγραφο ικανοποιεί το ερώτημά μας.

Αφού εξετάσουμε και αυτές τις δυνατότητες, θα έχουμε γνωρίσει το μεγαλύτερο μέρος της βασικής τεχνολογίας για την υποστήριξη περιστασιακών αναζητήσεων σε αδόμητες πληροφορίες. Η περιστασιακή αναζήτηση σε έγγραφα έχει πλέον κυριαρχήσει, αποτελώντας την κινητήρια δύναμη όχι μόνο για τις μηχανές αναζήτησης του Ιστού αλλά και για το είδος της αδόμητης αναζήτησης που αποτελεί τη βάση των μεγαλύτερων ιστότοπων ηλεκτρονικού εμπορίου. Αν και οι κύριες μηχανές αναζήτησης του Ιστού διαφέρουν δίνοντας έμφαση στα ερωτήματα ελεύθερου κειμένου, το μεγαλύτερο μέρος των τεχνολογιών ευρετηρίασης και επεξεργασίας ερωτημάτων και των ζητημάτων που καλούμαστε να αντιμετωπίσουμε στις δύο αυτές μορφές ανάκτησης είναι κοινό, όπως θα δούμε σε επόμενα κεφάλαια. Επιπλέον, με την πάροδο του χρόνου, οι μηχανές αναζήτησης του Ιστού έχουν προσθέσει στις δυνατότητές τους έστω και μερική υποστήριξη για τους δημοφιλέστερους τελεστές των επεκτεταμένων μοντέλων Boole: η αναζήτηση φράσεων είναι ιδιαίτερα δημοφιλής και οι περισσότερες μηχανές παρέχουν κάποια περιορισμένη υλοποίηση των τελεστών Boole. Μολαταύτα, παρόλο που αυτές οι επιλογές προτιμώνται από τους πιο έμπειρους αναζητητές, οι περισσότεροι άνθρωποι τις χρησιμοποιούν ελάχιστα και δεν βρίσκονται πλέον στο επίκεντρο των προσπαθειών που διενεργούνται για τη βελτίωση των επιδόσεων των μηχανών αναζήτησης.

? **Άσκηση 1.12** [⋆] Γράψτε ένα ερώτημα, χρησιμοποιώντας το συντακτικό της Westlaw, για την εύρεση οποιασδήποτε από τις λέξεις professor (καθηγητής), teacher (δάσκαλος), ή lecturer (λέκτορας), σε προτάσεις που περιέχουν και κάποια μορφή του ρήματος explain (εξηγώ).

Άσκηση 1.13 [⋆] Προσπαθήστε να χρησιμοποιήσετε τις δυνατότητες αναζήτησης Boole σε μερικές γνωστές μηχανές αναζήτησης. Για παράδειγμα, διαλέξτε μια λέξη, όπως burglar (διαρρήκτης), και υποβάλετε τα ερωτήματα (α) burglar, (β) burglar AND burglar, και (γ) burglar OR burglar. Εξετάστε το εκτιμώμενο πλήθος των αποτελεσμάτων και τα αποτελέσματα που επεστράφησαν στην κορυφή της λίστας. Έχουν νόημα στα πλαίσια της λογικής Boole; Συχνά δεν έχουν στις δημοφιλείς μηχανές αναζήτησης. Μπορείτε να καταλάβετε τι συμβαίνει; Αν δοκιμάζατε διαφορετικές λέξεις; Για παράδειγμα, δοκιμάστε τα ερωτήματα (α) knight, (β) conquer, και έπειτα (γ) knight OR conquer. Ποιος περιορισμός θα περιμένατε να ισχύει για το τρίτο ερώτημα, με βάση το πλήθος των αποτελεσμάτων των δύο πρώτων ερωτημάτων; Βλέπετε να ισχύει αυτός ο περιορισμός;

1.5 Βιβλιογραφία και περαιτέρω μελέτη

Η ενασχόληση με την ανάκτηση πληροφοριών μέσω υπολογιστή ξεκίνησε στα τέλη της δεκαετίας του 1940 (Cleverdon 1991· Liddy 2005). Η σημαντική αύξηση στην παραγωγή επιστημονικών συγγραμμάτων, περισσότερο στη μορφή τεχνικών εκθέσεων παρά ως επίσημα άρθρα σε επιστημονικά περιοδικά, σε συνδυασμό με τη διαθεσιμότητα των υπολογιστών, προκάλεσε το ενδιαφέρον για την αυτόματη ανάκτηση εγγράφων. Εκείνη την εποχή, όμως, η ανάκτηση εγγράφων βασιζόταν σε στοιχεία όπως όνομα συγγραφέα, τίτλος, και λέξεις-κλειδιά· η αναζήτηση σε όλη την έκταση του κειμένου έγινε διαθέσιμη πολύ αργότερα.

Το άρθρο του Bush (1945) αποτέλεσε σημαντική πηγή έμπνευσης για το νέο επιστημονικό πεδίο:

> Φανταστείτε ένα μελλοντικό μηχάνημα ατομικής χρήσης, ένα είδος αυτοματοποιημένου προσωπικού αρχείου και βιβλιοθήκης. Χρειάζεται ένα όνομα και, επινοώντας ένα στην τύχη, ας το ονομάσουμε "memex". Το memex είναι μια συσκευή στην οποία ο ιδιοκτήτης της αποθηκεύει όλα του τα βιβλία, τα αρχεία του, και στοιχεία επικοινωνίας, και η οποία είναι αυτοματοποιημένη με τέτοιον τρόπο ώστε να μπορεί να ερευνηθεί με εξαιρετική ταχύτητα και ευελιξία. Πρόκειται για ένα διευρυμένο προσωπικό συμπλήρωμα της μνήμης του.

Ο όρος *ανάκτηση πληροφοριών* (information retrieval) επινοήθηκε από τον Calvin Mooers το 1948/1950 (Mooers 1950).

Το 1958, η προσοχή του τύπου στράφηκε στα στοιχεία που παρουσιάστηκαν σε ένα συνέδριο για τις μηχανές «αυτόματης ευρετηρίασης» της IBM (δείτε Taube & Wooster 1958), τα οποία βασίζονταν κυρίως στην εργασία τού H. P. Luhn. Το εμπορικό ενδιαφέρον προσανατολίστηκε γρήγορα προς τα συστήματα ανάκτησης Boole, αλλά εκείνα τα πρώτα χρόνια παρουσιάστηκαν και έντονες αντι-

Το βιβλίο *Εισαγωγή στην ανάκτηση πληροφοριών* είναι το πρώτο διδακτικό σύγγραμμα που παρουσιάζει με συνοχή και συνέπεια τόσο την κλασική ανάκτηση πληροφοριών όσο και την ανάκτηση πληροφοριών σε διαδικτυακό περιβάλλον, συμπεριλαμβανομένης της αναζήτησης στον Ιστό και των συγγενών τομέων της ταξινόμησης και της συσταδοποίησης κειμένων. Γραμμένο από τη σκοπιά της Επιστήμης των Υπολογιστών, παρέχει μια σύγχρονη θεώρηση όλων των πλευρών της σχεδίασης και της υλοποίησης συστημάτων για τη συγκέντρωση εγγράφων, την ευρετηρίασή τους και την πραγματοποίηση αναζητήσεων στα κείμενα που περιέχουν, καθώς και των μεθόδων αξιολόγησης αυτών των συστημάτων, εισάγοντας παράλληλα τους αναγνώστες στη χρήση μεθόδων μηχανικής μάθησης σε συλλογές κειμένων.

Σχεδιασμένο ώστε να αποτελέσει το κυρίως διδακτικό σύγγραμμα σε ακαδημαϊκά προγράμματα μεταπτυχιακού ή προχωρημένου προπτυχιακού επιπέδου, το βιβλίο παρουσιάζει επίσης ενδιαφέρον για τους ερευνητές και τους επαγγελματίες του κλάδου. Επιπλέον, οι συγγραφείς έχουν δημοσιεύσει στον Ιστό ένα πλήρες σύνολο συνοδευτικών διαφανειών και ασκήσεων (στα Αγγλικά).

Ο **Christopher D. Manning** είναι Αναπληρωτής Καθηγητής της Επιστήμης των Υπολογιστών και Γλωσσολογίας στο Πανεπιστήμιο Stanford.

Ο **Prabhakar Raghavan** είναι Επικεφαλής του Τμήματος Έρευνας της Yahoo! και Σύμβουλος Καθηγητής της Επιστήμης των Υπολογιστών στο Πανεπιστήμιο Stanford.

Ο **Hinrich Schütze** κατέχει την έδρα Θεωρητικής Υπολογιστικής Γλωσσολογίας στο Ινστιτούτο Επεξεργασίας Φυσικής Γλώσσας του Πανεπιστημίου της Στουτγάρδης.

"Πρόκειται για το πρώτο βιβλίο που παρέχει μια πλήρη εικόνα των περιπλοκών που παρουσιάζονται στην κατασκευή μιας σύγχρονης μηχανής αναζήτησης σε κλίμακα Ιστού. Θα γνωρίσετε τις SVM κατάταξης, την ανάκτηση σε δομημένα έγγραφα XML, την ανάλυση DNS, και τη λανθάνουσα σημασιολογική ευρετηρίαση (LSI). Θα ανακαλύψετε τον υπόκοσμο του παραπλανητικού περιεχομένου, της απόκρυψης, και των σελίδων διόδου. Θα διαπιστώσετε πώς η MapReduce και άλλες προσεγγίσεις παράλληλης επεξεργασίας μάς επιτρέπουν να προχωρήσουμε πέρα από τα megabyte και να χειριζόμαστε με αποδοτικό τρόπο δεδομένα της τάξης των petabyte."

–PETER NORVIG, Διευθυντής Ερευνών, Google Inc.

"Το *Εισαγωγή στην Ανάκτηση Πληροφοριών* είναι ένα περιεκτικό, ενημερωμένο, και καλογραμμένο εισαγωγικό βιβλίο για μια ταχύτατα αναπτυσσόμενη περιοχή της επιστήμης των υπολογιστών, της οποίας η σημασία αυξάνεται διαρκώς. Επιτέλους, ένα υψηλής ποιότητας σύγγραμμα για έναν επιστημονικό τομέα που χρειαζόταν επειγόντως ένα τέτοιο βιβλίο."

–RAYMOND J. MOONEY, Καθηγητής Επιστημών Υπολογιστών, Πανεπιστήμιο Τέξας, Ώστιν

"Μέσω της καλογραμμένης παρουσίασης και της εξαιρετικής επιλογής θεμάτων, οι συγγραφείς μεταδίδουν με ζωντάνια τόσο τις θεμελιώδεις έννοιες όσο και την ταχύτατα αυξανόμενη επιρροή του πεδίου της ανάκτησης πληροφοριών."

–JON KLEINBERG, Καθηγητής Επιστήμης Υπολογιστών, Πανεπιστήμιο Cornell

εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ

Δαμοκού 4, Σταθμός Λαρίσης, 10440 ΑΘΗΝΑ, Τηλ. 210-5237635
www.klidarithmos.gr info@klidarithmos.gr
www.facebook.com/klidarithmos.gr

ISBN 978-960-461-456-1