Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων (ΤΜΗΥΤΔ)

Π Τ Υ Χ Ι Α Κ Η Ε Ρ Γ Α Σ Ι Α

" Υ λ ο π ο ί η σ η α λ γ ο ρ ί θ μ ο υ Α ν ά κ λ η σ η ς Π λ η ρ ο φ ο ρ ί α ς σ τ ο δ ι α δ ί κ τ υ ο "

ΤΡΙΑΝΤΟΠΟΥΛΟΣ ΕΥΣΤΑΘΙΟΣ

ΕΙΣΗΓΗΤΡΙΑ : Δασκαλοπούλου Ασπασία

Επίκουρος Καθηγητής

ΒΟΛΟΣ 2009

Ευχαριστίες

Πριν ξεκινήσω με αυτή την μελέτη, θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια μου κ.Ασπασία Δασκαλοπούλου για την ουσιαστική και μεγάλη βοήθεια που μου έδωσε στο διάστημα της διεξαγωγής της διπλωματικής μου, καθώς επίσης και τον δεύτερο επιβλέποντα κ.Δημήτριο Κατσαρό για τον πολύτιμο χρόνο που διέθεσε.

Τέλος θα ήθελα να ευχαριστήσω όλους εκείνους που με στήριξαν για την διεξαγωγή αυτής της έργασίας.

Περιεχόμενα

Εισαγωγή

Στην καθημερινή συνομιλία ανάμεσα σε δυο ανθρώπους, συχνά είναι γνωστό και στους δυο το πλαίσιο εντός του οποίου διεξάγεται η συνομιλία, τα λεγόμενα συμφραζόμενα, και επομένως μπορεί η αναφορά τους να είναι ελλειπτική. Έτσι μπορούμε να πούμε ότι σε κάθε γεγονός υπάρχει ένα σύνολο από γεγονότα πού είναι κάποιες παρελθοντικές πληροφορίες οι οποίες μας βοηθούν να γνωρίζουμε σε ένα βαθμό καλυτέρα αυτό το γεγονός.

Στο διαδίκτυο όμως τα πράγματα είναι απρόσωπα και επομένως διαφορετικά. Κατά την αναζήτηση πληροφοριών στο διαδίκτυο, πληκτρολογούμε κάποια ερωτήματα στον browser και παίρνουμε σαν αποτελέσματα πληροφορίες σχετικά με τα ερωτήματα που του ζητήσαμε. Όμως αυτές οι πληροφορίες δεν ανταποκρίνονται πάντα σε αυτό που ψάχνουμε, αλλά αντίθετα μερικές φορές είναι τελείως ανούσιες για εμάς. Ας πάρουμε για παράδειγμα την περίπτωση όπου ένας χρήστης πληκτρολογεί το ερώτημα ΄΄στρώματα΄΄ και ο browser του επιστρέφει πληροφορία που αφορά στρώματα ύπνου, ενώ ο χρήστης ενδιαφερόταν για στρώματα θαλάσσης. Αυτό συμβαίνει επειδή ο browser δεν λαμβάνει υπόψη τις προτιμήσεις του χρήστη. Λαμβάνοντας υπόψη τα παραπάνω, εμείς εξετάζουμε το ενδεχόμενο η πληροφορία που θα επιστρέφεται στον χρήστη να μην αναφέρεται μόνο στο ερώτημα του, αλλά να του προτείνει περεταίρω πληροφορία ΄΄συγγενική΄΄ προς αυτή που έδωσε και να σχετίζεται με το ιστορικό του το οποίο υπάρχει στον browser. Άρα, όσον αφορά το παράδειγμα που δώσαμε παραπάνω, γνωρίζοντας ο browser ότι ο χρήστης ενδιαφέρεται για θαλασσιά προϊόντα αφού υπάρχουν στο ιστορικό του browser αναφορές σε αυτά ,να του προταθεί σαν επιλογή και τα θαλάσσια στρώματα εκτός των άλλων στρωμάτων. Έτσι λοιπόν, σκοπός της εργασίας αυτής είναι η προσπάθεια ανάκτησης πληροφορίας μέσα από το διαδίκτυο. Παρακάτω θα μελετήσουμε έναν αλγόριθμο που αναφέρεται σε αυτό το πρόβλημα.

Άρα στο **πρώτο κεφάλαιο** κάνουμε μια εισαγωγή στην ανάκτηση πληροφορίας, σε κάποιες βασικές έννοιες της (διαδικασία χρήστη, λογική αναπαράσταση των κειμένων,) και αναφέρουμε την διαδικασία ανάκτησης δεδομένων.

Στο **δεύτερο κεφάλαιο** αναφερόμαστε σε βασικά μοντέλα ανάκλησης (το Boolean Μοντέλο, το Vector Space Μοντέλο και το Πιθανοτικό Μοντέλο), και πως μας βοήθησαν αυτά για να κάνουμε την υλοποίησή του επόμενου κεφαλαίου.

Στο **τρίτο κεφάλαιο** αναφέρουμε την βασική υλοποίηση και ιδέα ενός αλγορίθμου ανάκτησης όσον αφορά την αναζήτηση πληροφοριών στο διαδίκτυο, όπως επίσης και ποια είναι η χρησιμότητά του στο χρήστη του διαδικτύου.

Τέλος στο **τέταρτο και τελευταίο κεφάλαιο** γράφουμε τα συμπεράσματα αυτής της μελέτης και αναφέρουμε σκέψεις για μελλοντική μελέτη επάνω σε κάποια ζητήματα που αφορούν περαιτέρω την υλοποίηση.

Κεφάλαιο 1ο

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΑΝΑΚΛΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ

1.1 Η ανάγκη για Ανάκτηση Πληροφορίας

Η επιστήμη της ανάκτησης πληροφορίας, ασχολείται με την αναπαράσταση, την αποθήκευση, την οργάνωση και την πρόσβαση σε πληροφοριακά αντικείμενα. Η αναπαράσταση και η οργάνωση των πληροφοριακών αντικειμένων πρέπει να γίνονται με τρόπο, ώστε να παρέχουν στον εκάστοτε χρήστη, εύκολη πρόσβαση στην πληροφορία που τον ενδιαφέρει. Δυστυχώς ο καθορισμός της *πληροφοριακής ανάγκης* του χρήστη, είναι ένα δύσκολο πρόβλημα.

Το παραπάνω πρόβλημα αντικατοπτρίζεται, για παράδειγμα, στην έκφραση της ακόλουθης πληροφοριακής ανάγκης στο χώρο του Διαδικτύου.

> *Ανάκτησε όλες τις σελίδες που περιέχουν πληροφορίες για κινηματογραφικές ταινίες στις οποίες: (1) Πρωταγωνιστεί ο Κλίντ Ήστγουντ, (2) είναι γουέστερν, (3) υπάρχουν σε DVD.* Σελίδες σχετικές με το παραπάνω ερώτημα θα πρέπει να περιέχουν πληροφορίες, για τους συντελεστές της ταινίας, κριτικές καθώς και περίληψη του σεναρίου.

Είναι εμφανής η δυσκολία έκφρασης της παραπάνω πληροφοριακής ανάγκης με πληρότητα, χρησιμοποιώντας το περιβάλλον διεπαφής μιας *Διαδικτυακής Μηχανής Αναζήτησης*. Συνεπώς ο χρήστης πρέπει να είναι σε θέση να επαναδιατυπώσει την πληροφοριακή ανάγκη, σε μορφή *ερωτήματος* (query), το οποίο να μπορεί να γίνεται αντικείμενο επεξεργασίας από την μηχανή αναζήτησης (ή το σύστημα ΑΠ).

Η μετατροπή αυτή συνήθως γίνεται με τη χρήση, ενός συνόλου λέξεων κλειδιών (keywords) ή ισοδύναμα όρων δεικτοδότησης (index terms), που συνοψίζουν την περιγραφή της πληροφοριακής ανάγκης του χρήστη. Δοθέντος του ερωτήματος του χρήστη, το ζητούμενο από ένα σύστημα ΑΠ είναι να *ανακτήσει πληροφορία*, η οποία μπορεί να είναι χρήσιμη ή σχετική προς την πληροφοριακή ανάγκη. Έμφαση δίνεται στην ανάκτηση πληροφορίας σε αντίθεση με την ανάκτηση δεδομένων τη διαφορά των οποίων θα εξετάσουμε αμέσως.

1.1.1 Ανάκτηση Πληροφορίας και όχι Δεδομένων

Η ανάκτηση δεδομένων σε ένα περιβάλλον ανάκτησης πληροφορίας, συνίσταται στην εύρεση όλων των κειμένων τα οποία περιέχουν κάποιες από τις λέξεις κλειδιά που εμφανίζονται σε ένα ερώτημα προς το σύστημα. Αυτή η προσέγγιση δίνει συχνά κάτι διαφορετικό από αυτό που πραγματικά θέλει ο χρήστης. Στην πράξη, αυτό που περισσότερο ενδιαφέρει τον χρήστη ενός συστήματος ανάκτησης πληροφορίας, είναι να ανακτήσει πληροφορίες για ένα συγκεκριμένο θέμα, παρά η ανάκτηση δεδομένων σχετικών με κάποιο ερώτημα. Μια γλώσσα ανάκτησης δεδομένων, στοχεύει στην ανάκτηση όλων των αντικειμένων, που ικανοποιούν ένα σύνολο καλά ορισμένων συνθηκών, που διατυπώνονται με μια κανονική έκφραση ή με χρήση των εργαλείων της σχεσιακής άλγεβρας. Επίσης σε ένα σύστημα ανάκτησης δεδομένων (βλ. μια σχεσιακή βάση δεδομένων), τα δεδομένα είναι οργανωμένα σε μία καλά ορισμένη δομή και έχουν συγκεκριμένη σημασιολογία. Έτσι σε ένα σύστημα ανάκτησης δεδομένων, η ανάκτηση ενός και μόνο λανθασμένου αποτελέσματος, θεωρείται ένδειξη εσφαλμένης λειτουργίας του μηχανισμού ανάκτησης. Αντίθετα στα συστήματα ανάκτησης πληροφορίας, τα

ανακτώμενα αποτελέσματα μπορεί να είναι ανακριβή και η εμφάνιση κάποιων λαθών στα αποτελέσματα, περνά συχνά απαρατήρητη. Ο λόγος αυτής της διαφοροποίησης είναι ότι το σύστημα της ανάκτησης πληροφορίας, διαχειρίζεται κείμενα γραμμένα σε φυσική γλώσσα, τα οποία δεν είναι πάντα επαρκώς δομημένα και είναι συχνά αμφίσημα. Μην ξεχνάμε άλλωστε και την δυσκολία της διατύπωσης της ακριβούς πληροφοριακής ανάγκης με τη χρήση λέξεων κλειδιών.

Έτσι ενώ η ανάκτηση δεδομένων δίνει λύσεις στο χρήστη ενός συστήματος βάσης δεδομένων, δεν λύνει το πρόβλημα της ανάκτησης πληροφορίας, σχετικής με κάποιο θέμα. Για να μπορέσει ένα σύστημα ανάκτησης πληροφορίας να ανταποκριθεί στην πληροφοριακή ανάγκη του χρήστη, θα πρέπει να είναι σε θέση, να 'διερμηνεύσει' με κάποιον τρόπο το σημασιολογικό περιεχόμενο το αντικειμένων (κείμενα) που διαχειρίζεται, και να τα διατάξει σύμφωνα με το βαθμό σχετικότητάς τους προς το ερώτημα του χρήστη. Η διαδικασία της 'διερμηνείας' συνίσταται στην εξαγωγή συντακτικής και σημασιολογικής πληροφορίας από τα κείμενα, η οποία θα χρησιμοποιηθεί για να ανταποκριθεί το σύστημα στην πληροφοριακή ανάγκη του χρήστη. Το πρόβλημα δεν εντοπίζεται μόνο στην εξαγωγή της παραπάνω πληροφορίας. Επιπλέον θα πρέπει να είναι εφικτή η χρήση της εξαγόμενης πληροφορίας για να αποφασιστεί η σχετικότητα προς κάποιο ερώτημα. Ο κύριος στόχος άλλωστε ενός συστήματος ΑΠ, είναι να μπορεί να επιστρέψει όλα τα κείμενα που είναι σχετικά προς κάποιο ερώτημα, ανακτώντας παράλληλα και όσο το δυνατόν λιγότερα μη σχετικά κείμενα. Γι' αυτό το λόγο η έννοια της σχετικότητας, διαδραματίζει κυρίαρχο ρόλο στην ανάκτηση πληροφορίας.

1.1.2 Η Ανάκτηση Πληροφορίας στο κέντρο του ενδιαφέροντος

Η αρχική ανάγκη για ανάπτυξη της ανάκτησης πληροφορίας ήταν η αυτοματοποιημένη δεικτοδότηση κειμένων και η ανάπτυξη μεθόδων για την αναζήτηση χρήσιμων κειμένων σε μια συλλογή. Στις ημέρες μας η έρευνα έχει επεκταθεί σε πολλούς παραπάνω τομείς, συμπεριλαμβάνοντας την μοντελοποίηση, την ταξινόμηση και κατηγοριοποίηση κειμένων, την οπτικοποίηση δεδομένων, τις διεπαφές προς τον χρήστη μηχανές ψαξίματος στο Παγκόσμιο Ιστό, συστήματα φιλτραρίσματος πληροφορίας, συστήματα προσαρμοστικών υπερμέσων, συστήματα εκπαιδευτικού λογισμικού, Βιοπληροφορική. Η άποψη που επικρατούσε μέχρι στις αρχές τις δεκαετίας του 90, ήταν ότι η ανάκτηση πληροφορίας απευθυνόταν μόνο σε εφαρμογές βιβλιοθηκονομίας. Όλα τα παραπάνω άλλαξαν δραματικά τα τελευταία χρόνια και κυρίως μετά την έλευση του Παγκοσμίου Ιστού.

Ο Παγκόσμιος Ιστός γίνεται μια ολοένα και μεγαλύτερη παρακαταθήκη ανθρώπινης γνώσης, που επιτρέπει την ανταλλαγή πληροφορίας και ιδεών σε έκταση πολύ μεγαλύτερη από ότι είχαμε δει μέχρι τώρα. Η επιτυχία του Ιστού συνίσταται στην ευκολία που παρέχει στο χρήστη να δημιουργήσει τις δικές του Ιστοσελίδες, όντας έτσι ένα εύκολα προσβάσιμο και σχετικά φθηνό μέσο προσωπικής έκφρασης. Επιπλέον η ύπαρξη του Ιστού, θέτει νέους τρόπους επικοινωνίας επανορίζοντας τις έννοιες απόσταση και χρόνος. Τέλος οι τρέχουσες εξελίξεις στην ολοκλήρωση διαφορετικών υπηρεσιών γύρω από τον Ιστό, έχουν αλλάξει τον τρόπο που ο άνθρωπος βλέπει τον υπολογιστή. Έννοιες όπως Ηλεκτρονικό Εμπόριο και Ψηφιακές Βιβλιοθήκες είναι δημοφιλείς και δημιουργούν νέες και πολλά υποσχόμενες αγορές.

Παρά την επιτυχημένη διάδοση του Παγκοσμίου Ιστού, η εύρεση χρήσιμης πληροφορίας στις Ιστοσελίδες, γίνεται μια ολοένα και πιο δύσκολη και επίπονη διαδικασία. Μια προσέγγιση εδώ είναι ο χρήστης να περιπλανιέται στον Κυβερνοχώρο, ακολουθώντας συνδέσμους που οδηγούν από σελίδα σε σελίδα, και να προσπαθεί να εντοπίσει την πληροφορία που καλύπτει την πληροφοριακή του

ανάγκη. Η παραπάνω διαδικασία περιπλάνησης, είναι συχνά αναποτελεσματική, λόγω του μεγέθους του Παγκοσμίου Ιστού και γιατί τις περισσότερες φορές ο χρήστης δεν γνωρίζει ένα καλό 'σημείο εκκίνησης'. Για τους άπειρους χρήστες, το πρόβλημα της αναζήτησης γίνεται πολύ πιο δύσκολο, συχνά οδηγώντας τους σε απογοητευτικά αποτελέσματα. Το κύριο εμπόδιο εδώ, είναι η απουσία ενός καλά ορισμένου μοντέλου δεδομένων για των Παγκόσμιο Ιστό, το οποίο σημαίνει ότι ο ορισμός και η δόμηση της πληροφορίας είναι ελλιπείς. Αυτές οι δυσκολίες έστρεψαν το ενδιαφέρον στον τομέα της ανάκτησης πληροφορίας και οδήγησαν στην υιοθέτηση των τεχνικών που χρησιμοποιούνται στο πεδίο της ανάκτησης πληροφορίας, ως πολλά υποσχόμενες λύσεις.

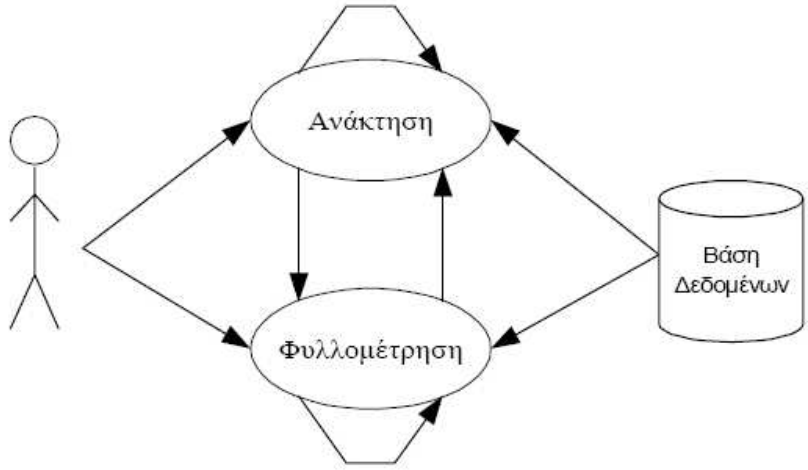
1.2 Βασικές έννοιες

Η αποδοτική ανάκτηση κειμένων αποτελεί συνάρτηση τόσο της διαδικασίας χρήστη όσο και της λογικής αναπαράστασης των κειμένων, όπως αυτή υιοθετείται από το σύστημα. Τις δύο αυτές παραμέτρους θα συζητήσουμε ευθύς αμέσως

1.2.1 Η διαδικασία χρήστη

Σε ένα σύστημα ανάκτησης, ο χρήστης πρέπει να μετατρέψει την πληροφοριακή του ανάγκη σε μορφή ερωτήματος σύμφωνα με την γλώσσα που του παρέχεται από το σύστημα. Σε ένα σύστημα ανάκτησης πληροφορίας, η παραπάνω διαδικασία ανάγεται στην επιλογή από τον χρήστη, ενός καταλλήλου συνόλου λέξεων, αντιπροσωπευτικές για τη σημασιολογία της πληροφοριακής του ανάγκης. Σε ένα σύστημα ανάκτησης δεδομένων, η διατύπωση ενός ερωτήματος, για παράδειγμα με τη χρήση μιας κανονικής έκφρασης συνίσταται στον καθορισμό του συνόλου των περιορισμών που θα πρέπει να ικανοποιεί το σύνολο της απάντησης. Και στις δύο περιπτώσεις, λέμε πως ο χρήστης αναζητά χρήσιμη πληροφορία και κατά συνέπεια εκτελεί μια διαδικασία *ανάκτησης*.

Έχοντας περιγράψει σε γενικές γραμμές την διαδικασία της αναζήτησης ας εξετάσουμε μια δεύτερη διαδικασία ανάκτησης, τη *φυλλομέτρηση* (browsing). Έστω ότι το ενδιαφέρον του χρήστη είτε δεν είναι καλά ορισμένο είτε καλύπτει ένα αρκετά ευρύ φάσμα πληροφοριών. Για παράδειγμα ο χρήστης μπορεί να ενδιαφέρεται για κείμενα σχετικά με αγώνες αυτοκινήτου. Σ' αυτή την περίπτωση θα μπορούσε ο χρήστης απλά να διαβάζει κείμενα από μια συλλογή για αγώνες αυτοκινήτου. Θα μπορούσε, για παράδειγμα, να βρει ενδιαφέροντα κείμενα σχετικά με αγώνες Φόρμουλα Ένα, κατασκευαστές αυτοκινήτων ή ακόμα και για τον αγώνα '24 ωρών του Λε Μαν'. Την ώρα που θα διαβάζει για τις '24 ώρες του Λε Μαν', μπορεί να στρέψει την προσοχή του σε μια παραπομπή για οδηγίες πρόσβασης στο σιρκουί του Λε Μαν και από 'κει για τον τουρισμό στη Γαλλία. Σ' αυτή την περίπτωση λέμε ότι ο χρήστης δεν ψάχνει τη συλλογή αλλά *φυλλομετρά* (browses), τα κείμενά της. Η φυλλομέτρηση είναι κι αυτή μια διαδικασία ανάκτησης πληροφορίας, της οποίας όμως οι σκοποί δεν είναι ξεκάθαρα προσδιορισμένοι τη στιγμή της εκκίνησης και που μπορεί να μεταβληθούν κατά τη διάρκεια της αλληλεπίδρασης με το σύστημα. Η διαδικασία της φυλλομέτρησης, όπως θα δούμε και παρακάτω σε επόμενο κεφάλαιο, είναι μια βασική διαδικασία ανάκτησης αφού μπορούμε να δούμε ποιες είναι οι προτιμήσεις του χρήστη.



Εικόνα 1-1: Αλληλεπίδραση του χρήστη με το σύστημα ΑΠ

Η διαδικασία χρήστη σε ένα σύστημα ανάκτησης μπορεί να λαμβάνει δύο διακριτές μορφές: *ανάκτηση* δεδομένων ή πληροφορίας και *φυλλομέτρηση*. Τα κλασσικά συστήματα ανάκτησης πληροφορίας παρέχουν συνήθως μόνο τη δυνατότητα ανάκτησης. Για παράδειγμα στο σύστημα μιας βιβλιοθήκης, παρέχεται απλά η δυνατότητα ανάκτησης της βιβλιογραφίας που αντιστοιχεί για παράδειγμα σε ένα συγγραφέα. Στη συγκεκριμένη περίπτωση όμως η πληροφοριακή ανάγκη είναι πολύ συγκεκριμένη, ένας συγγραφέας. Τα συστήματα Υπερκειμένου (Hypertext), είναι συνήθως κατασκευασμένα με γνώμονα την εύκολη φυλλομέτρηση. Στις μοντέρνες Ψηφιακές Βιβλιοθήκες όμως καθώς και στις Μηχανές Αναζήτησης στο Παγκόσμιο Ιστό, υπάρχει προσπάθεια να συνδυαστούν οι δύο παραπάνω μορφές για την βελτίωση των δυνατοτήτων ανάκτησης.

Η Εικόνα 1-1 δείχνει την αλληλεπίδραση με το χρήστη μέσα από τις διαφορετικές μορφές διαδικασίας χρήστη που αναφέραμε. Αξίζει να σημειωθεί ότι οι μορφές διαδικασίας χρήστη μπορούν να εναλλάσσονται. Τα περισσότερα σύγχρονα συστήματα ανάκτησης πληροφορίας, παρέχουν τη δυνατότητα ανάκτησης δεδομένων και πληροφορίας. Επίσης τα περισσότερα από αυτά συνήθως παρέχουν και κάποιες στοιχειώδεις μορφές φυλλομέτρησης (συνήθως οδηγώντας μέσω υπερσυνδέσμου σε κάποια σελίδα που επιστράφηκε ως αποτέλεσμα μιας ερώτησης).

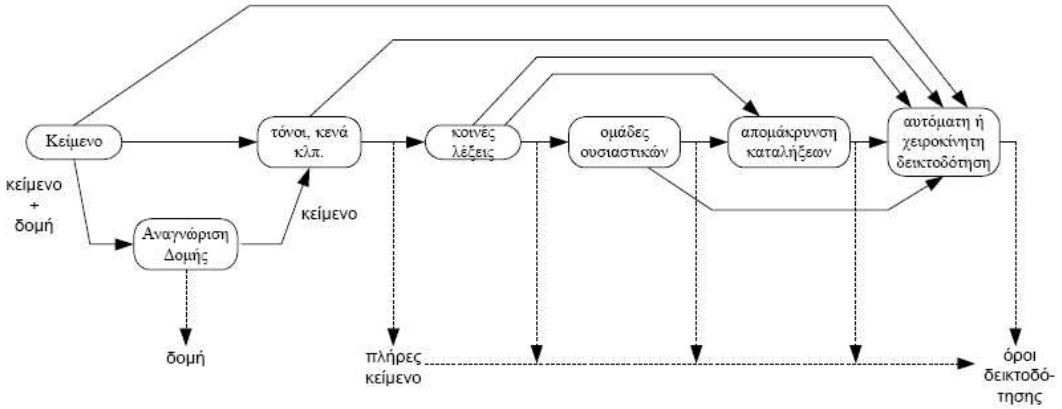
Στην ορολογία που χρησιμοποιείται στον Παγκόσμιο Ιστό, τόσο η διαδικασία της αναζήτησης όσο και η αντίστοιχη της φυλλομέτρησης είναι διαδικασίες '*ανάσυρσης'* (pulling). Αυτό σημαίνει ότι οι χρήστες διατυπώνουν μια απαίτηση πληροφορίας και ανασύρουν από τη συλλογή σχετικά κείμενα. Η αντίστροφη διαδικασία λέγεται '*προώθηση*' (pushing). Αυτή συνίσταται στον καθορισμό μιας σταθερής πληροφοριακής ανάγκης από τον χρήστη και της αποστολής ενός *πράκτορα λογισμικού* (software agent), ο οποίος εξετάζει την συνολική παρεχόμενη πληροφορία και 'προωθεί' τα σχετικά κείμενα προς το χρήστη. Για παράδειγμα κάποιος χρήστης θα ήθελε να παρακολουθεί από μια λίστα συζητήσεων, μόνο τα μηνύματα που τον *ενδιαφέρουν*.

1.2.2 Λογική αναπαράσταση των κειμένων

Για ιστορικούς κυρίως λόγους, τα κείμενα μιας συλλογής αναπαρίστανται συνήθως μέσω ενός συνόλου από όρους δεικτοδότησης (index terms) ή λέξεις-κλειδιά (keywords). Τέτοιες λέξεις-κλειδιά μπορεί να εξάγονται αυτόματα ή να παρέχονται από τον ανθρώπινο παράγοντα (όπως συνηθίζεται σε επιστημονικές δημοσιεύσεις). Ανεξάρτητα με το αν αυτές οι λέξεις κλειδιά παράγονται από κάποιον ειδικό ή εξάγονται αυτόματα, μας παρέχουν μια λογική αναπαράσταση των κειμένων.

Με τους σύγχρονους υπολογιστές μας παρέχεται η δυνατότητα να αναπαραστήσουμε το πλήρες σύνολο όρων που αποτελούν ένα κείμενο. Σ' αυτήν την περίπτωση λέμε ότι έχουμε αναπαράσταση *πλήρους κειμένου.* Σε πολύ μεγάλες συλλογές κειμένων όμως (βλ. Μηχανή Αναζήτησης), οι ανάγκες για αποθήκευση είναι τεράστιες, οπότε ακόμα και με τους σημερινούς υπολογιστές, χρειάζεται να μειώσουμε το μέγεθος της αναπαράστασης. Έτσι καταφεύγουμε σε λύσεις όπως, απομάκρυνση των πιο *κοινών λέξεων* (άρθρα και σύνδεσμοι που καταλαμβάνουν το 40% περίπου των κειμένων), *απομάκρυνση καταλήξεων* (κρατάμε μόνο τη γραμματική ρίζα των λέξεων) και την αναγνώριση ομάδων από ουσιαστικά (απομακρύνοντας ρήματα, επίθετα, επιρρήματα). Τέλος μπορεί να εφαρμοστεί και συμπίεση. Όλες οι παραπάνω ενέργειες ονομάζονται *πράξεις σε κείμενο*. Σκοπός αυτών των πράξεων είναι να μειώσουν την πολυπλοκότητα αναπαράστασης των κειμένων και να μας οδηγήσουν από την αναπαράσταση πλήρους κειμένου, σε αυτή των *όρων δεικτοδότησης*.

Το πλήρες κείμενο είναι σίγουρα η πιο ολοκληρωμένη λογική αναπαράσταση ενός κειμένου αλλά η χρήση της συνήθως δεν είναι υπολογιστικά αποδοτική. Ένα μικρό σύνολο από σημασιολογικές κατηγορίες, που παράγονται από εξειδικευμένο ανθρώπινο παράγοντα, είναι η πιο σύντομη και περιεκτική μορφή αναπαράστασης αλλά η χρήση της μπορεί να οδηγήσει σε χαμηλή ποιότητα ανάκτησης. Μεταξύ των δύο αυτών αναπαραστασιακών άκρων, βρίσκονται διάφορα επίπεδα λογικής αναπαράστασης, που μπορούν να χρησιμοποιηθούν για την λογική αναπαράσταση, όπως φαίνεται στην Εικόνα 1-2. Εκτός από τα ενδιάμεσα αυτά στάδια αναπαράστασης, το σύστημα είναι ίσως δυνατόν να αναγνωρίζει και κάποια δομικά στοιχεία, που συνήθως εμφανίζονται σε ένα κείμενο (π.χ. κεφάλαια, ενότητες, παράγραφοι κλπ.). Αυτή η πληροφορία μπορεί να είναι χρήσιμη και κυρίως όσον αφορά μοντέλα ανάκτησης δομημένου κειμένου, τα οποία όμως δεν αναπτύσσουμε σε αυτές τις σημειώσεις.



Εικόνα 1-2: Λογική Αναπαράσταση κειμένου, από το πλήρες κείμενο στους όρους δεικτοδότησης

1.3 Η διαδικασία της ανάκτησης

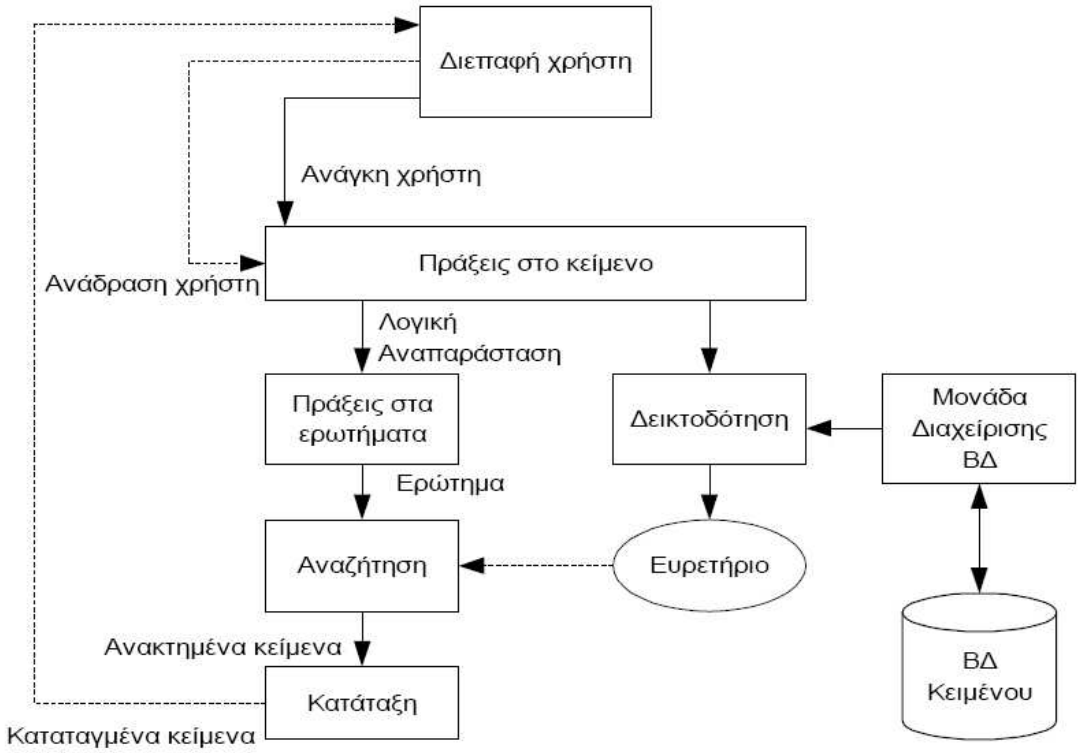
Για να περιγράψουμε τη διαδικασία της ανάκτησης, χρησιμοποιούμε μια απλή και γενικευμένη αρχιτεκτονική λογισμικού, όπως αυτή που φαίνεται στην Εικόνα 1-3. Πρώτ' απ' όλα πριν αρχικοποιηθεί η διαδικασία ανάκτησης, πρέπει να οριστεί η βάση δεδομένων των κειμένων. Αυτό συνήθως γίνεται από τον υπεύθυνο της βάσης δεδομένων, ο οποίος ορίζει τα εξής: α) τα κείμενα που θα χρησιμοποιηθούν, β) τις πράξεις που θα εφαρμοστούν στα κείμενα γ) το μοντέλο των κειμένων (δηλ. τη δομή των κειμένων και ποια είναι τα ανακτόμενα στοιχεία)

Από τη στιγμή που καθορίζεται η λογική αναπαράσταση των κειμένων, ο υπεύθυνος της ΒΔ, κατασκευάζει χρησιμοποιώντας τη Μονάδα Διαχείρισης ΒΔ, το ευρετήριο (index) των κειμένων. Το ευρετήριο είναι μια πολύ κρίσιμη δομή δεδομένων, γιατί επιτρέπει αποδοτική αναζήτηση σε μεγάλο όγκο δεδομένων. Μπορεί να χρησιμοποιηθεί μεγάλη ποικιλία δομών δεικτοδότησης αλλά η πιο δημοφιλής δομή είναι αυτή των *ανεστραμμένων αρχείων*. Τα έξοδα σε χώρο και χρόνο για τον καθορισμό της βάσης δεδομένων και την κατασκευή του ανεστραμμένου αρχείου, κατανέμονται εκτελώντας πολλά ερωτήματα πάνω στη βάση.

Δεδομένου του ότι έχουμε κατασκευάσει ευρετήριο για τη βάση δεδομένων, η διαδικασία της ανάκτησης μπορεί να ξεκινήσει. Ο χρήστης αρχικά καθορίζει μια *ανάγκη χρήστη*, η οποία αναλύεται συντακτικά και στην οποία εφαρμόζονται όλες οι πράξεις που εφαρμόζονται και στα κείμενα της βάσης. Στη συνέχεια πρέπει να εφαρμοστούν οι λεγόμενες *πράξεις στο ερώτημα* (query operations), για να προκύψει το πραγματικό ερώτημα, το οποίο αποτελεί αναπαράσταση, σε επίπεδο συστήματος, της ανάγκης χρήστη. Κατόπιν το ερώτημα επεξεργάζεται για να προκύψουν τα *ανακτημένα κείμενα.* Η επεξεργασία του ερωτήματος γίνεται γρήγορα, χάρη στο ευρετήριο, που χτίστηκε στο προηγούμενο βήμα.

Πριν παρουσιαστούν τα αποτελέσματα στο χρήστη, τα ανακτημένα κείμενα κατατάσσονται με βάση μια εκτίμηση για σχετικότητά τους. Στη συνέχεια, ο χρήστης εξετάζει το σύνολο των καταταγμένων κειμένων για να εντοπίσει χρήσιμη πληροφορία. Σ' αυτό το σημείο μπορεί να καταδείξει μια σειρά από κείμενα που είναι βέβαιο ότι ικανοποιούν την πληροφοριακή του ανάγκη και να ξεκινήσει έτσι έναν κύκλο *ανάδρασης χρήστη* (user feedback). Κατά τη διάρκεια ενός τέτοιου κύκλου, το σύστημα χρησιμοποιεί τα κείμενα που επιλέχθηκαν από τον χρήστη για να επαναδιατυπώσει το ερώτημα, με την ελπίδα ότι το επαναδιατυπωμένο ερώτημα είναι καλύτερη αναπαράσταση της πραγματικής ανάγκης χρήστη.

Δεδομένων των διαθέσιμων διεπαφών χρήστη που είναι διαθέσιμες στα σύγχρονα συστήματα ΑΠ (Μηχανές Αναζήτησης και Web browsers), εύκολα διαπιστώνει κανείς ότι ο χρήστης δεν διατυπώνει σχεδόν ποτέ την πραγματική του πληροφοριακή ανάγκη. Αυτό που στην πράξη συμβαίνει, είναι ο χρήστης να καλείται να παρέχει μια διατύπωση του ερωτήματος που θα επεξεργαστεί το σύστημα. Καθώς οι περισσότεροι χρήστες δεν έχουν γνώση των πράξεων που εφαρμόζονται στο κείμενο και στα ερωτήματα, το ερώτημα που παρέχουν είναι συχνά ανεπαρκώς διατυπωμένο. Γι' αυτό δεν ξενίζει το γεγονός ότι ελλιπώς διατυπωμένα ερωτήματα, οδηγούν σε κακή ανάκτηση πληροφορίας (όπως συμβαίνει συχνά στο Διαδίκτυο).



Εικόνα 1-3: Η Διαδικασία της Ανάκτησης Πληροφορίας

Τέλος επάνω σε αυτή την διαδικασία θα δούμε κάποια βασικά μοντέλα ανάκτησης και θα δείξουμε πως μπορούν αυτά να βελτιστοποιηθούν, έτσι ώστε να έχουμε καλύτερα αποτελέσματα στην αναζήτησή μας στο διαδίκτυο.

Κεφάλαιο 2^{ο}

ΜΟΝΤΕΛΑ ΑΝΑΚΛΗΣΗΣ

2.1 Εισαγωγή

Όπως γνωρίζουμε, η πιο συνηθισμένη πρακτική για την δεικτοδότηση και την ανάκτηση κειμένων είναι η χρήση των *όρων δεικτοδότησης* (index terms). Ένας όρος δεικτοδότησης είναι μια λέξη κλειδί ή μια ομάδα εννοιολογικά συσχετιζόμενων λέξεων, η εμφάνιση των οποίων λαμβάνει από μόνη της μια αυτόνομη έννοια (π.χ. computer algorithm). Κατά μια πιο απλοποιημένη εκδοχή, ένας όρος δεικτοδότησης είναι απλά μια λέξη που εμφανίζεται σε ένα κείμενο της συλλογής. Η ανάκτηση που βασίζεται στο ταίριασμα όρων δεικτοδότησης ερωτήματος και κειμένων της συλλογής, είναι πολύ απλή αλλά εισάγει ένα σύνολο προβληματισμών για την αποτελεσματικότητά της. Για παράδειγμα, η βασική υπόθεση που εισάγει η παραπάνω στρατηγική, είναι ότι η σημασιολογία τόσο των κειμένων όσο και της πληροφοριακής ανάγκης του χρήστη, μπορεί να εκφραστεί με φυσικό τρόπο, μέσα από ένα σύνολο λέξεων. Στην πράξη ένα σημαντικό κομμάτι από τη σημασιολογία του κειμένου χάνεται κατά τη μεταφορά στο χώρο του ευρετηρίου.

Ο λόγος γι' αυτήν την απώλεια είναι ότι οι λέξεις αποκτούν την ερμηνεία τους ανάλογα με το πλαίσιο συμφραζομένων στο οποίο εμφανίζονται. Από αυτή την παρατήρηση πηγάζουν δυο φαινόμενα, η *πολυσημία* και η *συνωνυμία*. Στην πολυσημία, έχουμε το φαινόμενο ο ίδιος όρος να λαμβάνει διαφορετικές έννοιες ανάλογα με τα συμφραζόμενα που συνοδεύουν την εμφάνισή του. Για παράδειγμα ο όρος 'spider' μπορεί να χρησιμοποιείται για να δηλώσει ένα 'web spider' αν το κείμενο μιλάει για το Διαδίκτυο ή το έντομο σε άλλες περιπτώσεις. Στην συνωνυμία, διαφορετικοί όροι μπορούν να περιγράφουν την ίδια έννοια γιατί εμφανίζονται στα ίδια πλαίσια συμφραζομένων. Για παράδειγμα η έννοια 'αυτοκίνητο', μπορεί να περιγράφεται ισοδύναμα από τις λέξεις: 'αυτοκίνητο', 'αμάξι', 'τετράτροχο', 'όχημα'. Η συνωνυμία και η πολυσημία, αποτελούν κλασσικά προβλήματα που συνδέονται με τον τρόπο λογικής αναπαράστασης των κειμένων μέσω ευρετηρίου.

Έχοντας υπόψη μας τα παραπάνω προβλήματα και με δεδομένο ότι η διαδικασία της αντιστοίχισης του ερωτήματος στη συλλογή των κειμένων, γίνεται στο χώρο του ευρετηρίου, μπορούμε να κατανοήσουμε γιατί συχνά τα αποτελέσματα μιας ερώτησης διατυπωμένης με λέξεις-κλειδιά δεν είναι τα αναμενόμενα. Αν μάλιστα λάβουμε υπόψη μας και το γεγονός ότι πολλοί χρήστες δεν είναι σε θέση να επιλέξουν τις κατάλληλες λέξεις-κλειδιά για τον σχηματισμό ερωτήσεων, το πρόβλημα μεγαλώνει. Ένα καλό παράδειγμα του παραπάνω προβλήματος είναι τα απογοητευτικά αποτελέσματα σε πολλά από τα ερωτήματα που υποβάλλονται σε μια Μηχανή Αναζήτησης στο Παγκόσμιο Ιστό (όπου και μεγάλο μέρος των χρηστών είναι χωρίς μεγάλη εμπειρία στο σχηματισμό ερωτήσεων). Η πρόκληση για ένα μοντέλο για ΑΠ, είναι να δημιουργήσει το υπόβαθρο, ώστε να υπάρξει ταίριασμα της πληροφοριακής ανάγκης χρήστη με τα κείμενα της συλλογής, παρά την ανακριβή αναπαράσταση και με όσο το δυνατόν μικρότερες αποκλίσεις.

Στο πνεύμα της ανάκτησης πληροφορίας, ταίριασμα σημαίνει εκτίμηση από το σύστημα, της σχετικότητας των κειμένων ως προς το δοθέν ερώτημα. Μια τέτοια εκτίμηση επιτυγχάνεται με την χρήση ενός αλγορίθμου κατάταξης (ranking), με βάση τον οποίο, γίνεται μια απλή διάταξη των κειμένων. Τα κείμενα που εμφανίζονται στις πρώτες θέσεις αυτής της διάταξης, θεωρούνται ως το πιο πιθανό να είναι σχετικά με την ερώτηση, με την πιθανότητα να φθίνει, όσο εξετάζουμε τη διάταξη προς τις χαμηλότερες θέσεις. Οι αλγόριθμοι κατάταξης έχουν ζωτική σημασία σε ένα σύστημα

ΑΠ. Συνεπώς μία βασική λειτουργία του μοντέλου είναι να παρέχει έναν αλγόριθμο κατάταξης για κάθε ερώτημα που υποβάλλεται.

Ο τρόπος θεώρησης της λογικής αναπαράστασης των κειμένων και η συσχέτισή του με τον αλγόριθμο κατάταξης, είναι το βασικό χαρακτηριστικό που διαφοροποιεί τα μοντέλα ΑΠ. Στο κεφάλαιο αυτό εξετάζουμε μια κατηγοριοποίηση των μοντέλων, κάποιους τυπικούς ορισμούς, και τέλος παρουσιάζουμε τα κυριότερα μοντέλα ΑΠ και πως αυτά μας βοήθησαν για την υλοποίηση του επόμενου κεφαλαίου.

2.2 Ταξινόμηση των Μοντέλων για Ανάκτηση Πληροφορίας

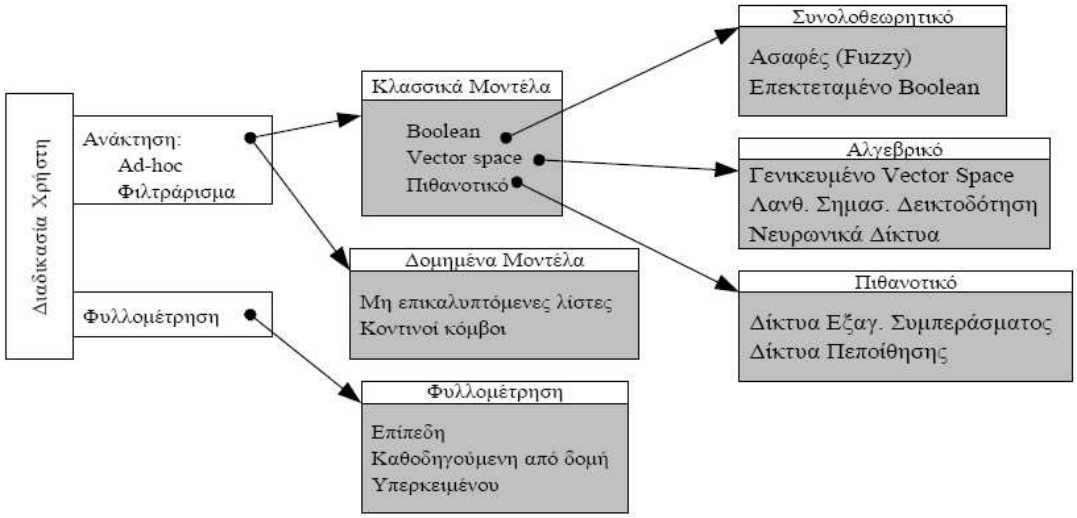
Τα τρία κλασσικά μοντέλα στην Ανάκτηση Πληροφορίας είναι το Boolean Μοντέλο (Μοντέλο Δυαδικής Λογικής), το Vector Space Μοντέλο (Μοντέλο Διανυσματικού Χώρου) και το Πιθανοτικό Μοντέλο. Στο μοντέλο Boolean, τόσο τα κείμενα όσο και τα ερωτήματα αντιμετωπίζονται ως ένα σύνολο από όρους δεικτοδότησης. Κατά συνέπεια το μοντέλο μπορεί να θεωρηθεί ως συνολοθεωρητικό. Στο Vector Space μοντέλο, τα κείμενα και τα ερωτήματα αναπαρίστανται ως διανύσματα σε έναν t-διάστατο[2] χώρο. Έτσι λέμε ότι το μοντέλο είναι αλγεβρικό. Το Πιθανοτικό μοντέλο εισάγει έναν τρόπο αναπαράστασης, ο οποίος βασίζεται στην πιθανοθεωρία. Κατά συνέπεια το μοντέλο είναι πιθανοτικού χαρακτήρα.

Με τον καιρό προτάθηκαν διάφορες νέες προσεγγίσεις σε καθεμία από τις κατηγορίες βασικών μοντέλων. Έτσι έχουμε στο συνολοθεωρητικό πεδίο τα μοντέλα: ασαφές (fuzzy) Boolean και επεκτεταμένο Boolean. Στα αλγεβρικά μοντέλα έχουμε το γενικευμένο Vector Space, την λανθάνουσα σημασιολογική δεικτοδότηση (latent semantic indexing, LSI) και το μοντέλο των νευρωνικών δικτύων. Στον πιθανοτικό τομέα εμφανίστηκαν τα δίκτυα εξαγωγής συμπεράσματος (inference networks) και τα δίκτυα πεποίθησης (belief networks). Η Εικόνα 3-1 δίνει σχηματικά την κατηγοριοποίηση αυτή.

Εκτός από την χρήση του περιεχομένου των κειμένων, ορισμένα μοντέλα εκμεταλλεύονται και την εσωτερική δομή που φυσιολογικά υπάρχει στο γραπτό λόγο. Σε αυτή την περίπτωση λέμε ότι έχουμε ένα δομημένο μοντέλο. Για τη δομημένη ανάκτηση κειμένου, συναντούμε δύο μοντέλα, τις μη επικαλυπτόμενες λίστες (non-overlapping lists) και τους κοντινούς κόμβους (proximal nodes).

Η διαδικασία του χρήστη μπορεί εκτός από αναζήτηση να έχει μορφή φυλλομέτρησης. Σε αυτή την κατηγορία εντοπίζουμε τρία μοντέλα για φυλλομέτρηση: επίπεδη (flat), καθοδηγούμενη από τη δομή (structure guided), φυλλομέτρηση υπερκειμένου (hypertext browsing).

Στο κεφάλαιο αυτό αναπτύσσουμε μόνο τα συνολοθεωρητικά και αλγεβρικά μοντέλα (εκτός των νευρωνικών δικτύων) και το βασικό πιθανοτικό μοντέλο και τα συγκρίνουμε με την υλοποίηση που ακολουθεί στο επόμενο κεφάλαιο, η οποία βασίζεται στο πιθανοτικό μοντέλο. Τα υπόλοιπα μοντέλα τα αναφέρουμε απλώς για πληρότητα και ο ενδιαφερόμενος θα πρέπει να ανατρέξει σε επιπλέον τεχνικά κείμενα για μια πλήρη επισκόπηση όλων των μοντέλων.



Εικόνα 3-1: Ταξινόμηση των μοντέλων Ανάκτησης Πληροφορίας

2.3 Τυπικός ορισμός των μοντέλων ΑΠ

Πριν προχωρήσουμε στην εξέταση των επί μέρους μοντέλων θα δώσουμε έναν τυπικό και ακριβή ορισμό για το τι είναι ένα μοντέλο ΑΠ.
Ορισμός *Ένα μοντέλο ανάκτησης πληροφορίας είναι μία τετράδα* [**D, Q,** *F, R*(q_i, d_j)] *όπου:*

1) **D** *είναι ένα σύνολο από λογικές αναπαραστάσεις για τα κείμενα της συλλογής*
2) **Q** *είναι ένα σύνολο από λογικές αναπαραστάσεις για τις πληροφοριακές ανάγκες του χρήστη. Αυτές οι αναπαραστάσεις καλούνται ερωτήματα*
3) **F** *είναι ένα υπόβαθρο για την μοντελοποίηση της αναπαράστασης των κειμένων, των ερωτημάτων και των σχέσεων μεταξύ τους*
4) **R**(q_i, d_j) είναι μια συνάρτηση κατάταξης, η οποία συνδέει έναν πραγματικό αριθμό με ένα ερώτημα $q_i \in$ **Q** και μια αναπαράσταση κειμένου $d_j \in$ **D**. Μια τέτοια κατάταξη ορίζει μια διάταξη πάνω στα κείμενα πάντα με βάση το ερώτημα. q_i.

Διαισθητικά ο παραπάνω ορισμός περιγράφει τη διαδικασία καθορισμού ενός μοντέλου ΑΠ. Η διαδικασία ορισμού ενός μοντέλου είναι η ακόλουθη. Αρχικά επινοείται ένας τρόπος αναπαράστασης για τα κείμενα και την πληροφοριακή ανάγκη του χρήστη. Έπειτα καθορίζεται ένα υπόβαθρο στο οποίο θα μπορούν αυτές οι αναπαραστάσεις να μοντελοποιηθούν. Το υπόβαθρο αυτό, θα πρέπει να παρέχει και τον μηχανισμό κατάταξης. Για παράδειγμα στο Boolean μοντέλο, το υπόβαθρο αυτό αποτελείται από τις αναπαραστάσεις των κειμένων και των ερωτήσεων ως σύνολα, και τις κλασσικές πράξεις πάνω στα σύνολα. Αντίστοιχα στο μοντέλο διανυσματικού χώρου, το υπόβαθρο αποτελείται από τις διανυσματικές αναπαραστάσεις κειμένων στον *t*-διάστατο διανυσματικό χώρο και τις επιτρεπτές αλγεβρικές πράξεις πάνω σε διανύσματα.

2.4 Κλασσικά μοντέλα ΑΠ

Σ' αυτή την ενότητα παρουσιάζουμε εν συντομία τα μοντέλα Boolean, το μοντέλο Vector Space καθώς και το πιθανοτικό.

2.4.1 Βασικές υποθέσεις

Τα κλασσικά μοντέλα στην ανάκτηση πληροφορίας θεωρούν ότι κάθε κείμενο περιγράφεται από ένα σύνολο από αντιπροσωπευτικές λέξεις κλειδιά, που ονομάζονται όροι δεικτοδότησης. Ένας *όρος δεικτοδότησης* (index term), είναι μια λέξη το σημασιολογικό περιεχόμενο της οποίας, περικλείει ένα μέρος του θέματος με το οποίο ασχολείται το κείμενο. Έτσι τα κείμενα μπορούν να αναπαρασταθούν ως σύνολα όρων, που συνοψίζουν το περιεχόμενό τους. Γενικά οι όροι δεικτοδότησης είναι συνήθως ουσιαστικά γιατί τα ουσιαστικά αναπαριστούν μια έννοια χωρίς την ανάγκη να εμφανίζονται δίπλα σε άλλο μέρος του λόγου και η σημασιολογία τους είναι εύκολα αντιληπτή. Σύνδεσμοι και επιρρήματα, θεωρούνται ότι έχουν κυρίως συμπληρωματικό χαρακτήρα. Συχνά όμως χρειάζεται να χρησιμοποιούμε και αυτά τα μέρη του λόγου στο ευρετήριο, όπως για παράδειγμα στις Μηχανές Αναζήτησης.

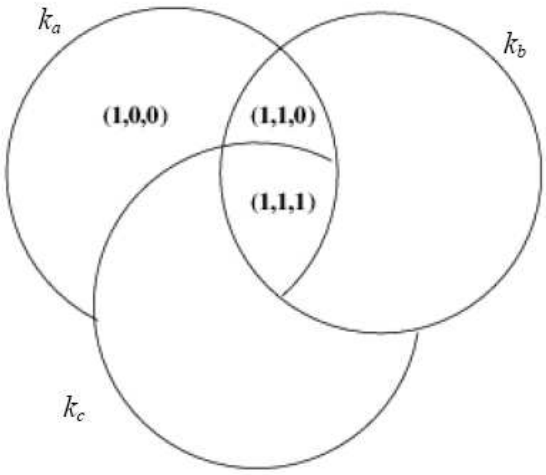
Με τη δεδομένη αναπαράσταση των κειμένων ως συλλογές όρων, μπορεί κάποιος να παρατηρήσει ότι δεν έχουν όλοι οι όροι την ίδια ισχύ ως προς την περιγραφή ενός κειμένου. Με άλλα λόγια η ερμηνεία ενός όρου συχνά μπορεί να δίνει μια γενικευμένη ή και ασαφή περιγραφή ενός κειμένου. Τέτοιοι όροι είναι αυτοί που εμφανίζονται με μεγάλη συχνότητα στην πλειονότητα των κειμένων μιας συλλογής. Έστω για παράδειγμα μία συλλογή κειμένων γύρω από υπολογιστές. Η λέξη 'computer' σε μια τέτοια συλλογή εμφανίζεται με μεγάλη βεβαιότητα σχεδόν σε κάθε κείμενο και αν και περιγράφει κάτι αρκετά συγκεκριμένο, δεν είναι αντιπροσωπευτική του κειμένου στο οποίο εμφανίζεται. Αντίθετα αν μια λέξη εμφανίζεται σε μικρό εύρος κειμένων, τότε είναι σχεδόν σίγουρο ότι έχει μεγαλύτερη *βαρύτητα* στην περιγραφή ενός κειμένου. Στο προηγούμενο παράδειγμα η λέξη 'inheritance', εμφανίζεται σίγουρα σε πολύ λιγότερα κείμενα απ' ότι η λέξη 'computer'. Η εμφάνισή της ως όρος δεικτοδότησης για κάποιο κείμενο, μας καθοδηγεί αμέσως ότι το συγκεκριμένο κείμενο συζητά για κληρονομικότητα σε αντικειμενοστραφή προγραμματισμό. Για να προσομοιώσουμε το γεγονός ότι διαφορετικοί όροι μπορεί να έχουν διαφορετική βαρύτητα ως προς στην δεικτοδότηση των κειμένων, σε κάθε όρο δεικτοδότησης αναθέτουμε και ένα αριθμητικό *βάρος*, θα δούμε όμως στο επόμενο κεφάλαιο ότι αυτό το βάρος δεν είναι αρκετό για την σωστή ανάκτηση των κείμενων διότι μεγάλο ρόλο παίζει και η σωστή ομαδοποίηση τους.

Συγκεκριμένα έστω k_i ένας όρος δεικτοδότησης, και d_j ένα κείμενο. Ο αριθμός $w_{i,j} \geq 0$ είναι το *βάρος*, που αντιστοιχεί στο ζεύγος (k_i, d_j) και αντιστοιχεί στο πόσο αντιπροσωπευτικός είναι ο k_i για το κείμενο d_j.

Ορισμός *Έστω t ο αριθμός των όρων δεικτοδότησης στο σύστημα και k_i ένας γενικός όρος δεικτοδότησης. Το σύνολο $K = \{k_1, k_2, \ldots, k_t\}$ περιέχει όλους τους όρους δεικτοδότησης. Ένα βάρος $w_{i,j} > 0$ συνδέεται με κάθε όρο k_i, που εμφανίζεται στο κείμενο d_j. Για κάποιον όρο δεικτοδότησης που δεν εμφανίζεται στο κείμενο, $w_{i,j} = 0$. Κάθε κείμενο d_j έχει ένα αντιπροσωπευτικό διάνυσμα $\vec{d}_j$, το οποίο αναπαρίσταται ως $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$. Επιπλέον έστω g_i, μια συνάρτηση που επιστρέφει το βάρος που συνδέεται με τον όρο, σε κάθε t-διάστατο διάνυσμα* (δηλ. $g_i(\vec{d}_j) = w_{i,j}$).

Τα παραπάνω βάρη υποθέτουμε ότι είναι μεταξύ τους ανεξάρτητα. Αυτό σημαίνει ότι, για παράδειγμα, η τιμή του $w_{i,j}$ δεν επηρεάζει την τιμή του $w_{i+1,j}$. Αυτή η υπόθεση είναι

υπερπλουστευτική δεδομένου ότι συχνά έχουμε συμπλέγματα όρων που εμφανίζονται μαζί.



Εικόνα 3-2: Οι συζευκτικές συνιστώσες του ερωτήματος $[q = k_a \wedge (k_b \vee \neg k_c)]$

Ένα τέτοιο παράδειγμα είναι οι όροι *δίκτυο* και *υπολογιστής.* Σε μια συλλογή με θέμα τα δίκτυα υπολογιστών, αναμένεται αυτοί οι δύο όροι να έχουν παρόμοιες συχνότητες εμφάνισης. Κατά συνέπεια οι δύο αυτοί όροι είναι συσχετισμένοι μεταξύ τους και ο υπολογισμός της ανάθεσης βαρών θα πρέπει να λαμβάνει υπόψη του αυτή τη συσχέτιση. Λαμβάνοντας υπόψη μας τις συσχετίσεις των όρων μεταξύ τους, η πολυπλοκότητα υπολογισμού των βαρών αυξάνει, συμπαρασύροντας και τον υπολογισμό της κατάταξης. Γι' αυτό στο εξής θα θεωρούμε ότι οι διακριτοί όροι δεικτοδότησης είναι μεταξύ τους ανεξάρτητοι.

2.4.2 Το Boolean μοντέλο

Το Boolean μοντέλο, είναι ένα απλό μοντέλο ανάκτησης πληροφορίας που βασίζεται στη θεωρία συνόλων και στην άλγεβρα Boole. Το υπόβαθρό του είναι εύληπτο και ταυτόχρονα κομψό και καλά ορισμένο στη βάση της άλγεβρας συνόλων. Τα ερωτήματα μπορούν να αναπαρασταθούν με σαφή τρόπο, με χρήση άλγεβρας Boole.

Συγκεκριμένα στο Boolean μοντέλο, κάθε όρος δεικτοδότησης θεωρείται ότι ανήκει εξ' ολοκλήρου ή δεν ανήκει σε ένα κείμενο. Κατά συνέπεια τα βάρη θεωρούνται δυαδικά, δηλ. $w_{i,j} \in \{0,1\}$. Το κάθε ερώτημα θεωρείται ότι αποτελείται από όρους δεικτοδότησης οι οποίοι συνδέονται με έναν από τους τελεστές *and, or, not.* Δηλαδή κάθε ερώτημα είναι μια Boolean έκφραση που μπορεί να γραφεί σε διαζευκτική κανονική μορφή (Disjunctive Normal Form, DNF). Για παράδειγμα το ερώτημα $[q = k_a \wedge (k_b \wedge \neg k_c)]$ μπορεί να γραφεί σε DNF ως $[q_{dnf} = (k_a \wedge k_b) \vee (k_a \wedge \neg k_c)]$. Έστω τώρα ένα διάνυσμα με δυαδικά βάρη που αντιστοιχεί σε ανάθεση αλήθειας (truth assignment) σε συζευκτικές εκφράσεις της τριάδας (k_a, k_b, k_c). Για παράδειγμα στην έκφραση $k_a \wedge k_b$ μια ανάθεση αλήθειας είναι η (1,1,0). Άρα το αρχικό ερώτημα μπορεί να αναλυθεί σε διάζευξη τέτοιων διανυσμάτων ως εξής, $[q_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)]$. Ο λόγος για την εισαγωγή των δυαδικών αυτών διανυσμάτων είναι γιατί υπάρχει απευθείας αντιστοιχία του ερωτήματος q_{dnf} στο διάγραμμα που φαίνεται στην Εικόνα 3-2

Ορισμός *Στο Boolean μοντέλο τα βάρη που ανατίθενται στους όρους δεικτοδότησης είναι δυαδικά δηλαδή,* $w_{ij} \in \{0,1\}$. *Ένα ερώτημα q είναι μια συνήθης Boolean έκφραση. Έστω* $\vec{q}_{dnf}$ *η διαζευκτική κανονική μορφή του ερωτήματος και* $\vec{q}_{cc}$ *καθεμία από τις συζευκτικές συνιστώσες (conjunctive components) του* $\vec{q}_{dnf}$ *(τα δυαδικά διανύσματα που προαναφέραμε). Η ομοιότητα του κειμένου* d_j *προς το ερώτημα q ορίζεται ως εξής:*

$$sim(d_j, q) = \begin{cases} 1, \text{ αν } \exists \vec{q}_{cc} \text{ τέτοιο ώστε } (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0, \text{ διαφορετικά} \end{cases}$$

Αν $sim(d_j, q) = 1$, *τότε το Boolean μοντέλο προβλέπει ότι το κείμενο* d_j *είναι σχετικό με το ερώτημα q (μπορεί και να μην είναι). Διαφορετικά η πρόβλεψη είναι ότι το κείμενο είναι άσχετο.*

Το Boolean μοντέλο προβλέπει ότι κάθε κείμενο είτε είναι σχετικό είτε όχι, και δεν υπάρχει η έννοια της *μερικής ικανοποίησης* των συνθηκών του ερωτήματος. Για παράδειγμα έστω d_j τέτοιο ώστε να είναι $\vec{d}_j = (0,1,0)$. Το κείμενο d_j περιέχει τον όρο k_b αλλά θεωρείται άσχετο ως προς το ερώτημα $[q = k_a \wedge (k_b \vee \neg k_c)]$. Λόγω αυτής της έλλειψης το Boolean μοντέλο, στην ουσία εκτελεί περισσότερο *ανάκτηση δεδομένων* (data retrieval) παρά πληροφορίας.

Τα κύρια πλεονεκτήματα του Boolean μοντέλου είναι ο φορμαλισμός του και η απλότητά του. Το κύριο μειονέκτημά του είναι ότι δεν υπάρχει διαβάθμιση σχετικότητας ως προς το ερώτημα κάτι που μπορεί να οδηγήσει σε χαμηλής ποιότητας ανάκτηση πληροφορίας. Ένα δεύτερο μειονέκτημά του είναι ότι συχνά δεν είναι εύκολη η έκφραση της πληροφοριακής ανάγκης του χρήστη με τον φορμαλισμό που επιβάλλει το Boolean μοντέλο (με Boolean άλγεβρα). Η πληροφοριακή ανάγκη μπορεί να έχει τόσο συγκεκριμένη μορφή, όταν για παράδειγμα ψάχνουμε σε μια βιβλιοθήκη για ένα περιοδικό. Τότε αρκεί εισάγουμε τον τίτλο του και να ανακτήσουμε τις ανάλογες εγγραφές. Λόγω αυτών των χαρακτηριστικών του, το Boolean μοντέλο έχει βρει εφαρμογή σε κυρίως εμπορικά συστήματα βιβλιοθηκών. Η υλοποίηση του επόμενου κεφαλαίο δεν πηγάζει από αυτό το μοντέλου λόγο τον παραπάνω μειονεκτημάτων του.

2.4.3 Το μοντέλο Vector Space

Το μοντέλο Vector Space, αντιμετωπίζει την ανεπάρκεια της ανάθεσης δυαδικών βαρών και εισάγει ένα υπόβαθρο στο οποίο επιτρέπεται προσεγγιστικό ταίριασμα. Τα βάρη που ανατίθενται στους όρους δεικτοδότησης, τόσο για τα κείμενα όσο και για τα ερωτήματα είναι μη δυαδικά και χρησιμοποιούνται για τον υπολογισμό του βαθμού ομοιότητας μεταξύ του ερωτήματος και κάθε αποθηκευμένου κειμένου. Κατόπιν τα κείμενα διατάσσονται με φθίνουσα σειρά, με κριτήριο τον βαθμό ομοιότητάς τους με το ερώτημα του χρήστη. Έτσι στο μοντέλο Vector Space λαμβάνονται υπόψη και κείμενα που ικανοποιούν μερικώς τις συνθήκες του ερωτήματος και το τελικό αποτέλεσμα είναι πολύ πιο ακριβές σε σχέση με την Boolean ανάκτηση.

Ορισμός *Στο μοντέλο Vector Space το βάρος $w_{i,j}$ που αντιστοιχεί στο ζεύγος (k_i, d_j) είναι θετικό και όχι δυαδικό. Επιπλέον ανατίθενται βάρη και στους όρους δεικτοδότησης του ερωτήματος. Έστω $w_{i,q}$ το βάρος που αντιστοιχεί στο ζεύγος $[k_i, q]$, όπου $w_{i,q} \geq 0$. Τότε το διάνυσμα του ερωτήματος ορίζεται ως $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$ όπου* t *είναι ο συνολικός αριθμός των όρων δεικτοδότησης στο σύστημα. Όπως και πριν το διάνυσμα του $\vec{d}_j$, είναι $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$.*

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|}$$

$$= \frac{\sum_{i=1}^{t} (w_{i,j} \times w_{i,q})}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

όπου $\left|\vec{d}_j\right|$ και $\left|\vec{q}\right|$ είναι οι νόρμες των διανυσμάτων.

Εφόσον $w_{i,j} \geq 0$ και $w_{i,q} \geq 0$, το $sim(d_j, q)$ παίρνει τιμές από 0 (καμία ομοιότητα) ως +1 (τέλειο ταίριασμα). Έτσι το μοντέλο Vector Space, αντί να προσπαθήσει να προσδιορίσει αν ένα κείμενο είναι ή όχι σχετικό, διατάσσει τα κείμενα με κριτήριο τον βαθμό ομοιότητάς τους προς το ερώτημα. Με αυτή τη στρατηγική ένα κείμενο μπορεί να ανακτηθεί ακόμα και αν ταιριάζει κατά προσέγγιση με το ερώτημα. Επειδή δεν θέλουμε να ανακτήσουμε όλα τα κείμενα που έχουν μη μηδενικό βαθμό σχετικότητας με το ερώτημα, αλλά αυτά που ταιριάζουν περισσότερο, θέτουμε ένα κατώφλι ελέγχου για το $sim(d_j, q)$. Κείμενα με βαθμό ομοιότητας μεγαλύτερο απ' αυτό το κατώφλι επιστρέφονται στο χρήστη. Πριν όμως υπολογίσουμε την κατάταξη των κειμένων πρέπει να εξετάσουμε τον τρόπο υπολογισμού των βαρών.

Το πρόβλημα υπολογισμού των βαρών ανάγεται στο εξής πρόβλημα ομαδοποίησης (clustering). Έστω μια συλλογή κειμένων C και ένα σύνολο A από κείμενα της συλλογής. Στο πρόβλημα της ΑΠ, το A είναι το σύνολο εκείνο των κειμένων που απαντούν σε μια πληροφοριακή ανάγκη. Η διατύπωση της πληροφοριακής ανάγκης που καθορίζει το A, μπορεί να είναι σχετικά ασαφής, οπότε τα θέματα που πρέπει να αντιμετωπιστούν είναι δυο ειδών. Πρώτον, πρέπει να καθοριστεί ποια χαρακτηριστικά χαρακτηρίζουν τα κείμενα του A. Και δεύτερον πρέπει να καθοριστεί ποια χαρακτηριστικά διαχωρίζουν τα κείμενα του συνόλου A από τα κείμενα του C. Η εξισορρόπηση της επίδρασης αυτών των δύο ομάδων χαρακτηριστικών είναι το αντικείμενο ενός καλού σχήματος ανάθεσης βαρών.

Ένα καλό μέτρο για τον χαρακτηρισμό των στοιχείων εντός του συνόλου A είναι η συχνότητα εμφάνισης του όρου k_i σε κάθε κείμενο d_j. Διαισθητικά όσο πιο συχνά εμφανίζεται ένας όρος k_i σε ένα κείμενο d_j, τόσο πιο καλή περιγραφή του d_j αποτελεί ο k_i. Η συχνότητα εμφάνισης του όρου, ονομάζεται συχνά και παράγοντας tf (tf = term frequency). Επίσης ένα μέτρο για τον διαχωρισμό των συνόλων A και C, αποτελεί η αντίστροφη συχνότητα εμφάνισης του k_i στα κείμενα της συλλογής. Διαισθητικά αν ο k_i έχει μεγάλη συχνότητα εμφάνισης στη συλλογή, δεν είναι πολύ χρήσιμος για να χαρακτηρίσει ένα κείμενο και άρα να διαχωρίσει μια ομάδα κειμένων μες στη συλλογή. Η αντίστροφη συχνότητα εμφάνισης αναφέρεται συνήθως ως παράγοντας idf (idf = inverse document frequency). Το καλύτερο σχήμα υπολογισμού βαρών, πρέπει να προκύψει από τον κατάλληλο συνδυασμό αυτών των δύο παραγόντων.

Ορισμός *Έστω N ο συνολικός αριθμός των κειμένων και n_i ο αριθμός των κειμένων στα οποία εμφανίζεται ο όρος k_i. Έστω $freq_{i,j}$ η συχνότητα εμφάνισης του k_i στο d_j. Τότε η κανονικοποιημένη συχνότητα $f_{i,j}$ του όρου k_i στο d_j δίνεται από τη σχέση*

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}} \qquad (2.1)$$

όπου το max υπολογίζεται πάνω σε κάθε όρο που αναφέρεται στο κείμενο d_j. Αν ο k_i δεν εμφανίζεται στο d_j τότε $f_{i,j} = 0$. Επιπλέον, έστω idf_i η αντίστροφη συχνότητα εμφάνισης για τον k_i, που δίνεται από τον τύπο

$$idf_i = \log \frac{N}{n_i} \qquad (2.2)$$

Τότε το καλύτερο γνωστό σχήμα υπολογισμού βαρών δίνεται από το γινόμενο

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i} \qquad (2.3)$$

ή από παραλλαγή του παραπάνω. Τέτοια σχήματα υπολογισμού βαρών λέγονται σχήματα tf-idf.

Για τα βάρη των όρων στο ερώτημα οι Salton και Buckley, πρότειναν

$$w_{i,q} = \left(0.5 + \frac{0.5\, freq_{i,q}}{\max_l freq_{l,q}} \right) \times \log \frac{N}{n_i} \qquad (2.4)$$

όπου $freq_{i,q}$ είναι η συχνότητα εμφάνισης του όρου k_i στο κείμενο που αντιπροσωπεύει την πληροφοριακή ανάγκη q. Ο αθροιστικός παράγοντας 0.5, έχει προκύψει πειραματικά πως εξισορροπεί το γεγονός ότι το ερώτημα απαρτίζεται συνήθως από πολύ λίγους όρους.

Τα κύρια πλεονεκτήματα του μοντέλου Vector Space, είναι τα εξής: 1) το σχήμα υπολογισμού βαρών που χρησιμοποιεί, βελτιώνει την απόδοση της ανάκτησης, 2) η στρατηγική προσεγγιστικού ταιριάσματος επιτρέπει την ανάκτηση κειμένων που προσεγγίζουν τις συνθήκες του ερωτήματος, 3) ο τρόπος του υπολογισμού της κατάταξης με βάση το συνημίτονο αφενός μεν επιτρέπει την ταξινόμηση των κειμένων βάσει του βαθμού ομοιότητάς τους με την ερώτηση αφετέρου δε υλοποιείται εύκολα με τις υπάρχουσες δομές δεικτοδότησης. Ένα μειονέκτημα είναι ότι οι όροι δεικτοδότησης θεωρούνται ανεξάρτητοι μεταξύ τους κάτι που το κάνει μη λειτουργικό για την εξαγωγή των αποτελεσμάτων όταν γίνεται η συσχέτιση με τα ερωτήματα του χρήστη.

Τελικά το μοντέλο Vector Space, παρά την απλότητα της σύλληψής και της υλοποίησής του είναι ένα στιβαρό μοντέλο. Η δυνατότητα της εφαρμογής προσεγγιστικού ταιριάσματος, δίνει αποτελέσματα που είναι δύσκολο να βελτιωθούν χωρίς επέκταση του ερωτήματος ή εφαρμογή ανάδρασης χρήστη. Τα αλγεβρικά μοντέλα που ακολούθησαν το Vector Space αν και έχουν κατά σημεία καλύτερη απόδοση, είναι πιο δύσκολα στην υλοποίησή τους. Πάντως το Vector Space δεν αντιμετωπίζει επαρκώς τα προβλήματα της Συνωνυμίας και της Πολυσημίας. Λόγω όμως της ευκολίας στην υλοποίησή του, παραμένει το πιο δημοφιλές μοντέλο ΑΠ.

2.4.4 Το πιθανοτικό μοντέλο

Σ' αυτή την ενότητα παρουσιάζουμε το κλασσικό πιθανοτικό μοντέλο που πρωτοπαρουσιάστηκε από τους Robertson και Sparck Jones, το οποίο αργότερα έγινε γνωστό και ως μοντέλο ανάκτησης δυαδικής ανεξαρτησίας (binary independence retrieval – BIR). Η συζήτηση του μοντέλου είναι σύντομη και σκοπό έχει να τονίσει τα χαρακτηριστικά του μοντέλου και να δώσει την διαίσθηση πίσω από αυτό. Σε αυτό το μοντέλο βασίζεται και πατάει η υλοποίηση του επόμενου κεφαλαίου.

Το πιθανοτικό μοντέλο επιχειρεί να αντιμετωπίσει το πρόβλημα της ΑΠ παρέχοντας ένα πιθανοτικό υπόβαθρο. Η βασική ιδέα είναι η εξής. Δεδομένου ενός ερωτήματος χρήστη, υπάρχει ένα σύνολο κειμένων που αποτελείται ακριβώς από τα σχετικά κείμενα και μόνο απ' αυτά. Σ' αυτό το σύνολο θα αναφερόμαστε με τον όρο ιδανικό σύνολο απάντησης. Δεδομένης της περιγραφής του ιδανικού συνόλου απάντησης, δεν θα είχαμε κανένα πρόβλημα να ανακτήσουμε τα κείμενα που το αποτελούν. Συνεπώς μπορούμε να θεωρήσουμε ότι η διατύπωση ενός ερωτήματος ταυτίζεται με τη διαδικασία καθορισμού των ιδιοτήτων του ιδανικού συνόλου απάντησης (όπως όταν θέλαμε να περιγράψουμε το σύνολο A στο Vector Space). Το πρόβλημά μας είναι ότι δεν γνωρίζουμε ποιες ακριβώς είναι αυτές οι ιδιότητες. Το μόνο που έχουμε στη διάθεσή μας είναι μια ομάδα από όρους δεικτοδότησης, η σημασιολογία των οποίων μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει αυτές τις ιδιότητες. Αυτές οι ιδιότητες δεν είναι γνωστές τη στιγμή της διατύπωσης του ερωτήματος, οπότε πρέπει να γίνει μια αρχική προσπάθεια να προσδιοριστούν. Η αρχική αυτή εκτίμηση μας επιτρέπει να δημιουργήσουμε μια αρχική πιθανοτική περιγραφή του ιδανικού συνόλου απάντησης, η οποία θα χρησιμοποιηθεί για την ανάκτηση ενός πρώτου συνόλου κειμένων. Ακολουθεί αλληλεπίδραση με το χρήστη με σκοπό τη βελτίωση της περιγραφής του ιδανικού συνόλου απάντησης.

Ο χρήστης εξετάζει το αρχικό σύνολο των επιστρεφόμενων κειμένων και αποφασίζει ποια κείμενα είναι σχετικά και ποια όχι (στην πράξη αρκεί η εξέταση λίγων αρχικών κειμένων). Κατόπιν το σύστημα αξιοποιεί αυτή την πληροφορία για να βελτιώσει την περιγραφή του συνόλου απάντησης. Επαναλαμβάνοντας αυτή τη διαδικασία αρκετές φορές, αναμένεται ότι η περιγραφή θα συγκλίνει προς την ιδανική περιγραφή του συνόλου απάντησης. Έτσι πάντα θα πρέπει να έχουμε υπόψη μας την αρχική περιγραφή του ιδανικού συνόλου απάντησης. Επιπλέον πρέπει να γίνει προσπάθεια να περιγραφεί η παραπάνω διαδικασία, πιθανοτικά.

Επάνω σε αυτό θα δούμε στο επόμενο κεφάλαιο ότι αυτή η αλληλεπίδραση του χρήστη με το σύστημα για να έχει ένα καλύτερο σύνολα απάντησης, θα μπορούσε να θεωρηθεί δεδομένη με βάση την γνώση των προτιμήσεων του χρήστη από το ιστορικό του φυλλομετρητή, κάτι όμως που θα δούμε παρακάτω.

Το πιθανοτικό μοντέλο βασίζεται στην ακόλουθη θεμελιώδη υπόθεση. Υπόθεση (Πιθανοτική Αρχή) Δοθέντος ενός ερωτήματος q και ενός κειμένου d_j της συλλογής, το πιθανοτικό μοντέλο προσπαθεί να εκτιμήσει την πιθανότητα ο χρήστης να βρει ενδιαφέρον το κείμενο d_j (δηλ. σχετικό προς το ερώτημα q). Υπόθεση του μοντέλου είναι ότι η πιθανότητα της σχετικότητας εξαρτάται από την αναπαράσταση του ερωτήματος και του κειμένου και μόνο. Επιπλέον γίνεται η υπόθεση ότι υπάρχει ένα υποσύνολο όλων των κειμένων, το οποίο ο χρήστης προτιμά ως απάντηση στο ερώτημα q. Ένα τέτοιο ιδανικό σύνολο απάντησης, ονομάζεται R και θα πρέπει να μεγιστοποιεί τη συνολική πιθανότητα σχετικότητας προς την πληροφοριακή ανάγκη του χρήστη. Τα κείμενα στο R προβλέπεται ότι είναι σχετικά προς το ερώτημα. Τα κείμενα που δεν ανήκουν σ' αυτό το σύνολο προβλέπεται ότι είναι μη-σχετικά.

Μια τέτοια υπόθεση είναι κάπως προβληματική γιατί δεν παρέχει ένα μηχανισμό για τον υπολογισμό των πιθανοτήτων σχετικότητας. Επιπλέον ούτε καν προκύπτει ο δειγματοχώρος για τον υπολογισμό αυτών των πιθανοτήτων.

Δοθέντος λοιπόν ενός ερωτήματος q, το πιθανοτικό μοντέλο αναθέτει σε κάθε κείμενο d_j, την πιθανότητα να είναι σχετικό προς το ερώτημα. Η πιθανότητα αυτή δίνεται από το λόγο P(d_j σχετικό με το q)/P(d_j μη σχετικό με το q). Λαμβάνοντας τον λόγο αυτό ως την συνάρτηση κατάταξης, ελαχιστοποιείται η πιθανότητα λανθασμένης κρίσης.

Ορισμός *Στο πιθανοτικό μοντέλο όλα τα βάρη των όρων δεικτοδότησης έχουν δυαδική μορφή δηλ,* $w_{i,j} \in \{0,1\}$, $w_{i,q} \in \{0,1\}$. *Ένα ερώτημα q είναι ένα υποσύνολο των όρων δεικτοδότησης. Έστω R το σύνολο των κειμένων για το οποία υπάρχει η γνώση (ή αρχικά η εκτίμηση) ότι είναι σχετικά. Έστω* $\overline{R}$ *το συμπλήρωμα του R (δηλ. το σύνολο των μη σχετικών κειμένων). Έστω* $P(R \mid \vec{d}_j)$ *η πιθανότητα το κείμενο* d_j *να είναι σχετικό προς το ερώτημα q και* $P(\overline{R} \mid \vec{d}_j)$ *η πιθανότητα το κείμενο* d_j *να μην είναι σχετικό προς το ερώτημα q. Η ομοιότητα* $sim(d_j, q)$ *του κειμένου* d_j *προς το ερώτημα q ορίζεται ως ο λόγος:*

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\overline{R} \mid \vec{d}_j)}$$

Από τον κανόνα του Bayes,

$$sim(d_j, q) = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \overline{R}) \times P(\overline{R})}$$

είναι η πιθανότητα το d_j να επιλέχθηκε τυχαία από το σύνολο R, δηλαδή να είναι σχετικό. Επιπλέον P(R) είναι η πιθανότητα το κείμενο που επιλέξαμε με τυχαίο τρόπο από ολόκληρη τη συλλογή, να είναι τυχαίο. Οι ερμηνείες των ποσοτήτων $P(d \mid R)_j r$ και P(R) είναι ανάλογες των παραπάνω. Καθώς οι ποσότητες P(R) και P(-R) είναι ίδιες για όλα τα κείμενα της συλλογής, μπορούμε να γράψουμε

$$sim(d_j, q) \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \overline{R})}$$

Λόγω του ότι υποθέσαμε στοχαστική ανεξαρτησία στους όρους μπορούμε να γράψουμε την παραπάνω σχέση ως,

$$sim(d_j, q) \approx \frac{\left(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid R)\right) \times \left(\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid R)\right)}{\left(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid \overline{R})\right) \times \left(\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid \overline{R})\right)}$$

όπου $P(k_i \mid R)$ είναι η πιθανότητα ο όρος δεικτοδότησης k_i να εμφανίζεται σε ένα κείμενο το οποίο επιλέχθηκε τυχαία από το σύνολο R. Ο όρος $P(k_i \mid R)$ δίνει την πιθανότητα ο όρος k_i να μην εμφανίζεται σε ένα κείμενο το οποίο επιλέχθηκε τυχαία από το σύνολο R. Οι πιθανότητες που σχετίζονται με το σύνολο R έχουν ανάλογη σημασία.

Λογαριθμίζοντας και λαμβάνοντας υπόψη μας ότι $P(k_i \mid R) + P(k_i \mid R) = 1$, και αγνοώντας τους παράγοντες που είναι σταθεροί για όλα τα κείμενα για συγκεκριμένο ερώτημα, μπορούμε να γράψουμε,

$$sim(d_j, q) \approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})} \right)$$

ο οποίος είναι ουσιαστικά ο τύπος με τον οποίο υπολογίζουμε την κατάταξη των κειμένων στο πιθανοτικό μοντέλο.

Καθώς δεν γνωρίζουμε το σύνολο R εξ' αρχής, είναι απαραίτητο να ορίσουμε μια μέθοδο υπολογισμού για τις πιθανότητες P(k_i | R) και P(k_i | R) . Παρακάτω θα δούμε με ποιον τρόπο μπορεί να γίνει ο υπολογισμός αυτός.

Αμέσως μετά την διατύπωση του ερωτήματος, δεν υπάρχουν ακόμα ανακτημένα κείμενα. Έτσι πρέπει να κάνουμε απλοποιητικές υποθέσεις σε ότι αφορά τις πιθανότητες. Αυτές οι υποθέσεις είναι: α) υποθέτουμε ότι η P(k_i | R) είναι σταθερή για όλους τους όρους k_i (τυπικά 0.5) και β) υποθέτουμε ότι η κατανομή των όρων δεικτοδότησης στα μη σχετικά κείμενα μπορεί να προσεγγιστεί από την κατανομή των όρων δεικτοδότησης στο σύνολο των κειμένων (με άλλα λόγια, το μέγεθος του συνόλου των μη σχετικών κειμένων R , είναι πολύ μεγαλύτερο F5από το μέγεθος R). Οι δύο παραπάνω υποθέσεις μας δίνουν,

$$P(k_i \mid R) = 0.5$$

$$P(k_i \mid \overline{R}) = \frac{n_i}{N}$$

όπου, όπως ορίστηκε προηγουμένως n_i είναι ο αριθμός των κειμένων που περιέχουν τον όρο k_i και N είναι ο συνολικός αριθμός των κειμένων της συλλογής. Έχοντας την αρχική εκτίμηση, μπορούμε να ανακτήσουμε ένα αρχικό σύνολο κειμένων που περιέχουν όρους που εμφανίζονται στο ερώτημα και να παρέχουμε μια πιθανοτική κατάταξη γι' αυτά. Κατόπιν βελτιώνουμε την αρχική κατάταξη με τον τρόπο που αναφέρουμε στη συνέχεια.

Έστω V ένα υποσύνολο των κειμένων που ανακτήθηκαν αρχικά και στα οποία δόθηκε μια κατάταξη από το πιθανοτικό μοντέλο. Για παράδειγμα το παραπάνω σύνολο θα μπορούσε να είναι τα κορυφαία r κείμενα, όπου το r είναι ένα προκαθορισμένο κατώφλι. Έστω επίσης V_i ένα υποσύνολο του V το οποίο αποτελείται από τα κείμενα που περιέχουν τον όρο k_i. Για λόγους απλότητας, θα χρησιμοποιούμε τους όρους V και V_i για να αναφερόμαστε στους πληθαρίθμους των αντιστοίχων συνόλων. Το πότε θα αναφερόμαστε στο ίδιο το σύνολο ή στο μέγεθός του θα είναι ξεκάθαρο από τα συμφραζόμενα. Για να βελτιώσουμε την πιθανοτική κατάταξη, πρέπει να βελτιώσουμε τις εκτιμήσεις για τα P(k_i|R) και P(k_i | R) . Αυτό επιτυγχάνεται με τις ακόλουθες υποθέσεις: α) μπορούμε να προσεγγίσουμε την P(k_i|R) με την κατανομή του όρου k_i στα κείμενα που ανακτήθηκαν μέχρι στιγμής (σύνολο V), β) μπορούμε να προσεγγίσουμε την P(k_i | R) , αν θεωρήσουμε όλα τα μη ανακτημένα κείμενα ως μη-σχετικά.
Έτσι μπορούμε να γράψουμε,

$$P(k_i \mid R) = \frac{V_i}{V}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}$$

Αυτή η διαδικασία μπορεί να επαναληφθεί αναδρομικά, υπολογίζοντας κάθε φορά νέα V και V_i. Έτσι είμαστε σε θέση να βελτιώσουμε τις εκτιμήσεις μας για τα P(k_i | R) και P(k_i | R) χωρίς καμία εμπλοκή του ανθρωπίνου παράγοντα. Ενδεχομένως όμως να είναι απαραίτητη η ανθρώπινη παρέμβαση στην κατασκευή του συνόλου V.

Οι τελευταίοι δυο τύποι για τα $P(k_i \mid R)$ και $P(k_i \mid R)$ παρουσιάζουν προβλήματα για μικρές τιμές των V και V_i που εμφανίζονται στην πράξη (όπως π.χ. V = 1 και V_i = 0). Για να αντιμετωπιστούν αυτά τα προβλήματα, συνήθως εισάγεται ένας προσθετικός παράγοντας οπότε οι παραπάνω τύποι γράφονται ως:

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0.5}{N - V + 1}$$

Συχνά ένας σταθερός προσθετικός παράγοντας, όπως το 0.5, δεν είναι επαρκής. Μια εναλλακτική λύση είναι να θεωρηθεί ως προσθετικός παράγοντας η ποσότητα n_i/N, που μας δίνει,

$$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

Το κύριο πλεονέκτημα του πιθανοτικού μοντέλου είναι ότι τα κείμενα κατατάσσονται σε φθίνουσα σειρά με βάση την πιθανότητα να είναι σχετικά με το αρχικό ερώτημα. Τα μειονεκτήματα είναι ότι 1) χρειάζεται μια αρχική εκτίμηση για τον διαχωρισμό της συλλογής των κειμένων σε σχετικά και μη, 2) δεν λαμβάνεται υπόψη η συχνότητα εμφάνισης του όρου μέσα σε ένα κείμενο (όλα τα βάρη είναι 0 ή 1), 3) η υιοθέτηση της άποψης ότι οι όροι είναι μεταξύ τους ανεξάρτητοι.

Όπως παρατηρούμε η όλη προσπάθεια είναι να γίνει σωστή κατάταξη των κειμένων με βάση το αρχικό ερώτημα ενός χρήστη, όμως μπορούμε να δούμε το πρόβλημα από μια άλλη σκοπιά. Έστω ότι έχουμε τρεις ανεξάρτητες συλλογές κειμένων Α, Β, Γ αν πάρουμε τους όρους δεικτοδότησης αυτών των κειμένων και τους ομαδοποιήσουμε έτσι ώστε να δούμε συγγενικές σχέσεις μεταξύ τους (χρήση μίας δομής δεδομένων ενός δένδρου συγγενικών μεταξύ τους κόμβων) και αν δώσουμε ένα θετικό βάρος που θα δείχνει πόσο συχνά το έχει επισκεφτεί αυτό ο χρήστης (ουσιαστικά αυτό μπορεί να προκύψει από το ιστορικό του φυλλομετρητή) τότε με βάση κάποιες ενέργειες που θα δούμε παρακάτω, μπορούμε να δώσουμε στο χρήστη καλύτερο αποτέλεσμα στην αναζήτησή του. Αυτή την ιδέα θα την συζητήσουμε στο επόμενο κεφάλαιο αναλυτικότερα.

Κεφάλαιο 3ο

Η ΥΛΟΠΟΙΗΣΗ ΜΟΥ

Εισαγωγή

Στο προηγούμενο κεφάλαιο αναφέραμε τα βασικά μοντέλα της ανάκλησης πληροφορίας, καθώς και τα πλεονεκτήματα τους και που υστερούν. Είδαμε ότι και τα τρία μοντέλα προσπαθούν να κατατάξουν καλύτερα τα κείμενα μιας συλλογής για να δώσουν όσο ένα καλύτερο αποτέλεσμα στα ερωτήματα του χρήστη, χωρίς όμως να δίνουν έμφαση στις σχέσεις μεταξύ των όρων δεικτοδότησης των κειμένων. Ας δούμε παρακάτω πως μπορούμε να δούμε από μία άλλη πλευρά το πρόβλημα αυτό.

3.1 Το πρόβλημα και οι στόχοι

Σε γενικές γραμμές το πρόβλημα που καλούμαστε να λυσουμε είναι:
Με δεδομένο ενα γράφο με βεβαρημένες κορυφές (οπου το βάρος κάθε κορυφής προσδιορίζει το ιστορικό αναφοράς του συγκεκριμένου κόμβου του χρήστη),και με δεδομένο μια ερώτηση (query) του χρήστη καλούμαστε να ανακτήσουμε πληροφορία όσο γίνεται πιο χρησιμη και προσφιλέστερη στις προτιμήσεις του χρήστη, έχοντας ένα βαθμό γνώσης των επιθυμιών του σηγκεκριμένου χρήστη.
Θα αναλύσουμε τα δύο χαρακτηριστικά γνωρίσματα του προβλήματός :

Γράφημα: είναι ένα δένδρο, όπου ο κάθε κόμβος του έχει ένα όνομα το οποίο αντιστοιχεί σε ένα ερώτημα (query) του χρηστη για κάποια πληροφορία που χρειάζεται, εναν αριθμό ο οποίος δείχνει την επισκεψιμότητα (ποσες φορές είχε επισκεφτεί με τουλάχιστον ένα κλικ με το ποντικι την συγκεκριμένη σελιδα, που αφορα αυτό το ερώτημα) και οι σχέσεις μεταξύ τον κόμβων αποτελούν σχέσεις γενίκευσης, στους κόμβους από κάτω προς τα πάνω (π.χ. τατουάζ με φυτά και πιο γενικά φυτά) και αντίστροφα ειδίκευσης. Αυτη η δομή δεδομένων μας δείχνει σχέσεις ανάμεσα στις προτιμήσεις του χρήστη που θα μας βοηθήσουν για την επιλογή εκείνης τις πληροφορίας όπου θα μπορούσε να χρειαστεί ο χρήστης. Το γράφημα αυτό θα μπορούσε να δημιουργείται στατικά εξαγόμενο από διάφορες βάσεις δεδομένων με ομαδοποίηση των πληροφοριών της βάσης σε μια αναπαράσταση δέντρων είτε ένας πιο ενδιαφέρον τρόπον δημιουργίας των γραφημάτων θα ήταν η δυναμική αναπαράσταση τους μέσα από την αναζήτηση του χρήστη, ωστόσο δεν θα αναφερθούμε στο πρόβλημα της κατασκευής τέτοιων γράφων στην παρούσα μελέτη.

Χρήστης: ο χρήστης είναι αυτός που δίνει μια ερώτηση(query) σε ένα browser, αυτή αντιστοιχίζεται σε ένα κόμβο του παραπάνω γραφήματος και ο αλγόριθμος που υλοποίησα του επιστρέφει με την μορφή αποτελέσματος ή ερώτησης προ το χρήστη κάποιους κόμβους που έχει επισκεφτεί προγενέστερα ο χρήστης αρκετές φορές.

3.1.1 Βασικό Θέματα

Ο χρήστης δίνει ένα ερώτημα (query) στο σύστημα (browser) με το οποίο ζητά να του επιστραφεί πληροφορία που πιθανόν να τον ενδιαφέρει άμεσα με βάση τις προηγούμενες προτιμήσεις και αρεσκείες του στο σύστημα. Για παράδειγμα γράφουμε στον browser ένα ερώτημα όπως ¨beatles¨ , εμείς όμως προγενέστερα έχουμε επισκεφτεί αρκετές φορές σελίδες που αναφέρονται στο συγκρότημα beatles, κάτι που γνωρίζει ο browser μεσα από το ιστορικό που κρατά, άρα πρέπει να ερωτηθούμε από τον browser μήπως πιθανόν εννοούσαμε το συγκρότημα και όχι τα έντομα σκαθάρια. Αρα, αν έχουμε ¨γνώση για τον χρήστη¨ μπορούμε να ανακτήσουμε πληροφορίες που χρειάζεται.

Εστιάζουμε την προσοχή μας στα εξής θέματα :

- Στον βαθμό γνώσης του χρήστη απο το σύστημα μέσω κατάλληλων δομών δεδομένων. Αυτο επιτυγχάνεται με την δημιουργία Δομών δεδομένων από το σύστημα μέσω της συχνής του χρήσης από τον χρήστη (ιστορικο αναζήτησης).
- Στην κατάλληλη ομαδοποιηση των πληροφοριών (clustering δεδομένων) έτσι ώστε να δημιουργήσουμε τις δομές δεδομένων που θα μας βοηθήσουν στην ανάκτηση πληροφορίας.
- Στους αλγορίθμους αναζήτησης πληροφοριών και συσχέτισης αυτών για την εξαγωγή δομών δεδομένων που να αποτελεί ανακτημένη πληροφορία για το χρήστη.
- Στην περαιτέρω αναζήτηση για ανάκτηση των πληροφοριών που αφορά τον χρήστη.

Στόχος αυτής της μελέτης είναι η δημιουργία ενός αλγορίθμου ανάκτησης πληροφορίας, όπου θα εξετάσουμε παρακάτω.

3.2 Ενδεικτική υλοποίηση με νοητά βήματα

Υλοποίηση

Δεδομένα : Ο γράφος του σχήματος 1 (όπου κάθε κόμβος έχει k πατέρες και n παιδιά) και η επιλογή ενός ερωτήματος(query) που αναφέρεται σε έναν από τους κόμβους του δένδρου

Ζητούμενο : Η Ανάκτηση πληροφορίας από το δένδρο του σχήματος 1 δηλαδή η επιστροφή τον κόμβων που θα είναι πιο κοντά στις προτιμήσεις του χρήστη. Έχουμε τα εξής βήματα :

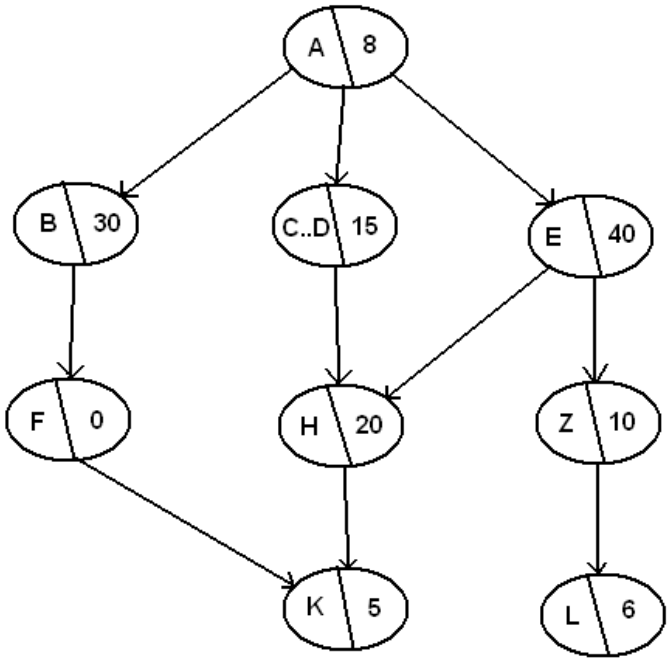
1^{ο} βήμα : Αναζήτηση του κόμβου που έδωσε ο χρήστης για είσοδο. Η αναζήτηση γίνεται με ένα αλγόριθμο Διάταξης Κατά επίπεδο (Level order) πολυπλοκότητας O(n), όπου επισκέπτεται τους κόμβους του γράφου, επίπεδο προς επίπεδο, από πάνω προς τα κάτω και από αριστερά προς τα δεξιά. Η περιγραφή της διαδικασίας δίδεται στον μη αναδρομικό αλγόριθμο Level –Order που αναφέρεται στο Appendix.

2ο βήμα : Επιλογή του επικρατέστερου πατερά. Εφόσον έχουμε βρει το κόμβο που ζητά ο χρήστης στο γράφημα, κοιτάμε ποιους πατεράδες έχει και επιλέγουμε τον επικρατέστερο με βάση το βαθμό επισκεψιμότητάς του (εκείνος με τον μεγαλύτερο βαθμό είναι ο επικρατέστερος). Εφόσον βρεθεί κόμβος με μηδενικό βαθμό επισκεψιμότητάς, τον επιστρέφουμε στο χρήστη ως προτεινόμενο.

3ο βήμα : Επιλογή του επικρατέστερου γιού. Κοιτάμε ποιους γιούς έχει ο κόμβος που βρήκαμε στο 1ο βήμα και επιλέγουμε τον επικρατέστερο με βάση το βαθμό επισκεψιμότητάς του. Εφόσον βρεθεί κόμβος με μηδενικό βαθμό επισκεψιμότητάς, τον επιστρέφουμε στο χρήστη ως προτεινόμενο.

4ο βήμα : Επιλογή του επικρατέστερου αδελφού. Κοιτάμε ποιους γιούς έχει ο επικρατέστερος πατέρας που βρήκαμε στο 1ο βήμα και επιλέγουμε τον επικρατέστερο με βάση το βαθμό επισκεψιμότητάς του. Εφόσον βρεθεί κόμβος με μηδενικό βαθμό επισκεψιμότητάς, τον επιστρέφουμε στο χρήστη ως προτεινόμενο.

5ο βήμα : Επιστροφή των αποτελεσμάτων των βημάτων 2 έως 4 στο χρήστη ως το αποτέλεσμα της αναζήτησής του.



Σχήμα 1

Σε κάθε κόμβο έχουμε δύο ορίσματα:
Το όνομα του κομβου που αντιστοιχίζεται σε ένα ερώτημα του χρήστη (query) /
Το βαθμό επισκεψιμότητας του κάθε κόμβου (εκφρασμένη σε ποσοτικές μονάδες αναφοράς στην σελίδα ή στο ερώτημα αυτό)

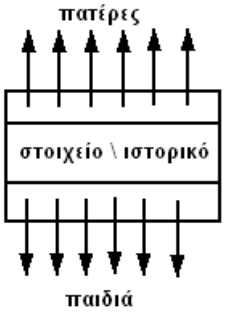
Παράδειγμα τρεξίματος : Έστω ότι ο χρήστης επιλέγει πληροφορία που αναφέρεται στο κόμβο Η, τότε ο παραπάνω αλγόριθμος θα του επιστρέψει τους κόμβους Η, Ε, Κ

και Z ως επιθυμητά αποτελέσματα. Αν επιλέξει τον κόμβο B τότε θα του επιστραφεί ο B, A, F και E ,παρατηρούμε ότι ο F του επιστρέφεται και ας μην τον έχει επιλέξει παλιότερα, αλλά είναι προτεινόμενος σ' αυτόν.

Παρατηρήσεις: Βλέπουμε πως ο αλγόριθμος γνωρίζοντας τις προτιμήσεις του χρήστη μέσα από το ιστορικό του κάθε κόμβου επιλέγει εκείνη την πληροφορία που είναι πιο κοντά στις προτιμήσεις του. Επίσης προτείνει στο χρήστη πληροφορία που σχετίζεται με τις προτιμήσεις του αλλά δεν την έχει επισκεφτεί ποτέ (οι συγγενικοί μηδενικοί κόμβοι του γραφήματος).

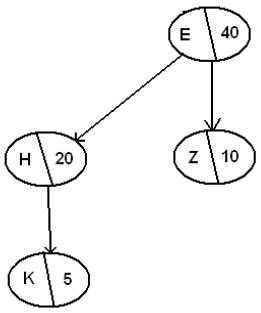
3.3 Περιγραφή του κώδικα σε java*

Στην αρχή του κώδικα ορίζεται επί μίας κλάσεως-κόμβου TNode, η οποία συνιστάται από το στοιχείο element, το ιστορικό του χρήστη history (integer), τον πίνακα δεικτών προς τους πατεράδες parents και τον πίνακα δεικτών προς τα παιδιά sons. Άρα έχουμε ένα αντικείμενο της μορφής:



Στο μεταξύ δημιουργώ δύο μεθόδους όπου βρίσκω, δοθέντος ενός κόμβου τον επικρατέστερο πατέρα του με βάση το ιστορικό του και το παιδί αντίστοιχα. Έτσι αν βρω τον επικρατέστερο πατέρα του αρχικού κόμβου τότε αν καλέσω σε αυτόν τον κόμβο την μέθοδο εύρεσης του επικρατέστερου παιδιού του θα έχω και τον επικρατέστερο αδερφό για τον αρχικό κόμβο, με αυτό τον τρόπο εξάγω ένα υπόδεντρο στον χρήστη σαν αποτέλεσμα της αναζήτησής του όπου του επιστρέφω τον επικρατέστερο πατέρα, παιδί και αδερφό του.

Άρα αν ο χρήστης δώσει σαν είσοδο τον κόμβο H (με βάση το σχήμα 1) το πρόγραμμα θα του επιστρέψει τους κόμβους E Z και K αντίστοιχα.



*ο αντίστοιχος κώδικας βρίσκεται στο appendix

Κεφάλαιο 4ο

ΑΞΙΟΛΟΓΗΣΗ

Συμπεράσματα

Ας κάνουμε μια ανασκόπηση επάνω σε αυτά που είδαμε στην συγκεκριμένη εργασία:

Αρχικά κάναμε μια εισαγωγή όπου αναφέραμε τα κίνητρα που μας έκαναν να ασχοληθούμε με την επιστήμη της ανάκτησης πληροφοριών και ποιο συγκεκριμένα με αλγορίθμους ανάκτησης πληροφορίας μέσα από το διαδίκτυο. Στη συνέχεια κάναμε μια εισαγωγή στο τι είναι; και με τι ασχολείται; η επιστήμη της ανάκλησης πληροφορίας, αναφερθήκαμε σε κάποιες βασικές έννοιες της καθώς περιγράψαμε την διαδικασία της ανάκτησης με την χρήση μιας απλής και γενικευμένης αρχιτεκτονικής λογισμικού.

Σε επόμενο κεφάλαιο είδαμε κάποια χαρακτηριστικά γνωρίσματα του φιλτραρίσματος πληροφοριών και παρουσιάσαμε σχηματικά πως λειτουργεί ένα μοντέλο φιλτραρίσματος πληροφορίας και ένα μοντέλο ανάκτησης και τα συγκρίναμε μεταξύ τους. Ύστερα από την εξέταση κάθε μιας από αυτές της έννοιες, παρατηρήσαμε ότι υπάρχει σχετικά λίγη διαφορά μεταξύ τους σε αφηρημένο επίπεδο (Σε ένα χρήστη είναι αδιάφορη η σύγκριση τους, θεωρούνται ταυτόσημες έννοιες). Καταρχήν, οι στόχοι τους είναι ισοδύναμοι. Δηλαδή και οι δύο ενδιαφέρονται για να δώσουν τις πληροφορίες στους ανθρώπους που τις χρειάζονται, και οι δύο ενδιαφέρονται για το ίδιο είδος πληροφοριών, και το ίδιο είδος περιβάλλοντος αναζήτησης. Έτσι αυτή η σύγκριση έγινε για να προσδιορίσουμε καλύτερα στον αναγνώστη την έννοια της ανάκλησης μέσα από την έννοια του φιλτραρίσματος.

Στην συνέχεια αναλύσαμε κάποια βασικά μοντέλα ανάκλησης και είδαμε ποια είναι τα πλεονεκτήματα και μειονεκτήματα του καθενός. Είδαμε μέσα από κάποιες αδυναμίες τους πως αυτά μας οδήγησαν να σκεφτούμε και να αναπτύξουμε την ιδέα της συγκεκριμένης υλοποίησής.

Έτσι λοιπόν σε μία ποιο ειδική σκοπιά βλέποντας ότι το μοντέλο του πιθανολογικού προτύπου παρουσιάζει αδυναμίες, όπως ότι δεν λαμβάνει υπόψη του τις σχέσεις μεταξύ των όρων των συλλόγων, σκεπτόμαστε να κάνουμε μια ποιο αντιπροσωπευτική αναπαράσταση αυτών τον όρων με την χρήση δομών δεδομένων όπως τον δένδρων για ομαδοποίηση αυτών τον όρων. Με αυτή την ομαδοποίηση πετυχαίνουμε να προσδιορίσουμε καλύτερα την πληροφοριακή ανάγκη του χρήστη. Ακόμα με την χρήση του ιστορικού για να προσδιορίσουμε τα βάρη σε κάθε κόμβο του παραπάνω δένδρου μπορούμε με συγκεκριμένες διαδικασίες που αναπτύξαμε στο προηγούμενο κεφάλαιο να του δώσουμε ακόμα καλύτερα αποτελέσματα στο χρήστη.

Παρακάτω θα εξετάσουμε κάποιες πτυχές αυτής της μελέτης που λόγω περιορισμένου χρόνου δεν μπορέσαμε να αναπτύξουμε και αναφέρονται για μελλοντική μελέτη των ενδιαφερόμενων αναγνωστών.

Μελλοντική μελέτη

Ας δούμε τώρα κάποια θέματα που χρειάζονται περεταίρω μελέτη και ανάπτυξη σε αυτή την έρευνα:

Ομαδοποίηση των όρων: Αναφέραμε στην υλοποίηση ότι γίνεται μια ομαδοποίηση των όρων των κειμένων των συλλογών σε μία δομή δένδρου όπου βασίζεται σε σχέσεις των όρων αυτών. Εδώ έχουμε κάποια μεγάλα ζητήματα και ερωτήματα του πως θα γίνει αυτή η ομαδοποίηση κάτι που ερευνά η επιστήμη του clustering δεδομένων. Οι ιδέες επάνω σε αυτή την σκοπιά είναι οι εξής:

- Μπορεί να γίνει ομαδοποίηση των όρων αυτών σε δένδρα αποφάσεων μέσα από μεγάλες βάσεις δεδομένων. Εμφανίζει μεγάλο ενδιαφέρον αυτός ο τομέας έρευνας λόγω της ταξινόμησης που παρέχουν οι βάσεις δεδομένων και τις σχετικά καλής ταχύτητας εύρεσης δεδομένων.
- Μια άλλη ιδέα που θα μπορούσε να μελετηθεί είναι η δημιουργία και οργάνωση των προτιμήσεων του χρήστη σε δένδρα που θα δημιουργούνται δυναμικά. Έτσι κάθε φορά που θα αναζητεί μια σελίδα ο χρήστης θα σχηματίζεται ένας κόμβος που θα ενσωματώνεται στο δένδρο που θέλουμε για την ανάκτηση, αν ήδη υπάρχει θα αυξάνεται το ιστορικό του κατά ένα(ο βαθμός επισκεψιμότητάς του).Το μόνο ζήτημα είναι ότι σε αυτά τα δένδρα δεν θα έχουμε μηδενικούς κόμβους, αρά δεν θα μπορεί να του προτείνει το σύστημα κάτι παραπάνω από τις προτιμήσεις του.

Αντιστοίχιση των βαρών στο δένδρο: εδώ πρέπει να δούμε με ποιόν τρόπο θα αυξάνουμε τα βάρη του δένδρου κάθε φορά που ο χρήστης αναφέρεται σε μια συγκεκριμένη σελίδα. Βασικά το πρόβλημα είναι η αντιστοίχηση της σελίδας που επιλέγει με τον αντίστοιχο κόμβο του δένδρου (αν υπάρχει).

Ψάξιμο σε παραπάνω επίπεδο για πιο γενική ή ειδική γνώση ανάκτησης: Τα ερωτήματα του χρήστη μπορεί να ζητάνε μια πιο γενική η ειδική απάντηση. Δηλαδή στο δένδρο της υλοποίησης μπορούμε να δούμε ότι αν ανεβούμε επίπεδα ή κατεβούμε αντίστοιχα γενικεύουμε ή ειδικεύουμε το ερώτημα του χρήστη.

Εναλλακτικές υλοποιήσεις του αλγορίθμου σε Java: μια άλλη ιδέα υλοποίησης σε java του δένδρου που αναφέρθηκε στο κεφάλαιο 4 είναι η χρήση δυναμικών λιστών για την παραγωγή των κόμβων και των παιδιών και πατεράδων τους άρα θα μπορούσε να έχουμε χρησιμοποιήσει το όλο δένδρο μια δυναμικών λιστών όπου κάθε κόμβος θα είχε από άλλες δυο δυναμικές λίστες μια για τα παιδιά και μία για τους πατέρες.

Appendix

LEVEL-ORDER (TNode u)

Input: ο κόμβος u του οποίου το υπόδεντρο Tu θα διαπεράσουμε
Output: ο κόμβος που αναζητά ο χρήστης

```
FIFO my Fifo  =  new FIFO();
TNode w;
MyFiFo.enqueue(v);
While (!MyFiFo.isEmpty()) do {
        w = (TNode) MyFiFo.dequeue();
        Έλεγξε αν είναι ο κόμβος που ζητήθηκε
        Foreach παιδι w του u do
                MyFiFo.enqueue(u);
}
End of LEVEL-ORDER
```

Η ΥΛΟΠΟΙΗΣΗ ΜΟΥ ΣΕ JAVA

```
/**
 * @(#)TNode.java
 *
 *
 * @author
 * @version 1.00 2009/5/22
 */
import java.util.*;
import java.lang.Object;
import java.lang.*;

public class TNode1  {



    //private LinkedList parents;
    private Object element;
    private TNode1[] parents;
    private TNode1[] sons;
    public int nofSons;
    public int nofParents;
    private int history;


    // απλός constructor
    public TNode1() {}

    //σύνθετος
    public TNode1(Object o, int h, int nofparents, int nofsons){
          setElement(o);
          setHistory(h);
          parents = new TNode1[nofparents];
          nofParents = nofparents;
          sons = new TNode1[nofsons];
          nofSons = nofsons;
    }


    //επιστρέφει το αποθηκευμένο στοιχείο
    public Object getElement(){
          return element;
    }

    //θέτει το αποθηκευμένο στοιχείο
    public void setElement(Object o){
```

```
        element = o;
}

//επιστρέφει το δείκτη προς το ι-στο παιδί
 public TNode1 getSon(int i) {
        if (i > nofSons-1){
                System.out.println("Δεν υπάρχει τέτοιος γιος...");
                return null;
        }
        return sons[i];
}


 //θέτει το δείκτη προς το ι-στο παιδί ίσο με υ
 public boolean setSon(int i, TNode1 v) {
        if (i > nofSons-1){
                System.out.println("Δεν υπάρχει τέτοιος γιος...");
                return false;
        }
                sons[i] = v;
                return true;
}


  //επιστρέφει το δείκτη προς το ι-στο πατέρα
 public TNode1 getParent(int i) {
        if (i > nofParents-1){
                System.out.println("Δεν υπάρχει τέτοιος πατέρας...");
                return null;
        }
        return parents[i];
}
```

```
//θέτει το δείκτη προς το ι-στο πατέρα ίσο με υ
    public boolean setParent(int i, TNode1 v) {
        if (i > nofParents-1){
                System.out.println("Δεν υπάρχει τέτοιος πατέρας...");
                return false;
        }
                parents[i] = v;
                return true;
    }

    /*public TNode1 getParent() {     //επιστρέφει το δείκτη προς το πατέρα
        return parent;
    }


    public void setParent(TNode1 v){  // θέτει τον δείκτη προς τον πατέρα
ίσο με υ
        parent = v;
    }
    */

    //επιστρέφει τον αριθμό του Ιστορικού του χρήστη
    public int getHistory() {
        return history;
    }

    public void setHistory(int h){
        history = h;
    }

    /*public TNode1 FindIndexChild(Object o){
        //Object element = o;
        TNode1 K;

        for (int i=0; i <= nofSons; i++){

        K = o.getson(i);
        }
        return K;
    }
    */
```

```
public static void main (String arguments[]){

        //Δημιουργεία ενα δενδρο N κόμβων

        //τα αντικειμένα με βάση τον constructor

        TNode1 A = new TNode1("A",8,0,3);
        TNode1 B = new TNode1("B",30,1,1);
        TNode1 C = new TNode1("C",15,1,1);
        TNode1 E = new TNode1("E",40,1,2);
        TNode1 F = new TNode1("F",0,1,1);
        TNode1 H = new TNode1("H",20,2,1);
        TNode1 Z = new TNode1("Z",10,1,1);
        TNode1 K = new TNode1("K",6,2,1);
        TNode1 L = new TNode1("L",5,1,1);
        /////

        //δημιουργεία σχέσεων γονέιου - παιδιών

        A.setSon(0,B);
        A.setSon(1,C);
        A.setSon(2,E);
        B.setSon(0,F);
        C.setSon(0,H);
        E.setSon(0,H);
        E.setSon(1,Z);
        F.setSon(0,K);
        H.setSon(0,K);
        Z.setSon(0,L);
        K.setSon(0,null);
        L.setSon(0,null);
        /////

        //δημιουργεία σχέσεων παιδιών - γονειού

        B.setParent(0,A);
        C.setParent(0,A);
        E.setParent(0,A);
        F.setParent(0,B);
        H.setParent(0,C);
        H.setParent(1,E);
        Z.setParent(0,E);
        K.setParent(0,F);
        }
}
```

Βιβλιογραφία

Η μελέτη αυτή στηρίχτηκε στα εξής ηλεκτρονικά έγγραφα:

1.Inference networks for document retrieval

Source Annual ACM Conference on Research and Development in Information Retrieval

http://portal.acm.org/citation.cfm?id=98006

2.Information filtering and information retrieval: Two sides of the same coin

http://reference.kfupm.edu.sa/content/i/n/information_filtering_and_information_re_297546.pdf

3.On modeling of information retrieval concepts in vector spaces

Source ACM Transactions on Database Systems (TODS)

http://portal.acm.org/citation.cfm?id=22957

4.FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting

http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V0B-49HMWRS-1&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1058062921&_rerunOrigin=google&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=09c643964800fff88a1d38d98d7b1c36

5. mCP Nets: Representing and Reasoning with Preferences of Multiple Agents

http://www.aaai.org/Papers/AAAI/2004/AAAI04-115.pdf

6. Dealing with Incomplete Preferences in Soft Constraint Problems

http://www.math.unipd.it/~frossi/isoft9.pdf

Και επίσης στα εξής συγγράμματα:

1. Ι.Βλαχάβας,Π.Κεφαλάς,Ν.Βασιλειάδης,Φ.Κόκκορας,Η.Σακελλαρίου,*Τεχνιτή Νοημοσύνη, 7η Έκδοση*

2. S.Russel, P.Norvig, *Τεχνητή Νοημοσύνη:Μια σύγχρονη Προσέγγιση, 2η Έκδοση*