ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΚΟ ΙΔΡΥΜΑ ΠΑΤΡΩΝ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

Τμήμα Επιχειρηματικού Σχεδιασμού & Πληροφοριακών Συστημάτων

Τεχνικές Ταξινόμησης Αποτελεσμάτων Μηχανών Αναζήτησης με βάση την Ιστορία του Χρήστη

Πτυχιακή εργασία

του

Χρήστου Κιτσάκη

Επιβλέπων : κ.Μάνδαλος Λουκάς

Πάτρα Μάιος 2012

ΠΡΟΛΟΓΟΣ

Η εξατομίκευση των αποτελεσμάτων των μηχανών αναζήτησης είναι ολοένα και πιο απαραίτητη στις μέρες μας, όπου η χρήση του Διαδικτύου και των μηχανών αναζήτησης αυξάνει με μεγάλους ρυθμούς. Είναι φανερό ότι η ενιαία κατάταξη των αποτελεσμάτων για όλους τους χρήστες δεν είναι η καλύτερη προσέγγιση και η εργασία μας απευθύνεται ακριβώς στο θέμα αυτό. Η λύση που προτείνουμε είναι ένα μοντέλο που εκπαιδεύεται από το ιστορικό των κλικ του χρήστη της μηχανής αναζήτησης, και καταγράφει τι χαρακτηριστικά έχουν τα έγγραφα που προτιμάει ο χρήστης. Χρησιμοποιώντας το μοντέλο αυτό, μπορούμε να παρουσιάσουμε τα αποτελέσματα της μηχανής αναζήτησης με κατάταξη βασισμένη στις προτιμήσεις του χρήστη. Η καταγραφή των ερωτημάτων που θέτει ο χρήστης, των αποτελεσμάτων που του παρουσιάζονται, καθώς και των κλικ που κάνει, γίνονται στο παρασκήνιο χωρίς να επιβαρύνουν το χρήστη. Με τον τρόπο αυτό, μπορούμε να μαζέψουμε μεγάλο όγκο πληροφοριών με μικρό κόστος. Καταγράφουμε τα δεδομένα αυτά σε ένα ευρετήριο για να μπορέσουμε να τα επεξεργαστούμε πιο εύκολα και να υπολογίσουμε τα χαρακτηριστικά τους. Επίσης, εκφέρουμε κάποιες σχετικές προτιμήσεις από τις επιλογές που κάνει ο χρήστης όταν αγνοεί εσκεμμένα κάποια αποτελέσματα και προτιμά να πατήσει σε κάποια άλλα. Όλα τα παραπάνω εισάγονται στον αλγόριθμο εκπαίδευσης Support Vector Machine και έτσι δημιουργείται το μοντέλο που θα χρησιμοποιήσουμε για την ανακατάταξη. Όταν ο χρήστης κάνει το ερώτημα του στη μηχανή αναζήτησης, έχει πλέον τη δυνατότητα να ζητήσει αναταξινόμηση των αποτελεσμάτων με βάση το εκπαιδευμένο μοντέλο. Κατόπιν αναλύονται τα χαρακτηριστικά των αποτελεσμάτων της μηχανής αναζήτησης και υπολογίζεται από τον αλγόριθμο κατάταξης Support Vector Machine τι θέση πρέπει να έχουν τα αποτελέσματα αυτά, με βάση τις προτιμήσεις που έχουν καταχωρηθεί στο μοντέλο. Έτσι τα αποτελέσματα παρουσιάζονται στο χρήστη με κατάταξη εξατομικευμένη στις προτιμήσεις του.

ΠΕΡΙΛΗΨΗ

Στο κεφάλαιο 2 θα δούμε τη συμπεριφορά του χρήστη μηχανών αναζήτησης μέσα από παγκόσμιες έρευνες.

Στο κεφάλαιο 3 θα κάνουμε μια εισαγωγή στις μηχανές αναζήτησης, τα χαρακτηριστικά και τον τρόπο λειτουργιά τους.

Στο κεφάλαιο 4 αναλύονται τα συγκεράσματα που μπορούμε να εξάγουμε από τον τρόπο συμπεριφοράς των χρηστών μηχανών αναζήτησης και γίνεται μια θεωρητική εισαγωγή στα συστήματα που θα χρησιμοποιήσουμε.

Το κεφάλαιο 5 εξηγεί τη διαδικασία της καταγραφής της δραστηριότητας του χρήστη της μηχανής αναζήτησης, καθώς και τη δημιουργία ευρετηρίου από τις πληροφορίες αυτές.

Στο κεφάλαιο 6 αναλύεται η διαδικασία εξαγωγής των προτιμήσεων του χρήστη και η εκπαίδευση του μοντέλου που θα χρησιμοποιηθεί για την ανακατάταξη των αποτελεσμάτων.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Σκοπός της εργασίας είναι η εξέταση των μηχανών αναζήτησης από τη σκοπιά της ταξινόμησης αποτελεσμάτων με βάση την ιστορία του χρήστη και η ανάπτυξη τεχνικών που θα βελτιώσουν τον τρόπο ταξινόμησης. Οι περισσότερες μηχανές αναζήτησης χρησιμοποιούν μεθόδους για την κατάταξη των αποτελεσμάτων έτσι ώστε να εμφανίζουν στην κορυφή τα καλύτερα αποτελέσματα.

Καθώς η χρήση του Διαδικτύου και συνεπώς και των χρηστών μηχανών αναζήτησης συνεχώς αυξάνεται, είναι φανερό ότι η παραδοσιακή μέθοδος της ενιαίας κατάταξης για όλους τους χρήστης δεν είναι αρκετά ικανοποιητική. Η εξατομίκευση των αποτελεσμάτων που πραγματοποιεί η εφαρμογή μας είναι μία μέθοδος που απευθύνεται σε αυτό το πρόβλημα των μηχανών αναζήτησης. Το ιστορικό περιλαμβάνει μόνο δεδομένα τύπου clickstream, δηλαδή την ακολουθία των κλικ που ο χρήστης έκανε κατά τη διάρκεια χρήσης της μηχανής αναζήτησης. Αναλύοντας τα δεδομένα αυτά, εξάγονται κάποια συμπεράσματα για τις προτιμήσεις του χρήστη. Ο χρήστης δε χρειάζεται να δηλώσει ρητά τις προτιμήσεις του, αλλά τις εξάγουμε έμμεσα, βασιζόμενοι στο γεγονός ότι κάνει κλικ σε κάποια αποτελέσματα ενώ εσκεμμένα αγνοεί κάποια άλλα που εμφανίζονται υψηλότερα στην αρχική κατάταξη. Παράλληλα καταγράφονται και τα χαρακτηριστικά των αποτελεσμάτων που εμφανίζονται στο χρήστη, ανεξαρτήτως αν τα επισκέφτηκε ή όχι. Τα χαρακτηριστικά αυτά περιλαμβάνουν το βαθμό ομοιότητας του ερωτήματος με το αποτέλεσμα, το domain της ιστοσελίδας καθώς και την κατάταξη την οποία τους δίνει το Google. Τα παραπάνω δεδομένα εισάγονται σε έναν ειδικό αλγόριθμο ταξινόμησης, ο οποίος δημιουργεί ένα μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να “μαντέψει” τις προτιμήσεις του χρήστη για μελλοντικά ερωτήματα. Έτσι χρησιμοποιώντας το μοντέλο αυτό μπορούμε να ανακατατάξουμε τα αποτελέσματα που παρέχει το Google με ένα τρόπο προσωποποιημένο στις προτιμήσεις του χρήστη.

Κεφαλαιο 1 :Έρευνα

1.1 Έρευνες για τη συμπεριφορά όσων χρησιμοποιούν μηχανές αναζήτησης

Δύο νέες έρευνες, η μία από την Αμερική και η άλλη από την Αγγλία, εξετάζουν την συμπεριφορά των χρηστών που χρησιμοποιούν τις μηχανές αναζήτησης και καταλήγουν σε δύο πολύ σημαντικά συμπεράσματα: α) όσα websites δεν έχουν καλή θέση, τουλάχιστον στις 3 πρώτες σελίδες αποτελεσμάτων, για τις σχετικές αναζητήσεις, χάνουν ένα πολύ μεγάλο αριθμό ενδιαφερόμενων υποψήφιων πελατών και β) οι χρήστες γίνονται ολοένα και περισσότερο απαιτητικοί για την ποιότητα των αποτελεσμάτων αλλά επιμένουν και περισσότερο.

Iprospect – Αμερική

Η πρώτη έρευνα με τίτλο «The iProspect Search Engine User Behavior Study» πραγματοποιήθηκε από την εταιρία Jupiter Research ως συνέχεια αντίστοιχων ερευνών που πραγματοποιήθηκαν κατά τα έτη 2002 και 2004.

Τα σημαντικότερα ευρήματα της έρευνας είναι τα ακόλουθα:

Η σημασία της εμφάνισης στις πρώτες θέσεις αποτελεσμάτων

Όταν οι συμμετέχοντες στην έρευνα ερωτήθηκαν πόσες περίπου καταχωρίσεις εξετάζουν, από αυτές που εμφανίζονται στα αποτελέσματα που παρουσιάζουν οι μηχανές αναζήτησης, κατά την διάρκεια της πραγματοποίησης μιας αναζήτησης, απάντησαν ότι (με δυνατότητα μιας απάντησης):

Απάντηση			
Ελάχιστες (από τις πρώτες)	16%	24%	23%
Την πρώτη σελίδα	32%	36%	39%

Τις πρώτες 2 σελίδες	23%	20%	19%
Τις πρώτες 3 σελίδες	10%	8%	9%
Περισσότερες από 3 σελίδες	19%	13%	10%

Όπως βλέπουμε από τα αντίστοιχα ποσοστά, 62 % των χρηστών επιλέγουν κάποιο website από αυτά που εμφανίζονται στην πρώτη σελίδα αποτελεσμάτων, 90 % των χρηστών επιλέγει μία από τις καταχωρίσεις που βρίσκονται έως και την τρίτη σελίδα ενώ μόνο ένα 10 % εξετάζει τα αποτελέσματα πέραν τις τρίτης σελίδας.

Αυτό το οποίο έχει επίσης ιδιαίτερη σημασία είναι ότι με το πέρασμα των χρόνων ολοένα και περισσότεροι χρήστες επιλέγουν κάποιο από τα websites της πρώτης σελίδας αποτελεσμάτων, είτε αυτό εμφανίζεται στα φυσικά αποτελέσματα είτε στις πληρωμένες καταχωρίσεις αναδεικνύοντας για ακόμη μια φορά τη σημασία που έχει το search engine marketing.

Οι χρήστες επιμένουν μέχρι να βρουν αυτό που ζητούν

Ποια είναι η συμπεριφορά των χρηστών σε σχέση με την ποιότητα των αποτελεσμάτων;

Το 41 % των χρηστών που δεν βρίσκουν ικανοποιητικά αποτελέσματα στην πρώτη σελίδα αποτελεσμάτων, είτε αλλάζουν μηχανή αναζήτησης είτε προσπαθούν ξανά στην ίδια μηχανή χρησιμοποιώντας μια ποιο συγκεκριμένη φράση. Το ποσοστό αυτό πριν από 4 χρόνια ήταν 28%.Το 82 % των χρηστών που δεν βρίσκουν αυτό που ζητούν στα αποτελέσματα μιας μηχανής αναζήτησης, κατά την διάρκεια της πρώτης τους προσπάθειας, προσπαθούν ξανά στην ίδια μηχανή αναζήτησης χρησιμοποιώντας μια ποιο συγκεκριμένη φράση. Μόνο ένα 3 % εγκαταλείπει πλήρως την διαδικασία της αναζήτησης εάν δεν βρει ικανοποιητικά αποτελέσματα.

Branding και η θέση εμφάνισης ενός website

Τα αποτελέσματα της έρευνας έδειξαν ότι το 36 % των χρηστών θεωρεί ότι οι εταιρίες τα websites των οποίων εμφανίζονται στις πρώτες θέσεις αποτελεσμάτων, είναι κορυφαίες στον τομέα τους. Το 39 % έχει μια ουδέτερη αντίδραση στην ερώτηση αυτή ενώ μόνο το 25 % των

χρηστών θεωρεί ότι η εμφάνιση στα πρώτα αποτελέσματα δεν έχει καμιά σχέση με την ποιότητα ή την αναγνωρισιμότητα μιας εταιρίας.

Harvest Digital – Αγγλία

Η δεύτερη έρευνα πραγματοποιήθηκε από τη εταιρία Harvest Digital στην Αγγλία και αφορά την συμπεριφορά των «έμπειρων» χρηστών σε σχέση με τις μηχανές αναζήτησης.

Η συμπεριφορά των έμπειρων (Άγγλων) χρηστών

Κυρίαρχη θέση στις προτιμήσεις (και) των Άγγλων έχει το Google, αλλά μόνο ένα 24 % των χρηστών επιλέγει και χρησιμοποιεί μονάχα μια μηχανή αναζήτησης. Είναι μάλιστα ιδιαίτερα εντυπωσιακό ότι 20 % των χρηστών χρησιμοποιεί, σε τακτά χρονικά διαστήματα, 4 ή περισσότερες μηχανές αναζήτησης.

Είναι προφανές ότι οι Άγγλοι παρά το γεγονός ότι βασίζονται σε πολύ μεγάλο βαθμό στις μηχανές αναζήτησης, δεν εμπιστεύονται ιδιαίτερα τα αποτελέσματα τους. Μονό ένα 22 % των χρηστών δήλωσε ότι είναι σίγουροι για το γεγονός ότι οι μηχανές αναζήτησης θα τους παρέχουν πάντοτε τις πληροφορίες που χρειάζονται.

Πάντως, οι Άγγλοι χρήστες θεωρούν ότι το πρόβλημα οφείλεται σε αυτούς και όχι στις μηχανές αναζήτησης με το 36 % να δηλώνει ότι δεν χρησιμοποιεί τις σωστές φράσεις κλειδιά όταν πραγματοποιεί μια αναζήτηση και το 32 % ότι ψάχνει για πληροφορίες οι οποίες είναι πολύ εξειδικευμένες. Μόλις ένα 8% δηλώνει ότι οι μηχανές αναζήτησης είναι αναποτελεσματικές.

Το υπόλοιπο 24 % κατηγορεί τους διαφημιστές που ασχολούνται με το search engine marketing χωρίς όμως να είναι ξεκάθαρο εάν η συμπεριφορά αυτή σχετίζεται με τις πληρωμένες καταχωρίσεις ή την αντίληψη των χρηστών ότι οι διαφημιστές πληρώνουν για να αποκτήσουν μια υψηλή θέση στα «φυσικά αποτελέσματα» των μηχανών αναζήτησης.

Και σε αυτή την έρευνα αναδεικνύεται η σημασία του search engine marketing καθώς το 43 % των χρηστών δηλώνει ότι ο σημαντικότερος λόγος για να επισκεφθούν ένα website είναι η παρουσία του στην πρώτη σελίδα των αποτελεσμάτων. Το 32 % λέει ότι βασικό κριτήριο αποτελεί η περιγραφή του website, η οποία θα πρέπει να είναι σχετική με το αντικείμενο της

αναζήτησης, ενώ ένα 17 % θεωρεί ως το ποιο σημαντικό κριτήριο την παρουσία στην κορυφή της πρώτης σελίδας.

1.2 Google: Οι πλούσιοι ψάχνουν το μέλλον κ οι φτωχοί το παρελθόν

Σε ακριβώς αντίθετη κατεύθυνση στον χρόνο είναι **προσανατολισμένες οι αναζητήσεις των χρηστών στο Google**, ανάλογα με το αν αυτοί προέρχονται από **πλούσιες ή από πιο φτωχές χώρες**. Μία νέα πρωτότυπη βρετανο-αμερικανική επιστημονική έρευνα κατέληξε για πρώτη φορά στο συμπέρασμα ότι **η online συμπεριφορά των χρηστών στις μηχανές αναζήτησης εξαρτάται από το επίπεδο του** ανά κεφαλή Ακαθάριστου Εγχώριου Προϊόντος (ΑΕΠ) στη χώρα τους. Όσο πιο **ψηλό είναι το ΑΕΠ ανά κεφαλή, τόσο πιο πολύ οι ερωτήσεις και αναζητήσεις αφορούν το μέλλον,** ενώ όσο πιο χαμηλό είναι το ΑΕΠ ανά κεφαλή, τόσο πιο πολύ οι πληροφορίες **που οι χρήστες ψάχνουν, αφορούν το παρελθόν.**

Οι ερευνητές, με επικεφαλής τον καθηγητή Στίβεν Μπίσοπ του Τμήματος Μαθηματικών του University College του Λονδίνου (UCL), που έκαναν τη σχετική δημοσίευση στο επιστημονικό περιοδικό "Scientific Reports", σύμφωνα με το New Scientist, ανακάλυψαν ότι

υπάρχει σαφής συσχέτιση ανάμεσα στα online ενδιαφέροντα των χρηστών στο Google και στο **επίπεδο ανάπτυξης και πλούτου στη χώρας καταγωγής τους.**

Οι επιστήμονες, με τη βοήθεια του "Google Trends", εξέτασαν **45 δισεκατομμύρια αναζητήσεις που έκαναν οι χρήστες από 45 διαφορετικές χώρες το 2010** και υπολόγισαν την ποσοστιαία αναλογία των αναζητήσεων που αφορούσαν το επόμενο έτος (2011), καθώς και το προηγούμενο (2009). Δημιούργησαν έτσι ένα νέο **«δείκτη προσανατολισμού προς το μέλλον»** και διαπίστωσαν μία σαφή τάση στις χώρες με υψηλότερο ΑΕΠ/κεφαλή οι **χρήστες να «κοιτάζουν» προς το μέλλον στις online αναζητήσεις τους.**

Ενδεικτικά αναφέρεται ότι η **Ρωσία έχει δείκτη μόνο 0,6, η Ιταλία 1**, ενώ οι ακόμα **πλουσιότερες Γαλλία, Γερμανία και Βρετανία πολύ μεγαλύτερο, γύρω στο 2**. Η ίδια σχέση ισχύει και όσον αφορά τις αναζητήσεις του 2010 που αφορούσαν ακόμα πιο πίσω στο παρελθόν (2008) ή ακόμα πιο μπροστά στο μέλλον (2012).

Οι ερευνητές ανέφεραν ότι η **οικονομική ανάπτυξη μιας χώρας ωθεί τους πολίτες να αναζητούν πληροφορίες για το μέλλον** και, αντίστροφα, αυτή η στραμμένη προς το μέλλον online αναζήτηση ενθαρρύνει τη **διαδικασία της οικονομικής ανάπτυξης και της δημιουργίας εισοδημάτων και πλούτου στις χώρες τους.**

Κεφάλαιο 2 : Μηχανές αναζήτησης (search engines)

2.1 Εισαγωγή στις μηχανές αναζήτησης

Ένα από τα σημαντικότερα χαρακτηριστικά του Διαδικτύου (Internet) είναι η ευκολία που παρέχει στην είσοδο οποιασδήποτε πληροφορίας, επιτρέποντας στους χρήστες του να εισάγουν στοιχεία για κάθε θέμα. Τα στοιχεία αυτά είναι συνήθως ελεύθερα διαθέσιμα σε όλους τους χρήστες, καθιστώντας έτσι το Διαδίκτυο στο σύνολό του μία μοναδική πηγή πληροφόρησης και εύρεσης στοιχείων, που παρόμοιά της δεν υπήρξε ποτέ μέχρι τώρα στην πορεία της ανθρωπότητας.

Η ραγδαία αύξηση της χρήσης του Παγκόσμιου Ιστού (World Wide Web), αλλά και των υπόλοιπων υπηρεσιών του δικτύου, έδωσε στους χρήστες τη δυνατότητα να αποκτήσουν εύκολη πρόσβαση στην πληροφορία, αλλά παράλληλα και τη δυνατότητα παροχής στο δίκτυο όλων όσων αυτοί θεωρούν κατάλληλα. Ενώ όμως η πληθώρα πληροφοριών λογικά θα έπρεπε να είναι ευεργετική για τους χρήστες, οι οποίοι έχουν πλέον στη διάθεσή τους έναν τεράστιο όγκο στοιχείων, αυτή η ίδια πληθώρα προξενεί ένα σημαντικό πρόβλημα, που δεν είναι άλλο από το ότι οι χρήστες αδυνατούν τις περισσότερες φορές να εντοπίσουν τα σημεία εκείνα του δικτύου που περιέχουν τις πληροφορίες τις οποίες αυτοί χρειάζονται. Μολονότι όλο και κάποιον τρόπο μπορεί να σκεφτεί ένας χρήστης για να το επιτύχει, κανένας τρόπος δεν μπορεί να συγκριθεί σε πληρότητα, ταχύτητα και αποτελεσματικότητα με την χρήση των περίφημων μηχανών αναζήτησης (search engines) του Παγκόσμιου Ιστού.

Στο Διαδίκτυο υπάρχουν αρκετές μηχανές αναζήτησης, οι οποίες τις περισσότερες φορές ξεκίνησαν από πειραματικά ερευνητικά προγράμματα (projects) και εξελίχθηκαν σε ολόκληρες εταιρείες, ενώ από πλευράς χρήσης εξυπηρετούν χιλιάδες χρήστες καθημερινά. Συνήθως, η παροχή των προσφερόμενων υπηρεσιών γίνεται δωρεάν, αν και ορισμένες μηχανές επιβάλλουν κάποιους περιορισμούς στη δωρεάν χρήση διαθέτοντας και πρόσβαση επί πληρωμή.

2.2 Τι είναι Μηχανές αναζήτησης;

Οι μηχανές αναζήτησης είναι προγράμματα που επιτρέπουν την αναζήτηση με λέξεις κλειδιά (keywords) σε τεράστιες βάσεις δεδομένων αρχείων του διαδικτύου. Είναι τα περισσότερο διαδεδομένα μέσα για την εύρεση πληροφορίας στο Ίντερνετ και αποτελούν μια αποτελεσματική μέθοδο για προσέλκυση επισκεπτών στο δικτυακό τόπο μιας επιχείρησης. Έρευνες έδειξαν ότι ένα πολύ μεγάλο ποσοστό των πελατών – καταναλωτών χρησιμοποιούν τις Μηχανές Αναζήτησης για να εντοπίσουν μια ιστοσελίδα με περιεχόμενο που τους ενδιαφέρει. Οι μηχανές αναζήτησης διαθέτουν βάσεις δεδομένων που περιλαμβάνουν ευρετήριο με το πλήρες κείμενο των ιστοσελίδων. Όταν ένας χρήστης χρησιμοποιεί μία μηχανή αναζήτησης, στην πραγματικότητα ερευνά τη βάση δεδομένων των καταχωρημένων ιστοσελίδων (και όχι το ίδιο το WWW). Όταν ψάχνουμε στο Διαδίκτυο χρησιμοποιώντας μια Μηχανή Αναζήτησης, αναζητούμε κατά κάποιο τρόπο ένα παλιό αντίγραφο της πραγματικής ιστοσελίδας, όπως αυτό υπάρχει στη βάση δεδομένων της μηχανής. Οι βάσεις δεδομένων των μηχανών αναζήτησης είναι ρυθμισμένες ώστε να δίνουν ταχύτατα αποτελέσματα, πράγμα το οποίο θα ήταν αδύνατο να συμβεί αν οι μηχανές προσπαθούσαν να ερευνήσουν τα δισεκατομμύρια ιστοσελίδων σε πραγματικό χρόνο. Όταν κάνουμε «κλικ» πάνω στους συνδέσμους (links) που παρέχονται από τα αποτελέσματα αναζήτησης της Μηχανής, ανακτούμε από τον server την τωρινή έκδοση της σελίδας.

2.3 Ορισμός της μηχανής αναζήτησης

Μία μηχανή αναζήτησης (search engine) θα μπορούσε να οριστεί ως το εργαλείο που επιτρέπει να εξερευνήσει κανείς τις βάσεις δεδομένων οι οποίες περιέχουν το κείμενο δεκάδων εκατομμυρίων ιστοσελίδων.

Ένας άλλος ορισμός για τη μηχανή αναζήτησης είναι ότι είναι ένα πρόγραμμα σχεδιασμένο ώστε να επιτρέπει τον εντοπισμό και την πρόσβαση σε αρχεία αποθηκευμένα σε έναν υπολογιστή, για παράδειγμα σε έναν κοινό διακομιστή (server) στο Διαδίκτυο ή σε έναν άλλον, ανεξάρτητο και μεμονωμένο υπολογιστή.

2.4 Σύντομη ιστορική ανασκόπηση των μηχανών αναζήτησης

Από τα πρώτα βήματα του internet μέχρι και το 1993 το FTP (File Transfer Protocol) ήταν ο πιο διαδεδομένος τρόπος ανταλλαγής αρχείων μεταξύ των χρηστών. Η 10η Σεπτεμβρίου του 1990 έμελλε να είναι γνωστή ως η ημερομηνία εισαγωγής της έννοιας των μηχανών αναζήτησης στο Internet. Ο Peter Deutsch μαζί με τους Alan Emtage και Bill Heelan όλοι τους φοιτητές του πανεπιστημίου Mc Gill στον Καναδά ανήγγειλαν στο Usenet [Το USENET δημιουργήθηκε στα τέλη της δεκαετίας του 1970 (1979), ως ένα άτυπο μέσο διανομής ειδήσεων για το Λειτουργικό Σύστημα UNIX . Ήταν ειδικότερα κάτι σαν πίνακας ανακοινώσεων (bulletin board) μεταξύ δύο πανεπιστημίων στη Βόρεια Καρολίνα των Η.Π.Α.] τη λειτουργία του Archie, καλώντας τους χρήστες του δικτύου να το χρησιμοποιήσουν. Το Archie (συντομογραφία του Archiver, αρχειοθέτης) ήταν ένα σύστημα καταγραφής, σε καθημερινή βάση, των περισσότερων διακομιστών FTP που λειτουργούσαν, καθώς και των αρχείων που αυτοί περιλάμβαναν.
Το 1991 δημιουργήθηκε στο Πανεπιστήμιο της Μινεσότα από τους Mark McCahill, Farhad Anklesaria, Paul Lindner, Dan Torrey, και Bob Alberti ένα νέο πρωτόκολλο, το Gopher (παραφθορά του “go for”, ήταν κάτι σαν το εμβρυακό στάδιο του Web), το οποίο χρησίμευε για την κατηγοριοποίηση και την παρουσίαση των εγγράφων ενός διακομιστή. Το 1992 στο Πανεπιστήμιο της Νεβάδα, αναπτύχθηκε από τους Steven Foster και Fred Barrie η Veronica (Very Easy RodentOriented Netwide Index to Computer Archives), μια μηχανή αναζήτησης που χρησιμοποιούσε το πρωτόκολλο Gopher. Σύντομα παρουσιάστηκε η Jughead (Jonzy’s Universal Gopher Hierarchy Excavation and Display), μια άλλη μηχανή αναζήτησης που χρησιμοποιούσε και αυτή το Gopher.

Το 1994 ήταν η εποχή του WebCrawler, πνευματικού παιδιού του Brian Pinkerton, φοιτητή του Πανεπιστημίου Ουάσινγκτον. Το WebCrawler ήταν η πρώτη μηχανή που κατέγραφε ολόκληρο το περιεχόμενο των σελίδων που επισκεπτόταν. Το 1995 έκανε την εμφάνισή του το Excite, δημιούργημα έξι φοιτητών του Πανεπιστημίου του Στάνφορντ, που βασίστηκε σε στατιστικές αναλύσεις σχετικές με τη συγγένεια των λέξεων.

Επόμενοι σταθμοί εξέλιξης αποτελούν οι μηχανές Lycos και AltaVista, με την πρώτη να καλύπτει τον εντυπωσιακά μεγάλο αριθμό σελίδων της εποχής (60 εκατ.,1996). Η AltaVista έμεινε γνωστή για τις αμείωτες επιδόσεις της, παρά τα εκατομμύρια των επισκέψεων που

δεχόταν καθημερινά. Επιπρόσθετα το 1996, το MetaCrawler ήρθε να εισαγάγει την έννοια των μεταμηχανών, δηλαδή μηχανές σε ρόλο διαμεσολαβητή, οι οποίες μεταβιβάζουν τα ερωτήματα του χρήστη σε πλήθος “πραγματικών” μηχανών αναζήτησης και επιστρέφουν τα συγκεντρωτικά αποτελέσματα. Το 1998, δύο φοιτητές του Στάνφορντ, οι Larry Page και Sergey Brin ανέτρεψαν τα δεδομένα και εφάρμοσαν ένα προηγμένο σύστημα αξιολόγησης των δικτυακών τόπων, τη μηχανή αναζήτησης Google.

2.5 Τα μέρη μιας μηχανής αναζήτησης

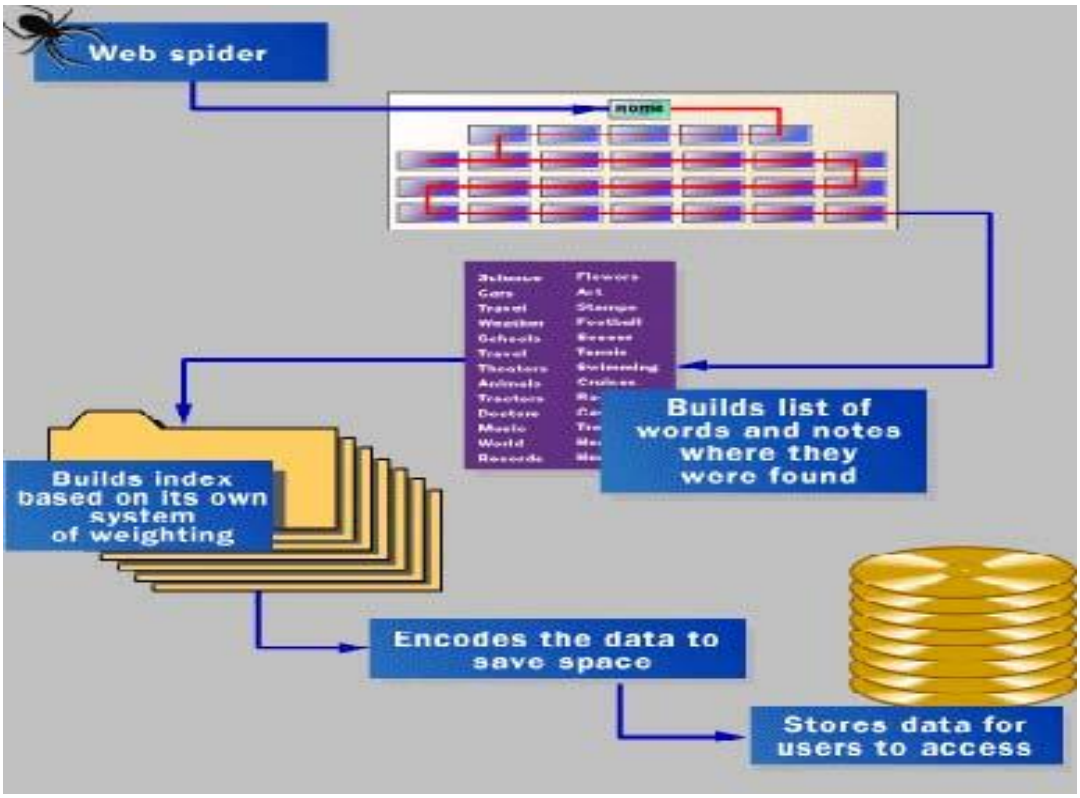
Μια μηχανή αναζήτησης αποτελείται από τρία βασικά μέρη:

Α. Η αράχνη

Τα προγράμματα αράχνες (ή Spider ή Robot) είναι οι ανιχνευτές των μηχανών αναζήτησης. Η αποστολή τους είναι να βρίσκουν και να ανακτούν ιστοσελίδες στο διαδίκτυο και να τις μεταβιβάζουν στο ευρετήριο της μηχανής αναζήτησης. Παρά το ότι το όνομα των αραχνών υπονοεί ότι ταξιδεύουν πάνω στον παγκόσμιο ιστό στη πραγματικότητα η λειτουργία τους είναι περίπου όμοια με έναν περιηγητή, που στέλνει αίτηση για μία ιστοσελίδα, κατεβάζει την ιστοσελίδα και τη διαβιβάζει στον μηχανισμό του ευρετηρίου. Φυσικά οι αράχνες ζητούν και διαβάζουν τις ιστοσελίδες πολύ γρηγορότερα από έναν περιηγητή. Στη πραγματικότητα οι περισσότερες αράχνες ζητούν ταυτόχρονα εκατοντάδες ακόμη και χιλιάδες διαφορετικές ιστοσελίδες. Εξαιτίας αυτής της δυνατότητας τους οι αράχνες είναι προγραμματισμένες να κατανέμουν τις αιτήσεις τους σε πολλούς διακομιστές ώστε να μην κατακλύζουν έναν διακομιστή με τις αιτήσεις τους και να μην καταλαμβάνουν μεγάλο κομμάτι του εύρους ζώνης, ώστε να μην μπορούν να εξυπηρετηθούν οι χρήστες**.**

Περισσότερα για τις αράχνες (Spiders) ή robot ή crawlers

Με την τρομακτική αύξηση του διαδικτύου έγινε πάρα πολύ δύσκολο να καταγράφονται όλες οι νέες ιστοσελίδες που εμφανιζόντουσαν κάθε μέρα. Η ιδέα του Wanderer υιοθετήθηκε από πολλούς προγραμματιστές ώστε να δημιουργήσουν robot ή spiders (αράχνες) ή crawlers όπως επικράτησε να ονομάζονται. Το robot, είναι ένα λογισμικό που εξετάζει την hypertext κατασκευή του Web ανακτώντας το κείμενο της ιστοσελίδας και ανακτώντας περιοδικά (μία φορά στους έξι μήνες συνήθως) όλα τα κείμενα για τα οποία υπάρχει αναφορά σε αυτό. Αυτοματοποιεί επαναλαμβανόμενες εργασίες σε ασσύληπτες ταχύτητες για έναν άνθρωπο. Τα προγράμματα αυτά που ερευνούν συστηματικά το διαδίκτυο για ιστοσελίδες, εξερευνούν όλες τις συνδέσεις από ένα δικτυακό τόπο εκκίνησης, που περιλαμβάνει πολλές συνδέσεις με άλλες ιστοσελίδες. Η ιδέα ήταν ότι εξ ορισμού κάθε ιστοσελίδα πρέπει να συνδέεται με κάποια άλλη. Ερευνώντας ένα μεγάλο αριθμό ιστοσελίδων και ακολουθώντας όλες τις συνδέσεις, ένας χρήστης θα ανακαλύψει νέες ιστοσελίδες που περιλαμβάνουν άλλες συνδέσεις. Με τον τρόπο αυτό το μεγαλύτερο τμήμα του διαδικτύου μπορεί να εξερευνηθεί, επαναλαμβάνοντας την παραπάνω διαδικασία. Χρησιμοποιήθηκε κατά κόρον όλα τα επόμενα χρόνια μέχρι τις μέρες μας, κυρίως από μηχανές αναζήτησης, για την επικαιροποίηση και την κατηγοριοποίηση των ιστοσελίδων του Web. Πάντως η διαδικασία αυτή προκάλεσε πολλές αντιδράσεις, καθώς ορισμένες όχι σωστά προγραμματισμένες αράχνες, προκαλούσαν τεράστια κίνηση στο δίκτυο επειδή επισκέπτονταν πολλές φορές τις ίδιες ιστοσελίδες. Οι περισσότεροι διαχειριστές τις αντιμετώπιζαν εχθρικά, ενώ οι προγραμματιστές δημιουργούσαν όλο και περισσότερες αράχνες.



Εικόνα 1: Η λειτουργία μίας αράχνης δικτύου

Το κόστος για την λειτουργία της αράχνης είναι αρκετά υψηλό, καθώς η εταιρεία που διατηρεί τη μηχανή αναζήτησης θα πρέπει συνεχώς να αυξάνει την υπολογιστική της ισχύ ώστε να μπορεί να καλύπτει την εκρηκτική ανάπτυξη του WWW, καθώς και να αναβαθμίζει τακτικά το εύρος των συνδέσεων της με το διαδίκτυο. Για αυτό το λόγο είναι δυνατόν ορισμένες μηχανές αναζήτησης, εκτός απ' τον περιορισμό στον αριθμό των ιστοσελίδων από κάθε δικτυακό τόπο, να περιορίζουν και το συνολικό αριθμό των ιστοσελίδων στο ευρετήριο τους (π.χ. διαγράφοντας τις πιο παλιές), ή να περιορίζουν τη συχνότητα των επισκέψεων στις ίδιες σελίδες ή τέλος να περιορίζουν την αράχνη σε ορισμένες περιοχές του διαδικτύου, όπου πιστεύουν ότι περιέχουν αξιόπιστες πληροφορίες.

Β.Μηχανισμός ευρετηρίου

Όταν η αράχνη επισκέπτεται μία ιστοσελίδα, την παραδίδει στον μηχανισμό ευρετηρίου, ο οποίος αποθηκεύει το πλήρες κείμενο της ιστοσελίδας στη βάση δεδομένων της μηχανής αναζήτησης, συνήθως σε δομή ανεστραμμένου ευρετηρίου. Το ανεστραμμένο ευρετήριο είναι ταξινομημένο αλφαβητικά, με κάθε καταχώριση του ευρετηρίου να περιλαμβάνει μία λέξη, μία λίστα με ιστοσελίδες και σε ορισμένες περιπτώσεις τις ακριβείς θέσεις της λέξης μέσα στην ιστοσελίδα. Αυτή η δομή θεωρείται ιδανική για τις έρευνες με λέξεις κλειδιά, παρέχοντας γρήγορη πρόσβαση σε ιστοσελίδες που περιλαμβάνουν αυτές τις λέξεις κλειδιά. Με σκοπό τη βελτίωση της αναζήτησης, ορισμένες μηχανές αναζήτησης εξαλείφουν συνηθισμένες λέξεις που ονομάζονται stop words. Επίσης ο μηχανισμός ευρετηρίου εκτελεί και άλλες ενέργειες βελτίωσης της απόδοσης όπως η εξάλειψη των σημείων στίξης, των πολλαπλών διαστημάτων και ορισμένες φορές μετατρέπει όλα τα γράμματα σε πεζά.

Η καταχώριση στο ευρετήριο ολόκληρου του κειμένου των ιστοσελίδων, επιτρέπει σε μία μηχανή αναζήτησης να προσφέρει περισσότερες δυνατότητες από την εύρεση ιστοσελίδων που να εμπεριέχουν τις λέξεις κλειδιά. Αν η θέση κάθε λέξης καταγράφεται μπορούν να χρησιμοποιηθούν τελεστές εγγύτητας (NEAR) για τον περιορισμό του αριθμού αποτελεσμάτων των αναζητήσεων. Επίσης η μηχανή μπορεί να αναζητήσει φράσεις ή ακόμη και μεγαλύτερα κομμάτια κειμένου. Τέλος, αν η μηχανή καταγράφει εκτός του κειμένου της ιστοσελίδας και τον κώδικα HTML, η αναζήτηση μπορεί να περιοριστεί σε ορισμένα χαρακτηριστικά μίας ιστοσελίδας όπως ο τίτλος, η διεύθυνση και άλλα. Όταν η αράχνη ανακαλύψει αλλαγές σε κάποιες ιστοσελίδες, τότε ενημερώνονται και τα αντίγραφα στο ευρετήριο. Βέβαια, το τι ακριβώς αντιγράφει στο ευρετήριο, η αράχνη εξαρτάται από κάθε μηχανή αναζήτησης. Οι περισσότερες αξιόλογες μηχανές διαθέτουν το πλήρες κείμενο των ιστοσελίδων στο ευρετήριο τους, υπάρχουν όμως και κάποιες που ευρετηριάζουν μόνο τον τίτλο μιας ιστοσελίδας και τις πρώτες γραμμές κειμένου.

Γ. Μηχανισμός αναζήτησης

Ο μηχανισμός αναζήτησης είναι χωρίς αμφιβολία το πιο πολύπλοκο τμήμα μίας μηχανής αναζήτησης. Περιλαμβάνει πολλά τμήματα όπως: (α) τη διασύνδεση με το χρήστη (φόρμα αναζήτησης), (β) το μηχανισμό που αξιολογεί το ερώτημα και εντοπίζει τις πιο σχετικές ιστοσελίδες στη βάση δεδομένων της μηχανής και (γ) το μορφοποιητή των αποτελεσμάτων. Η φόρμα αναζήτησης και η μορφοποίηση των αποτελεσμάτων είναι περίπου ίδιες σε όλες τις μηχανές αναζήτησης. Όλες οι

μηχανές διαθέτουν φόρμες απλής και προχωρημένης αναζήτησης και δίνουν στους χρήστες τη δυνατότητα να περιορίσουν την αναζήτηση με διάφορες παραμέτρους. Επίσης η εμφάνιση των αποτελεσμάτων είναι παρόμοια και περιλαμβάνει συνήθως και επιπλέον υπερσυνδέσεις (με γνώμονα το πόσο δημοφιλείς είναι).

Όταν πραγματοποιηθεί μια αναζήτηση και γίνει η συλλογή των αποτελεσμάτων από τη βάση δεδομένων της Μηχανής, τα αποτελέσματα αυτά επιστρέφονται στο χρήστη με τη μορφή μιας λίστας με συνδέσεις στις αντίστοιχες σελίδες. Ο τρόπος με τον οποίο εμφανίζονται τα αποτελέσματα αυτά, αλλά και η ταξινόμησή τους στη λίστα, διαφέρει από Μηχανή σε Μηχανή. Τα αποτελέσματα της αναζήτησης είναι δυνατό :

- να ακολουθούν κάποιον αλγόριθμο ταξινόμησης προκειμένου να εξαχθεί η σειρά με την οποία θα εμφανιστούν στη λίστα
- να τοποθετούνται τυχαία στη λίστα
- να τοποθετούνται ανάλογα με τα χρήματα που πληρώνει ο ιδιοκτήτης του κάθε δικτυακού τόπου στην εταιρία της Μηχανής Αναζήτησης ειδικά για να τοποθετηθεί η ιστοσελίδα του σε καλύτερη σειρά στη λίστα.
- να χρησιμοποιείται συνδυασμός των παραπάνω μεθόδων, ανάλογα με την κάθε περίπτωση.

Μερικοί τύποι σελίδων και συνδέσμων εξαιρούνται, λόγω πολιτικής από τις περισσότερες Μηχανές Αναζήτησης. Άλλες, πάλι, εξαιρούνται επειδή οι «αράχνες» των Μηχανών Αναζήτησης δεν μπορούν να τις προσπελάσουν (αν μία ιστοσελίδα δεν έχει συνδέσμους από άλλες ιστοσελίδες). Αυτές οι σελίδες που εξαιρούνται αναφέρονται ως το «Αόρατο Διαδίκτυο» (Invisible ή Deep Web) – πρόκειται για ότι δεν επιστρέφεται από τις Μηχανές Αναζήτησης ως αποτέλεσμα. Άλλο παράδειγμα είναι πως, μία μηχανή αναζήτησης δεν θα μας δώσει καμιά πληροφορία για τον τηλεφωνικό αριθμό ενός ατόμου (αν αυτός δεν είναι καταγεγραμμένος στο κείμενο μιας ιστοσελίδας), ακόμη και αν μπορούμε να αντλήσουμε αυτή τη πληροφορία από την ιστοσελίδα του παρόχου της τηλεφωνικής σύνδεσης. Το Αόρατο Διαδίκτυο εκτιμάται ότι είναι μεγαλύτερο δυο με τρεις φορές, ή και περισσότερο, από το «ορατό» Διαδίκτυο.

2.6 Είδη μηχανών αναζήτησης

Οι μηχανές αναζήτησης είναι ένα εξαιρετικά ισχυρό μέσο για την προώθηση μιας ιστοσελίδας. Πολλές μελέτες έχουν δείξει ότι ποσοστό μεταξύ του 40% με 80% των χρηστών βρίσκει αυτό το οποίο ψάχνει χρησιμοποιώντας την δυνατότητα μιας μηχανής αναζήτησης του internet. Σύμφωνα με την Search Engine Watch, εκτελούνται καθημερινά 625 εκατομμύρια αναζητήσεις.Το καταπληκτικό με τις μηχανές αναζήτησης είναι ότι φέρνουν στοχευμένη επισκεψιμότητα σε μια ιστοσελίδα. Οι επισκέπτες αυτοί έχουν ήδη κίνητρα για να κάνουν μια συναλλαγή με μια ιστοσελίδα γεγονός που τους έκανε και να την αναζητήσουν. Ανάλογα με τον τρόπο βελτιστοποίησης εμφάνισης μιας ιστοσελίδας στις μηχανές αναζήτησης, είναι και τα αποτελέσματα (κατάταξη) που την εμφανίζουν στο κοινό που την αναζητά.

Α. ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΕ ΑΝΙΧΝΕΥΤΗ (CRAWLER-BASED)

Οι crawler-based μηχανές αναζήτησης χρησιμοποιούν αυτοματοποιημένα προγράμματα λογισμικού για την έρευνα και την κατηγοριοποίηση των ιστοσελίδων. Τα προγράμματα πουχρησιμοποιούνται από τις μηχανές αναζήτησης για να έχουν πρόσβαση στις ιστοσελίδες λέγονται αράχνες (spiders), ανιχνευτές (crawlers), ρομπότ (robots) ή bots.Μια spider μπορεί να βρει μια ιστοσελίδα, να κάνει λήψη αυτής (downloading) και ναναλύσει τις πληροφορίες που περιλαμβάνει. Αυτό είναι μια πάγια διαδικασία. Η ιστοσελίδαεν συνεχεία θα προστεθεί στην βάση δεδομένων της μηχανής αναζήτησης. Κατά συνέπεια, κάθε φορά που ένας χρήστης πραγματοποιεί μια αναζήτηση, η μηχανή αναζήτησης θα ελέγχει τις ιστοσελίδες που βρίσκονται στην βάση δεδομένων της, βάση των keywords (λέξεις-κλειδιά) που χρησιμοποίησε ο χρήστης, έτσι ώστε να παρουσιάσει μια λίστα αποτελεσμάτων.Τα αποτελέσματα από προτεινόμενους συνδέσμους (links) για να επισκεφτεί, είναι υπό μορφή λίστας σε σελίδες με διάταξη κατά την οποία είναι “κοντά” (όπως ορίζεται από το bots), σε αυτό που ο χρήστης αναζήτα στο διαδίκτυο.

Οι crawler-based μηχανές αναζήτησης ψάχνουν συνεχώς στο διαδίκτυο για νέες ιστοσελίδες και ενημερώνουν (updating) τις βάσεις δεδομένων τους με πληροφορίες από νέες ή

τροποποιημένες σελίδες (pages). Χαρακτηριστικό παράδειγμα είναι οι Google και Ask μηχανές αναζήτησης.

Β) ΚΑΤΑΛΟΓΟΙ (DIRECTORIES)

Ένας "κατάλογος" (directory), χρησιμοποιεί συντάκτες (ανθρώπους) που αποφασίζουν σε ποια κατηγορία ανήκει μια ιστοσελίδα. Τοποθετούν τις ιστοσελίδες εντός συγκεκριμένων κατηγοριών στους "καταλόγους" της βάσης δεδομένων. Οι συντάκτες (άνθρωποι) ελέγχουν πλήρως την ιστοσελίδα και την κατατάσσουν, με βάση τις πληροφορίες που βρίσκουν, χρησιμοποιώντας ένα προκαθορισμένο σύνολο κανόνων. Οι μεγάλες μηχανές αναζήτησης τέτοιου τύπου είναι οι Yahoo (www.yahoo.com) και Dmoz (www.dmoz.org). Από τα τέλη του 2002 η Yahoo παρείχε αποτελέσματα αναζήτησης χρησιμοποιώντας τεχνολογία που βασίζονταν σε ανιχνευτή (crawler-based) καθώς και στο δικό της κατάλογο (directory).

Γ) ΥΒΡΙΙΚΕΣ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ (HYBRID SEARCH ENGINES)

Οι hybrid μηχανές αναζήτησης χρησιμοποιούν έναν συνδυασμό από crawler-based και directory αποτελέσματα. Όλο και περισσότερες μηχανές αναζήτησης, στις μέρες μας, κινούνται προς ένα hybrid-based μοντέλο. Οι yahoo και Google μηχανές αναζήτησης ανήκουν στην κατηγορία αυτή.

Δ) ΜΕΤΑ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ

Οι meta μηχανές αναζήτησης παίρνουν τα αποτελέσματα από τα αποτελέσματα όλων των άλλων μηχανών αναζήτησης, και τα συγκεντρώνουν σε έναν μεγάλο κατάλογο. Παραδείγματα meta μηχανών αναζήτησης είναι οι Metacrawler (www.metacrawler.com) και Dogpile (www.dogpile.com).

Ε) ΚΑΘΕΤΕΣ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ (VERTICAL SEARCH ENGINES)

Μια κάθετη (vertical) μηχανή αναζήτησης, σε αντίθεση με μια γενική μηχανή αναζήτησης διαδικτύου (web search engine), εστιάζει σε ένα συγκεκριμένο τμήμα του online περιεχομένου. Η κάθετη περιοχή περιεχομένου μπορεί να βασίζεται στην επικαιρότητα, τον τύπο ή το είδος του περιεχομένου. Οι συνηθισμένες vertical μηχανές αναζήτησης περιλαμβάνουν αγορές (shopping), την αυτοκινητοβιομηχανία, την νομική ενημέρωση, ιατρικές πληροφορίες και τα ταξίδια. Σε αντίθεση με τις γενικές διαδικτυακές μηχανές αναζήτησης, οι οποίες επιχειρούν να καταχωρίσουν στο ευρετήριο τους μεγάλα τμήματα του παγκόσμιου ιστού χρησιμοποιώντας έναν διαδικτυακό ανιχνευτή (web crawler), οι vertical μηχανές αναζήτησης συνήθως χρησιμοποιούν ένα εστιασμένο πρόγραμμα ανίχνευσης που επιχειρεί να τοποθετήσει στο ευρετήριο τους μόνο σελίδες που είναι σχετικές με ένα προκαθορισμένο θέμα ή σύνολο συναφών θεμάτων.

Μερικές vertical ιστοσελίδες αναζήτησης εστιάζουν σε μεμονωμένες vertical, ενώ άλλες ιστοσελίδες περιλαμβάνουν πολλαπλές vertical αναζητήσεις εντός μίας μηχανής αναζήτησης.

Η vertical αναζήτηση προσφέρει πολλά πιθανά οφέλη από ότι οι γενικές μηχανές αναζήτησης όπως:

- Μεγαλύτερη ακρίβεια λόγω περιορισμένης εμβέλειας
- Αξιοποίηση των domain γνώσεων συμπεριλαμβανομένων των ταξινομήσεων και
- οντολογιών.
- Υποστήριξη ειδικών μοναδικών εργασιών / στόχων του χρήστη. (Wikipedia, 2011)

Ζ) ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ ΕΙΙΚΟΤΗΤΑΣ (SPECIALITY SEARCH ENGINES)

Οι μηχανές αναζήτησης ειδικότητας έχουν αναπτυχθεί για να ανταποκριθούν στις ανάγκες

που έχουν εξειδικευμένοι τομείς. Στον πίνακα που ακολουθεί υπάρχουν πολλές μηχανές

αναζήτησης ειδικότητας.

Κεφαλιο 3 : Θεωρητικό υπόβαθρο

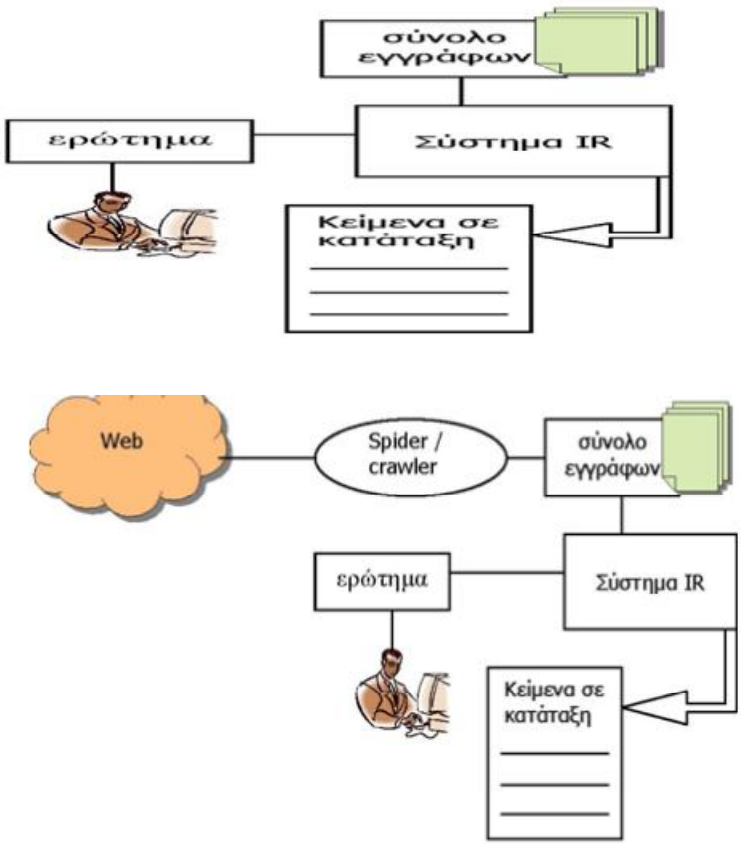
3.1 Ανάκτηση πληροφορίας (Information retrieval)

3.1.1 Ορισμός της Ανάκτησης Πληροφοριών

Η ανάκτηση πληροφοριών μπορεί να οριστεί ως η ανεύρεση, εντός μεγάλων συλλογών πληροφοριών (συνήθως αποθηκευμένων ψηφιακά σε Η/Υ), υλικού με αδόμητη μορφή (συχνά ψηφιακών τεκμηρίων που περιέχουν κείμενο σε φυσική γλώσσα, φωτογραφίες και βίντεο), το οποίο ικανοποιεί κάποια πληροφοριακή αναζήτηση. Ένας ακόμη ορισμός (Tague-Sutcliffe, J.M., 1996) ορίζει την ανάκτηση πληροφοριών ως “μια διαδικασία στην οποία το σύνολο των αρχείων ή των τεκμηρίων αναζητούνται για να βρεθούν στοιχεία που μπορούν να βοηθήσουν στην ικανοποίηση των πληροφοριακών αναγκών ή ενδιαφερόντων ενός ατόμου ή μίας ομάδας”. Κατά τους Harter και Hert (1997), “ανάκτηση πληροφοριών είναι η πρακτική ενέργεια που συντελείται από το χρήστη με στόχο να ικανοποιήσει μία ανθρώπινη ανάγκη συμβουλευόμενος αποθηκευμένες πληροφορίες”.

3.1.2 Αρχιτεκτονική ενός συστήματος IR

Σε ένα τυπικό παράδειγμα συστήματος IR, έχουμε ένα σύνολο εγγράφων κειμένων σε φυσική γλώσσα και ένα ερώτημα (query) το οποίο εκφράζει την πληροφοριακή ανάγκη του χρήστη σε μορφή ακολουθίας όρων. Στόχος του συστήματος είναι να επιστρέψει στο χρήστη ένα σύνολο κειμένων που σχετίζονται με την ερώτηση του, σε κατάταξη (ranked), ανάλογα με το βαθμό συσχέτισης.



3.1.3 Αυτόματη ανάλυση κειμένου

Ένα σημαντικό πρόβλημα στα συστήματα ανάκτησης πληροφοριών είναι η αναπαράσταση του περιεχομένου των εγγράφων. Πρέπει να εφαρμοστεί ανάλυση κειμένου για να εκχωρηθεί σε κάθε έγγραφο ένα σύνολο περιγραφητών (descriptors) ικανών να εκπροσωπήσουν το περιεχόμενό του. Η ανάθεση των descriptors θα πρέπει να πληροί τρεις σκοπούς:

1 Να επιτρέπει την εύρεση των αντικειμένων που ασχολούνται με θέματα που ενδιαφέρουν το χρήστη.

2 Να συσχετίζουν αντικείμενα μεταξύ τους, και έτσι να συσχετίζουν θεματικές περιοχές, με τον εντοπισμό διακριτών αντικειμένων που ασχολούνται με παρόμοιες ή σχετικές θεματικές περιοχές.

3 Να προβλέπουν τη σχετικότητα των επιμέρους στοιχείων πληροφοριών με συγκεκριμένες απαιτήσεις πληροφόρησης, μέσω της χρήσης όρων ευρετηρίου με σαφώς

καθορισμένο νόημα.

Η διαδικασία της εκχώρησης descriptors μπορεί να πραγματοποιείται αυτόματα, χρησιμοποιώντας ένα σύστημα ηλεκτρονικού υπολογιστή, ή χειροκίνητα, χρησιμοποιώντας ειδικούς στο θέμα. Εμείς θα ασχοληθούμε μόνο με την πρώτη, δηλαδή την αυτόματη ευρετηρίαση (automatic indexing). Εκτός από την εκχώρηση descriptors, η ανάλυση του κειμένου θα πρέπει να παρέχει έναν τρόπο μέτρησης της σημασίας κάθε descriptor για σκοπούς αναγνώρισης περιεχομένου, το οποίο είναι γνωστό ως στάθμιση (weighting). Ακολουθεί μια περιγραφή με βάση στατιστικές μεθόδους και των δύο διαδικασιών ανάλυσης κειμένου και ανάκτησης πληροφοριών, της αυτόματης ευρετηρίασης και της στάθμισης.

3.1.4 Αυτόματη ευρετηρίαση (Automatic indexing)

Στο επίκεντρο όλων των μηχανών αναζήτησης είναι η έννοια του ευρετηρίου: η επεξεργασία των αρχικών δεδομένων σε μια ιδιαίτερα αποδοτική δομή δεδομένων που επιτρέπει την ταχεία πρόσβαση σε τυχαίες λέξεις αποθηκευμένες στο εσωτερικό του και διευκολύνει τη γρήγορη αναζήτηση. Η διαδικασία του indexing αρχίζει με την αναγνώριση κάθε ξεχωριστής λέξης από το έγγραφο ως πιθανός περιγραφέας (descriptor). Μετά την αναγνώριση των λέξεων, πρέπει να εξαλειφθούν οι stopwords δηλαδή λέξεις υψηλής συχνότητας που είναι φτωχοί descriptors (π.χ. άρθρα κλπ). Τα stopwords δεν μόνο είναι άχρηστα για την αναγνώριση περιεχομένου, αλλά καταλαμβάνουν και περίπου το 50% του κειμένου του εγγράφου.

Το επόμενο βήμα είναι η κατάργηση των προθεμάτων και επιθεμάτων των λέξεων, έτσι ώστε κάθε λέξη να είναι μειωμένη στη ρίζα της (stem). Η διαδικασία αυτή καλείται stemming και χρησιμοποιείται για τη βελτίωση της αποτελεσματικότητας της ανάκτησης και τη μείωση του μεγέθους του ευρετηρίου.

Το stemming γενικά είτε βελτιώνει την αποτελεσματικότητα της ανάκτησης ή δεν έχει καμία επίδραση, αλλά γενικά οι αντίστοιχοι αλγόριθμοι κάνουν συχνά λάθη. Για παράδειγμα ένας αλγόριθμος μειώνει τις λέξεις «public» και «publication» στη ρίζα «public», αν και οι δύο λέξεις είναι διαφορετικές και πρέπει να διακρίνονται. Η υπόθεση που γίνεται είναι ότι το ποσοστό αυτών των σφαλμάτων δεν έχει πραγματική επίδραση στην απόδοση της ανάκτησης.

Κάθε ρίζα που έχει ανιχνευθεί από ένα έγγραφο μπορεί να είναι ένας από τους descriptors του, που είναι γνωστοί ως όροι ευρετηρίου. Ωστόσο οι όροι ευρετηρίου περιλαμβάνουν και άλλους όρους πέρα από τις ρίζες που έχουν εξαχθεί από τα έγγραφα. Χρησιμοποιούνται λεξικά, έτσι ώστε να περιλαμβάνονται και οι έννοιες που σχετίζονται με τους όρους ευρετηρίου με την ελπίδα της διεύρυνσης της ερμηνείας τους. Ευρεία ερμηνεία μπορεί επίσης να επιτευχθεί με τη χρήση ενός θησαυρού (thesaurus), που παρέχει μια ομαδοποίηση σε κατηγορίες των όρων που χρησιμοποιούνται σε μια δεδομένη θεματική περιοχή. Η διαδικασία ονομάζεται αυτόματη ταξινόμηση των λέξεων – κλειδιών (automatic keyword classification) και μπορεί να αξιοποιηθεί είτε με την αντικατάσταση κάθε όρου του συνόλου των descriptors από το όνομα της κατηγορίας που ανήκει, ή με την αντικατάσταση κάθε όρου του συνόλου των descriptors από όλες τις λέξεις – κλειδιά της κατηγορία που ανήκει.

Αφού παραχθούν όσο το δυνατόν περισσότεροι όροι ευρετηρίου, ακολουθεί η διαδικασία της στάθμισης, προκειμένου να προσδιοριστούν εκείνοι οι όροι που έχουν τη μεγαλύτερη σημασία για την αναγνώριση του περιεχομένου.

3.1.5. Στάθμιση (Weighting)

Η διαδικασία της στάθμισης εκχωρεί ένα βάρος σε κάθε όρο του ευρετηρίου αναλόγως τη σημασία του για την αναγνώριση περιεχομένου. Οι περισσότερες μέθοδοι στάθμισης είναι βασισμένοι στην παρατήρηση ότι η συχνότητα της εμφάνισης μια λέξης σε ένα κείμενο σχετίζεται με τη σημασία της για την αναπαράσταση περιεχομένου. Αν οι ξεχωριστές λέξεις του εγγράφου παρουσιαστούν με φθίνουσα σειρά συχνότητας εμφάνισής τους στο έγγραφο αυτό, τότε συνήθως παρατηρείται ο νόμος σταθερής κατάταξης συχνότητας του Zipf (constant rankfrequency law). Ο νόμος αυτός λέει ότι η συχνότητα μιας λέξης πολλαπλασιασμένης με την σειρά κατάταξής της ισούται με τη συχνότητα μιας άλλης λέξης πολλαπλασιασμένης με την σειρά κατάταξής της. Ο νόμος αυτός εξηγείται από το γεγονός ότι συνήθως οι άνθρωποι προτιμούν να επαναλαμβάνουν λέξεις που έχουν χρησιμοποιήσει ήδη παρά να χρησιμοποιούν νέες.

Οι ιδιότητες που χαρακτηρίζουν μια λέξη ως χρήσιμο όρο ευρετηρίου είναι οι παρακάτω:

- Πρέπει να σχετίζεται με το περιεχόμενο του εγγράφου ώστε να μπορεί να είναι

ανακτήσιμο όταν χρειαστεί, έχοντας έτσι ανακτήσει ένα μεγάλο μέρος των σχετικών εγγράφων. Το ποσοστό των σχετικών εγγράφων που έχουν ανακτηθεί είναι γνωστό ως ανάκληση (recall).

• Πρέπει να ξεχωρίζει το έγγραφο της από το υπόλοιπα, για να αποτρέψει την ανάκτηση και αντικειμένων που δε θέλουμε, έχοντας έτσι σχετικό ένα μεγάλο μέρος των εγγράφων που έχουν ανακτηθεί. Το ποσοστό των ανακτημένων εγγράφων που είναι σχετικά είναι γνωστό ως ακρίβεια (precision).

Όροι με μεγάλη συχνότητα εμφάνισης στο έγγραφο φαίνεται να είναι χρήσιμοι για την πρώτη απαίτηση. Αυτό υποδηλώνει τη χρήση ενός παράγοντα συχνότητας όρου (term frequency – tf) σαν πρώτο κομμάτι του συστήματος στάθμισης. Όροι με χαμηλή συχνότητα εμφάνισης σε όλη τη συλλογή εγγράφων φαίνεται να είναι χρήσιμοι για τη δεύτερη απαίτηση. Αυτό υποδηλώνει τη χρήση ενός παράγοντα αντίστροφης συχνότητας εγγράφου (inverse document frequency – idf) σαν δεύτερο κομμάτι του συστήματος στάθμισης. Χρησιμοποιώντας το γινόμενο του term frequency tfij και του inverse document frequency idfj ενός όρου j ενός εγγράφου i, μπορούμε να αποκτήσουμε ένα καλό μέτρο της σημασίας αυτού του όρου για την αναγνώριση περιεχομένου αυτού του εγγράφου χρησιμοποιώντας το ακόλουθο βάρος wij για τον όρο αυτό:

wij = tfij · idfj

Το term frequency tfij υπολογίζεται ως ο αριθμός των φορών που εμφανίζεται ο όρος j στο έγγραφο i. Το inverse document frequency idfj υπολογίζεται ως:

$$idf_j = \log\left(\frac{N}{f_j}\right)$$

όπου fj είναι ο συνολικός αριθμός εμφανίσεων του όρου j στη συλλογή εγγράφων και N ο αριθμός εγγράφων στη συλλογή. Εκτός από τα tf και idf, είναι χρήσιμος και ένας παράγοντας κανονικοποίησης, ειδικά σε συλλογές εγγράφων με πολύ διαφορετικά μήκη. Τα πιο μεγάλα έγγραφα τείνουν να έχουν μεγαλύτερη πιθανότητα να ανακτηθούν σαν σχετικά, αν και όλα τα

σχετικά έγγραφα πρέπει να θεωρούνται εξίσου σημαντικά ανεξαρτήτως μεγέθους. Το κανονικοποιημένο βάρος tfij · idfj μπορεί να οριστεί ως εξής:

$$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_j \left(w_{ij}^2\right)}}$$

Το μοντέλο στάθμισης tf · idf δεν παρουσιάζει σημαντικές θεωρητικές ιδιότητες, σε αντίθεση με τη πιθανοτική στάθμιση (probabilistic weighting). Σύμφωνα με αυτήν, ένα κατάλληλο βάρος wj για έναν όρο j δίνεται από την ακόλουθη έκφραση:

$$w_j = \frac{r/(R-r)}{(n-r)/[-n-(R-r)]}$$

όπου R είναι ο αριθμός των σχετικών εγγράφων, r ο αριθμός των σχετικών εγγράφων που περιέχουν τον όρο j, N ο αριθμός όλων των εγγράφων στη συλλογή και n ο αριθμός των εγγράφων που περιέχουν τον όρο j. Η παραπάνω έκφραση είναι γνωστή ως βάρος σχετικότητας (relevance weight) και ορίζει τη σημασία ενός όρου χρησιμοποιώντας το ποσοστό των σχετικών εγγράφων στα οποία εμφανίζεται ο όρος διαιρούμενου με το ποσοστό των μησχετικών εγγράφων στα οποία εμφανίζεται ο όρος. Υποθέτοντας ανεξαρτησία των όρων (οι όροι εμφανίζονται ανεξάρτητα από τον κάθε άλλο) και δυαδική μορφή της συχνότητας όρου (δηλαδή 1 αν υπάρχει ο όρος στο έγγραφο και 0 αν δεν υπάρχει), έχει αποδειχτεί ότι η πιθανοτική στάθμιση σχετικότητας καταλήγει να είναι αρκετά παρόμοια με τη στάθμιση tf · idf.

3.1.6. Τεχνικές ανάκτησης

Α) Ανάκτηση διανυσματικού χώρου (Vector space retrieval)

Στην περίπτωση ενός συστήματος ανάκτησης πληροφορίας, ανακτώνται τα έγγραφα που θεωρούνται σχετικά με το ερώτημα ενός χρήστη. Όλες οι στρατηγικές ανάκτησης είναι βασισμένες σε μία σύγκριση μεταξύ του ερωτήματος και των εγγράφων, που αναγνωρίζει τα πιθανά σχετικά έγγραφα για το συγκεκριμένο ερώτημα. Η ανάκτηση διανυσματικού χώρου θεωρεί ένα χώρο εγγράφων αποτελούμενο από έγγραφα. Ο τριδιάστατος χώρος επεκτείνεται σε u διαστάσεις όταν είναι παρόντες u όροι ευρετηρίου. Για κάθε έγγραφο i, uδιάστατα διανύσματα εγγράφου Di κατασκευάζονται από ένα σύνολο από u όρους ευρετηρίου t1, t2, … tu:

Di = (di1, di2, …, diu) όπου dij είναι το βάρος που έχει εκχωρηθεί στον όρο j του εγγράφου i. Παρομοίως, ένα uδιάστατο διάνυσμα Q κατασκευάζεται για κάθε ερώτημα που εισάγει ένας χρήστης:

Q = (q1, q2, …, qu) όπου qj είναι το βάρος που έχει εκχωρηθεί στον όρο j του ερωτήματος Q. Χρησιμοποιώντας τις παραπάνω διανυσματικές αναπαραστάσεις, υπολογίζονται αξίες ομοιότητας για κάθε ζευγάρι εγγράφου ερωτήματος:

$$Sim(D_i, Q) = \frac{\sum_{j-1}^{u} (d_{ij} \cdot q_j)}{\sqrt{\sum_{j-1}^{u} d_{ij}^2 \cdot \sum_{j-1}^{u} q_j^2}}$$

Έγγραφα που έχουν αξία ομοιότητας με το ερώτημα παραπάνω από ένα προκαθορισμένο όριο θεωρούνται σχετικά με το ερώτημα αυτό. Έτσι, το τελικό αποτέλεσμα ενός IR (Information Retrieval) συστήματος βασισμένο σε διανυσματικό χώρο, είναι ένα σύνολο από έγγραφα που συνήθως κατατάσσονται σε φθίνουσα σειρά αξίας ομοιότητας με το ερώτημα του χρήστη.

Β) Πιθανοτική ανάκτηση (Probabilistic retrieval)

Η πιθανοτική ανάκτηση παίρνει υπόψη τις ιδιότητες σχετικότητας των εγγράφων. Σύμφωνα με το μοντέλο ανάκτησης δυαδικής ανεξαρτησίας (binary independence retrieval model), κάθε έγγραφο (ερώτημα) αναπαριστάται από ένα υδιάστατο δυαδικό διάνυσμα x (r) :

x = (x1, x2, …, xu)
r = (r1, r2, …, ru)

όπου xj (rj) υποδηλώνει την απουσία ή παρουσία του j όρου στο έγγραφο (ερώτημα) όταν είναι 0 ή 1 αντίστοιχα.
Ένα έγγραφο είναι σχετικό με ένα συγκεκριμένο ερώτημα αν η πιθανότητα του εγγράφου να είναι σχετικό, δεδομένου του διανύσματος εγγράφου x, είναι μεγαλύτερη από την πιθανότητα να μην είναι σχετικό το έγγραφο:

P(Relevant|x) > P(Nonrelevant|x)

Από τον παραπάνω κανόνα απόφασης προκύπτει η επόμενη συνάρτηση αντιστοίχησης, από την οποία υπολογίζονται αξίες κατάστασης ανάκτησης g (retrieval status values) για κάθε ζευγάρι εγγράφου – ερωτήματος:

$$g(x,r) = \sum_{j-1}^{u} \left(r_j \cdot x_j \cdot \log \frac{p_j(1-j_i)}{(1-p_j)q_j} \right) + C$$

Όπου pj η πιθανότητα να υπάρχει ο όρος ευρετηρίου j αν το έγγραφο είναι σχετικό, qj η πιθανότητα να υπάρχει ο όρος ευρετηρίου j αν το έγγραφο δεν είναι σχετικό. Το C είναι σταθερά για ένα δεδομένο ερώτημα και δεν επηρεάζει την κατάταξη των εγγράφων.
Τα έγγραφα κατατάσσονται με φθίνουσα σειρά των αξιών κατάστασης ανάκτησης.
Ένας τρόπος να εκτιμήσουμε τις πιθανότητες pj και qj είναι κάνοντας μια αρχική αναζήτηση βασισμένη σε άλλες στρατηγικές ανάκτησης και χρησιμοποιώντας τα κορυφαία έγγραφα σαν σχετικά, ή εφαρμόζοντας ανάδραση σχετικότητας των χρηστών (user relevance feedback).
Υποθέτοντας ότι όλα τα pj είναι τα ίδια και ότι τα qj υπολογίζονται ως nj/N, όπου nj είναι ο αριθμός των εγγράφων στα οποία εμφανίζεται ο όρος j και N το μέγεθος της συλλογής, τότε η συνάρτηση πιθανοτικής αντιστοίχησης γίνεται πολύ παρόμοια με τη συνάρτηση

αντιστοίχησης του διανυσματικού χώρου με δυαδική στάθμιση.

Άλλες στρατηγικές ανάκτησης περιλαμβάνουν το μοντέλο λογικής ανάκτησης (boolean retrieval model) και το clusterbased μοντέλο. Στο πρώτο μοντέλο, κάθε έγγραφο σχετίζεται με ένα σύνολο λέξεων – κλειδιών και κάθε ερώτημα έχει τη μορφή μιας λογικής έκφρασης με τελεστές and, or και not. Τα ανακτηθέντα έγγραφα είναι αυτά που περιέχουν όρους ευρετηρίου στο συνδυασμό που ορίζεται από το ερώτημα. Στα μοντέλα, τα έγγραφα ομαδοποιούνται σε συστάδες (clusters). Τα clusters προμηθεύουν ένα ακόμη μηχανισμό για επιπλέον αντιστοιχήσεις μεταξύ όρων ερωτημάτων και συστάδων εγγράφων.

Η ανάδραση σχετικότητας είναι μια γνωστή τεχνική που χρησιμοποιείται σε πολλές στρατηγικές ανάκτησης για να αυξήσουν την αποτελεσματικότητα της ανάκτησης. Μπορεί να επιτευχθεί είτε με την επαναστάθμιση των όρων ερωτημάτων βασισμένη στην κατανομή αυτών των όρων στο σύνολο των σχετικών και μησχετικών εγγράφων που ανακτήθηκαν σε απάντηση του ερωτήματος, είτε αλλάζοντας τους πραγματικούς όρους στο ερώτημα. Οι χρήστες κρίνουν τη σχετικότητα των ανακτηθέντων εγγράφων αφού έχει γίνει μια αρχική αναζήτηση.

3.2 Έμμεση ανατροφοδότηση (Implicit feedback)

3.2.1 Ιδιότητες έμμεσης ανατροφοδότησης

Κάθε φορά που ένας χρήστης διατυπώνει ένα ερώτημα ή κάνει κλικ σε ένα αποτέλεσμα αναζήτησης, παρέχεται ανατροφοδότηση (feedback) στη μηχανή αναζήτησης. Σε αντίθεση με έρευνες ή άλλες μορφές ρητής ανατροφοδότησης, η έμμεση ανατροφοδότηση αυτή είναι ουσιαστικά ελεύθερη, αντικατοπτρίζει τη φυσική χρήση της μηχανή αναζήτησης, και είναι συγκεκριμένη σε ένα χρήστη και μία συλλογή. Μια έξυπνη μηχανή αναζήτησης θα μπορούσε να χρησιμοποιήσει αυτή την έμμεση ανατροφοδότηση για να μάθει λειτουργίες εξατομικευμένης κατάταξης.

Το βασικό κίνητρο για τη χρήση της έμμεσης ανατροφοδότησης είναι ότι καταργεί το κόστος για τον χρήστη της εξέτασης και βαθμολόγησης κάθε στοιχείου ρητά. Ενώ εξακολουθεί να υπάρχει ένα υπολογιστικό κόστος αποθήκευσης και επεξεργασίας των δεδομένων, αυτό μπορεί να είναι αόρατο από το χρήστη. Σε ένα δικτυωμένο περιβάλλον συνήθως είναι

δύσκολο για το χρήστη να διαχωρίσει την καθυστέρηση του δικτύου από την επιπλέον επεξεργασία της εφαρμογής. Αν και υπάρχουν σαφώς όρια στις ανοχές του χρήστη, η αποθήκευση και μεταφορά των δεδομένων έμμεσης ανατροφοδότησης δεν είναι υπολογιστικά έντονο έργο. Επίσης σε συστήματα ρητής βαθμολόγησης των στοιχείων, αν ο χρήστης δεν αντιλαμβάνεται κάποιο άμεσο όφελος, τότε μπορεί να συνεχίσει να χρησιμοποιεί το σύστημα χωρίς όμως να κάνει βαθμολόγηση. Έτσι το σύστημα αυτό θα μπορούσε να οδηγηθεί σε έλλειψη οποιασδήποτε διαβάθμισης.

Το πλεονέκτημα της έμμεσης ανατροφοδότησης σε σχέση με τη ρητή είναι ότι μπορούν να συλλέγονται δεδομένα σε πολύ χαμηλότερο κόστος, σε πολύ μεγαλύτερες ποσότητες, και χωρίς επιβάρυνση για το χρήστη του συστήματος ανάκτησης τους. Ωστόσο, η έμμεση ανατροφοδότηση είναι πιο δύσκολο να ερμηνευθεί και δυνητικά θορυβώδης.
Οι συμπεριφορές των χρηστών που δηλώνουν ένα έμμεσο ενδιαφέρον για την ιστοσελίδα μπορούν να κατηγοριοποιηθούν ως εξής :

- Δείκτες ενδιαφέροντος σήμανσης. Διάφορες ενέργειες του χρήστη μπορούν να θεωρηθούν σαν μια μορφή σήμανσης, που μπορεί να ερμηνευτεί σαν δείκτης ενδιαφέροντος. Μερικές από αυτές είναι η αποθήκευση σε αρχείο ή προσθήκη στα αγαπημένα, η εκτύπωση, ή η προώθηση του εγγράφου μέσω ηλεκτρονικού ταχυδρομείου.
- Δείκτες ενδιαφέροντος χειρισμού. Μερικές ενέργειες, όπως η αντιγραφή και η επικόλληση, μπορούν να θεωρηθούν σαν δείκτες ενδιαφέροντος. Άλλες περιλαμβάνουν το άνοιγμα ενός νέου παραθύρου του περιηγητή (π.χ. πιθανώς ο χρήστης να θέλει να κρατήσει ανοιχτό το υπάρχον παράθυρο γιατί η σελίδα είναι ενδιαφέρουσα), η αναζήτηση στη σελίδα για κάποιο κείμενο, ή η κύλιση (scrolling) μέσα στο έγγραφο
- Δείκτες ενδιαφέροντος πλοήγησης. Αν ο χρήστης ξοδεύει χρόνο με τη σελίδα ανοιχτή, ακολουθεί ή δεν ακολουθεί ένα σύνδεσμο, τότε μπορούμε να θεωρήσουμε αυτές τις ενέργειες σαν δείκτες πλοήγησης.
- Δείκτες ενδιαφέροντος επανάληψης. Είναι λογικό να υποθέσουμε ότι όταν κάνουμε κάτι σε μεγάλες ποσότητες, αυτό φανερώνει μεγαλύτερο ενδιαφέρον. Έτσι όταν ο χρήστης επισκέπτεται πολλές φορές την ίδια σελίδα ή ξοδεύει περισσότερο χρόνο σε μια σελίδα, μπορούμε να συμπεράνουμε ότι είναι ενδιαφέρουσα για αυτόν.

Κάποιες από αυτές τις πηγές δεδομένων έχουν και πρόσθετη πληροφορία (π.χ. σε μια ενέργεια αγοράς ενός αντικειμένου αντιστοιχεί και μια τιμή, εκτός από την πληροφορία ότι ο

χρήστης ενδιαφέρεται για το αντικείμενο αυτό). Είναι λογικό να υποθέσουμε περισσότερα από την αγορά του αντικειμένου, παρά από μια απλή επιθεώρησή του. Καθώς το διαδίκτυο γίνεται ένα ολοένα και πιο εμπορικό περιβάλλον, αυξάνονται και οι πληροφορίες αυτού του τύπου. Οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για να προτείνουν στο χρήστη άλλα προϊόντα που πιθανώς να του αρέσουν. Στην εργασία αυτή θα χρησιμοποιήσουμε δεδομένα feedback μόνο από τις ακολουθίες των κλικ που κάνουν οι χρήστες της μηχανής αναζήτησης.

3.2.2 Clickstream δεδομένα

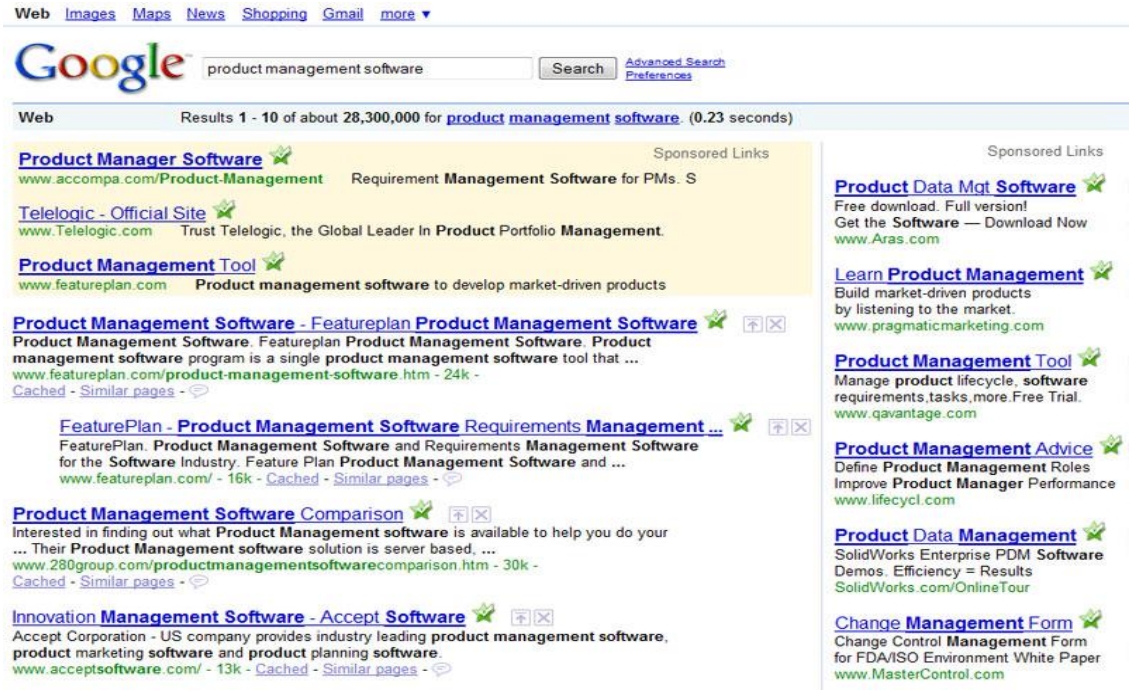
Με τον όρο clickstream εννοούμε το εικονικό μονοπάτι που αφήνει ένας χρήστης καθώς περιηγείται στο διαδίκτυο. Το εικονικό αυτό μονοπάτι, που το συνθέτουν όλα τα κλικ που κάνει ο χρήστης, καταγράφει όλη τη δραστηριότητά του στο διαδίκτυο, συμπεριλαμβανομένων των ιστοσελίδων που επισκέπτεται, για πόσο χρονικό διάστημα ήταν σε κάθε σελίδα καθώς και με ποια σειρά τις επισκέφθηκε. Τα clickstream δεδομένα είναι πολύ χρήσιμα για την ανάλυση της δραστηριότητας στο διαδίκτυο, τη δοκιμή λογισμικού, την έρευνα αγοράς, καθώς και για την ανάλυση της παραγωγικότητας των εργαζομένων.

Η μη εγκεκριμένη από το χρήστη συλλογή των clickstream δεδομένων μπορεί να εγείρει ανησυχίες για την ιδιωτική ζωή, ιδίως εφόσον ορισμένοι πάροχοι υπηρεσιών Internet, προκειμένου να ενισχύσουν τα έσοδά τους, πωλούν τα clickstream δεδομένα των χρηστών τους. Αν και η πρακτική αυτή δεν μπορεί να εντοπίσει μεμονωμένους χρήστες άμεσα, είναι όμως δυνατόν να εντοπιστούν έμμεσα συγκεκριμένοι χρήστες, με βάση το ιστορικό πλοήγησής τους. Οι περισσότεροι χρήστες δε γνωρίζουν την πρακτική αυτή, καθώς και την πιθανότητα έκθεσης της ιδιωτικής τους ζωής.
Άλλοι οργανισμοί χρησιμοποιούν τη συλλογή δεδομένων κλικ με την άδεια του χρήστη για έρευνες, ή για να επιτρέψουν στο χρήστη να επιστρέψει εύκολα σε μια σελίδα που έχει ήδη επισκεφθεί. Παρόμοια, και στη δική μας περίπτωση, ο χρήστης εν γνώσει του χρησιμοποιεί τη μηχανή αναζήτησης, προκειμένου να λάβει προσωποποιημένα αποτελέσματα βασισμένα στο ιστορικό χρήσης του.

Τα clickstream δεδομένα στης μηχανές αναζήτησης μπορούν να θεωρηθούν ως τριπλέτες (q, r, c), όπου q το ερώτημα, r η κατάταξη που παρουσιάστηκε στο χρήστη και c το σύνολο των

συνδέσμων στους οποίους έκανε κλικ ο χρήστης.

Παρακάτω φαίνεται ένα παράδειγμα που ο χρήστης έκανε το ερώτημα “support vector machine”, πήρε την κατάταξη που φαίνεται, και έκανε κλικ στα αποτελέσματα 1, 3 και 5. Όλα αυτά τα δεδομένα καταγράφονται στο αρχείο ιστορικού για περαιτέρω επεξεργασία. Έτσι, βλέπουμε ότι η ποσότητα των δεδομένων που είναι διαθέσιμη είναι σχεδόν απεριόριστη, αφού με τόσο εύκολο τρόπο παίρνουμε τόσο πολλή πληροφορία.



Εικόνα 2 : Κατάταξη για το ερώτημα Product management software

Τα clickstream δεδομένα μπορούν να αποθηκευτούν με μικρό επιπλέον κόστος και χωρίς να διακινδυνεύουν τη λειτουργικότητα και χρησιμότητα της μηχανής αναζήτησης. Συγκεκριμένα, σε σχέση με μεθόδους ρητής ανάδρασης (explicit feedback), δεν προσθέτει καθόλου επιπλέον κόστος στο χρήστη. Το ερώτημα και η κατάταξη που επιστρέφεται μπορεί εύκολα να αποθηκευτεί όταν παρουσιάζεται στο χρήστη. Για την καταγραφή των κλικ, ένας απλός διακομιστής μεσολάβησης (proxy server) μπορεί να κρατάει ένα αρχείο ιστορικού (log file).

Σε κάθε ερώτημα εκχωρείται ένα μοναδικό αναγνωριστικό ID, το οποίο αποθηκεύεται στο log αρχείο μαζί με τις λέξεις – κλειδιά του ερωτήματος και την κατάταξη που επιστρέφεται στο χρήστη. Οι σύνδεσμοι στην σελίδα των αποτελεσμάτων δεν οδηγούν απευθείας στο έγγραφο, αλλά δείχνουν σε έναν proxy server. Οι σύνδεσμοι αυτοί ενσωματώνουν το

αναγνωριστικό ID του ερωτήματος και την URL διεύθυνση του εγγράφου. Όταν ο χρήστης κάνει κλικ στο σύνδεσμο, ο proxy server καταγράφει το URL και το αναγνωριστικό ID του ερωτήματος στο αρχείο ιστορικού. Κατόπιν ο proxy server προωθεί τον χρήστη στη διεύθυνση URL. Αυτή η διαδικασία μπορεί να γίνει διαφανής για το χρήστη και να μην επηρεάσει την επίδοση του συστήματος.

3.2.3 Σχετικές προτιμήσεις που εξάγονται από τα clickstream δεδομένα

Όπως εξηγήσαμε και παραπάνω , είναι δύσκολο να ερμηνεύσουμε τα κλικ των χρηστών σε απόλυτη κλίμακα. Έτσι θα προσπαθήσουμε να εξάγουμε έμμεσα κάποιες προτιμήσεις από ζευγάρια αποτελεσμάτων. Η στρατηγική που θα χρησιμοποιήσουμε είναι βασισμένη στην ιδέα ότι δεν πρέπει να χρησιμοποιούνται σαν ανάδραση (feedback) μόνο τα κλικ του χρήστη, αλλά και το γεγονός ότι σε κάποιους συνδέσμους δεν έκανε κλικ.

Ας πάρουμε την υποθετική κατάταξη των αποτελεσμάτων l1 ως l7 και ας υποθέσουμε ότι ο χρήστης έκανε κλικ στους συνδέσμους l1, l3, και l5

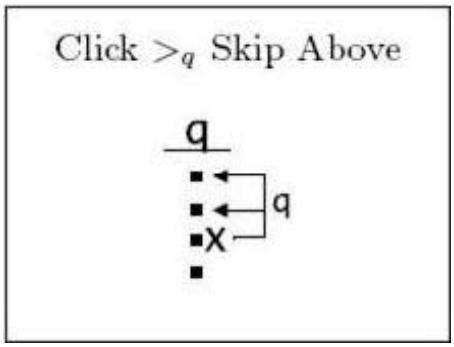
l1 l2 l3 l4 l5 l6 l7

Ενώ είναι δύσκολο να υποθέσουμε ότι τα αποτελέσματα l1, l3, και l5 είναι σχετικά σε απόλυτη κλίμακα, είναι πολύ πιο λογικό να υποθέσουμε ότι το αποτέλεσμα l3 είναι πιο σχετικό από το l2. Όπως είπαμε οι χρήστες σαρώνουν τα αποτελέσματα από πάνω προς τα κάτω, και άρα στο παράδειγμά μας, ο χρήστης, πριν κάνει κλικ στο l3, είδε το l2 και πήρε την απόφαση να μην πατήσει πάνω του. Αυτό δείχνει την προτίμηση του για το l3 σε σχέση με το l2. Παρόμοια μπορούμε να εξάγουμε ότι το l5 είναι πιο σχετικό από τα l2 και l4.

Σημειώνοντας με rel() την αξιολόγηση σχετικότητας του χρήστη αυτού, παίρνουμε την παρακάτω πληροφορία rel(l3) > rel(l2), rel(l5) > rel(l2), rel(l5) > rel(l4) Οι σχετικές προτιμήσεις που εξάγουμε από κάθε ζευγάρι αποτελεσμάτων αντιτίθεται στη μεροληψία παρουσίασης των αποτελεσμάτων, όπου το υψηλότερο στην κατάταξη αποτέλεσμα φαίνεται σαν πιο σχετικό.

Η στρατηγική που δείξαμε στο παραπάνω παράδειγμα και που θα χρησιμοποιήσουμε για την εξαγωγή σχετικών προτιμήσεων του χρήστη περιγράφεται παρακάτω :

Στρατηγική 1 – “Click > Skip above” Για μία κατάταξη (l1, l2, l3, …) και ένα σύνολο C που περιλαμβάνει τις θέσεις κατάταξης των αποτελεσμάτων στα οποία έγινε κλικ, εξάγεται η προτίμηση rel(li) > rel(lj) για όλα τα ζευγάρια 1≤j<1,με i E C και j EC.



Εικόνα 3 Στρατηγική "Click > Skip above

Δεδομένου ενός ερωτήματος q, οι τελείες αναπαριστούν τα αποτελέσματα και τα X εκείνα που έγιναν κλικ.

Δηλαδή δεδομένου ενός αποτελέσματος στο οποίο έγινε κλικ, κάθε αποτέλεσμα υψηλότερης κατάταξης στο οποίο δεν έγινε κλικ έχει μικρότερη σημασία για το χρήστη. Αυτό το συμπέρασμα προέρχεται από το ότι οι χρήστες βλέπουν τα αποτελέσματα σε σειρά, και ένας χρήστης είναι απίθανο να κάνει κλικ σε ένα έγγραφο που θεωρεί λιγότερο σημαντικό από ένα άλλο πιο σχετικό έγγραφο που έχει δει.

3.3 Προγραμματιστικά εργαλεία

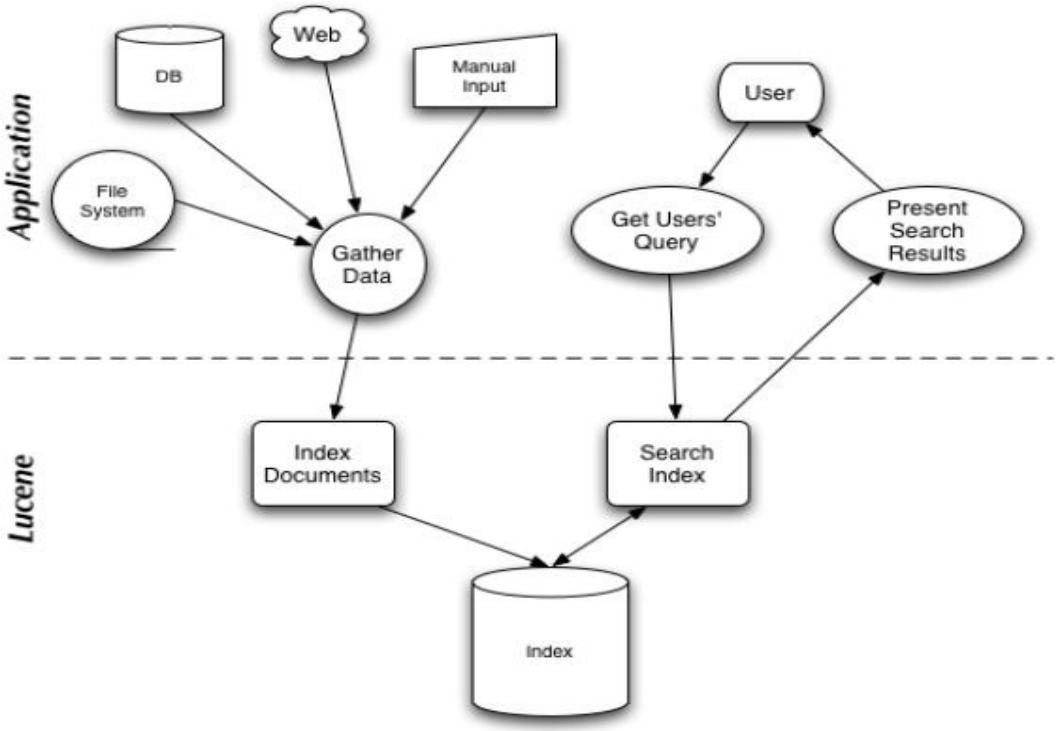
3.3.1 Osmot

Το Osmot είναι μια ανοιχτού κώδικα μηχανή αναζήτησης για την εκμάθηση λειτουργιών ανάκτησης ταξινομημένων αποτελεσμάτων και την αξιολόγηση των κατατάξεων αυτών. Η μηχανή αναζήτησης Osmot υλοποιεί την καταγραφή σε αρχείο, την ανάλυση του αρχείου, την εκμάθηση, την αλλαγή κατάταξης και την αξιολόγηση της λειτουργικότητας. Το Osmot έχει σχεδιαστεί για να μπορεί να χρησιμοποιήσει οποιαδήποτε συνάρτηση κατάταξης και μηχανή αναζήτησης και μπορεί στη συνέχεια να μάθει να βελτιώνει την κατάταξη των αποτελεσμάτων της μηχανής αναζήτησης.

3.3.2. Lucene

Το Lucene είναι μία υψηλής απόδοσης βιβλιοθήκη Ανάκτησης Πληροφοριών (Information Retrieval). Αυτό επιτρέπει την προσθήκη δυνατοτήτων ευρετηρίου και αναζήτησης στις εφαρμογές στις οποίες χρησιμοποιείται. Το Lucene είναι λογισμικό ανοιχτού κώδικα που αρχικά υλοποιήθηκε σε Java και υποστηρίζεται από την Apache Software Foundation.

Το Lucene μπορεί να φτιάξει ευρετήριο και να δώσει τη δυνατότητα αναζήτησης σε οτιδήποτε δεδομένα που μπορούν να μετατραπούν σε μορφή κειμένου. Το Lucene δεν ενδιαφέρεται για την πηγή των δεδομένων, τη μορφή, ή ακόμη και τη γλώσσα, αρκεί να είναι κείμενο. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για δημιουργία ευρετηρίου και αναζήτηση σε δεδομένα που αποθηκεύονται σε αρχεία: ιστοσελίδες σε απομακρυσμένους δικτυακούς διακομιστές, έγγραφα που είναι αποθηκευμένα στο τοπικό σύστημα, δεδομένα αποθηκευμένα σε βάσεις δεδομένων, αρχεία απλού κειμένου, έγγραφα του Microsoft Word, HTML ή PDF αρχεία, ή σε οποιαδήποτε άλλη μορφή από την οποία μπορεί να εξαχθεί κείμενο.



Εικόνα 4 : Διαδικασία indexing και αναζήτησης με το Lucen

3.3.3 Apache Tomcat

Ο Apache Tomcat υλοποιεί τις προδιαγραφές για Java Servlet και JavaServer Pages (JSP) και παρέχει ένα περιβάλλον HTTP web server για να μπορεί να τρέχει κώδικας σε γλώσσα Java. Τα Servlets είναι αντικείμενα της γλώσσας προγραμματισμού Java που επεξεργάζονται με δυναμικό τρόπο αιτήματα (requests) και κατασκευάζουν απαντήσεις (responses). Αυτό επιτρέπει να έχουμε δυναμικό περιεχόμενο στο server χρησιμοποιώντας την πλατφόρμα της Java. Το περιεχόμενο που κατασκευάζεται και επιστρέφεται συνήθως είναι HTML. Οι JavaServer Pages επίσης δίνουν τη δυνατότητα για δημιουργία δυναμικών ιστοσελίδων, ενσωματώνοντας κώδικα Java μαζί με την HTML. Οι JSP σελίδες μεταγλωττίζονται σε Java Servlets όταν καλούνται για πρώτη φορά.

3.3.4 SVM

Το SVM είναι μια υλοποίηση των Support Vector Machines (SVM) στη γλώσσα C. Τα κύρια χαρακτηριστικά του προγράμματος που μας ενδιαφέρουν είναι ότι μπορεί να επιλύσει

προβλήματα κατάταξης και ταξινόμησης, έχει γρήγορο αλγόριθμο βελτιστοποίησης και κάνει εκτιμήσεις του ποσοστού σφάλματος, της ακρίβειας, και της ανάκλησης.
Για να εκπαιδεύσουμε ένα SVM μοντέλο, πρέπει να τροφοδοτήσουμε τον αλγόριθμο SVMμε ένα αρχείο που περιλαμβάνει τον βαθμό της προτίμησης του χρήστη για τα αποτελέσματα κάθε ερωτήματος, καθώς και κάποια χαρακτηριστικά αυτών των αποτελεσμάτων. Η ακρίβεια του μοντέλου μεγαλώνει αναλόγως με την ποσότητα των δεδομένων εκπαίδευσης καθώς και με τα χαρακτηριστικά που έχουν υλοποιηθεί.
Έχοντας πλέον δημιουργήσει ένα μοντέλο από τις προτιμήσεις του χρήστη, μπορούμε να εισάγουμε στον αλγόριθμο SVMτα αποτελέσματα μιας αναζήτησης του χρήστη και εκείνο θα τα ανακατατάξει με βάση το μοντέλο αυτό.

Κεφαλαιο 4 : Δραστηριότητα χρήστη και καταγραφή

4.1 Καταγραφή δραστηριότητας αναζήτησης στα log αρχεία

Η δημιουργία του μοντέλου, που θα χρησιμοποιήσουμε για την ανακατάταξη των αποτελεσμάτων που δίνει το Google, προϋποθέτει την εκπαίδευση του με κάποια δεδομένα εκπαίδευσης. Ακολουθεί η περιγραφή της διαδικασίας για τη συλλογή αυτών των δεδομένων.

Τα δεδομένα που χρειαζόμαστε για την εκπαίδευση θα προκύψουν από την καταγραφή της δραστηριότητας του χρήστη όταν κάνει αναζητήσεις χρησιμοποιώντας την μηχανή αναζήτησής μας. Η καταγραφή αυτή περιλαμβάνει τα ερωτήματα που κάνει, τα αποτελέσματα που του επιστρέφει η μηχανή αναζήτησης και τα χαρακτηριστικά τους, καθώς και τα κλικ που κάνει στα αποτελέσματα αυτά. Η καταγραφή της δραστηριότητας γίνεται στο παρασκήνιο και ουσιαστικά είναι αόρατη στο χρήστη, ο οποίος μπορεί απλά να προσέξει μια μικρή καθυστέρηση, η οποία όμως κυμαίνεται σε λογικά πλαίσια και θα μπορούσε να οφείλεται και σε καθυστερήσεις του δικτύου.

4.1.1. Web Interface της εφαρμογής

Για να αρχίσει η διαδικασία της καταγραφής της δραστηριότητας του χρήστη, πρέπει ο χρήστης να χρησιμοποιήσει το web interface της εφαρμογής μας. Η εφαρμογή τρέχει σε κάποιον υπολογιστή που έχει το ρόλο του server, και χρειάζεται τον Apache Tomcat για να λειτουργήσει.

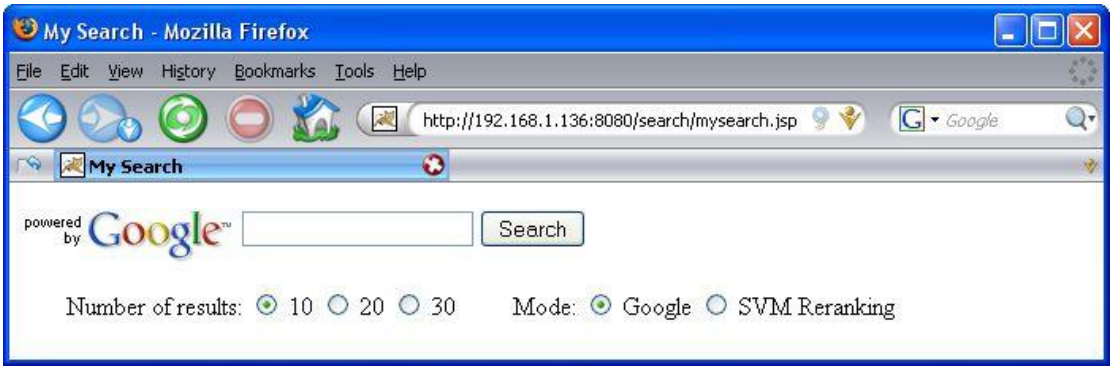
Η διεύθυνση της εφαρμογής έχει την εξής μορφή:

http://<Server IP>:<Tomcat Port>/search/mysearch.jsp

όπου <Server IP> η IP διεύθυνση του υπολογιστή στον οποίο τρέχει ο Tomcat και έχει εγκατεστημένη την εφαρμογή, και <Tomcat Port> ο αριθμός της θύρας στην οποία "ακούει" ο Tomcat.

Βλέπουμε ότι είναι μια JSP σελίδα, γεγονός το οποίο μας επιτρέπει να έχουμε δυναμικό περιεχόμενο με χρησιμοποίηση κώδικα Java. Την πρώτη φορά που κάποιος καλεί τη σελίδα από το server, υπάρχει μια μικρή επιπλέον καθυστέρηση μέχρι να γίνει η μεταγλώττισή της σε Java Servlet. Όλες οι επόμενες φορές όμως θα βρουν τη σελίδα ήδη μεταγλωττισμένη και άρα θα είναι γρήγορη η φόρτωσή της.

Αφού πληκτρολογήσει ο χρήστης τη διεύθυνση της εφαρμογής, θα δει το περιβάλλον αναζήτησης που φαίνεται στην παρακάτω εικόνα.



Εικόνα 5 : Αρχική εικόνα του Web Interface της εφαρμογής

Όπως βλέπουμε, πρόκειται για ένα απλό και λιτό περιβάλλον, σε αντιστοιχία με τη φιλοσοφία του Google. Οι επιλογές που παρέχονται στο χρήστη είναι μια περιοχή κειμένου όπου

πληκτρολογεί το ερώτημα του, πόσα αποτελέσματα θέλει να του επιστραφούν (10, 20 ή 30), και με τι τρόπο κατάταξης – απλή κατάταξη του Google ή ανακατάταξη των αποτελεσμάτων σύμφωνα με το SVM μοντέλο.

4.1.2 Περιεχόμενο αρχείων καταγραφής

Τα βασικά δεδομένα καταγραφής της δραστηριότητας του χρήστη αποθηκεύονται σε δύο αρχεία καταγραφής. Στο πρώτο αρχείο (out.log), που θα χρησιμοποιηθεί για την εξαγωγή των σχετικών προτιμήσεων του χρήστη, αποθηκεύονται:

Για κάθε ερώτημα:

- Η ημερομηνία και ώρα που έγινε το ερώτημα.
- Ένα μοναδικό αναγνωριστικό του ερωτήματος (query id).
- Οι λέξεις – κλειδιά του ερωτήματος.
- Η IP διεύθυνση του χρήστη που έκανε το ερώτημα.
- Οι URL διευθύνσεις των αποτελεσμάτων που παρουσιάστηκαν στο χρήστη.

Για κάθε κλικ σε κάποιο αποτέλεσμα:

- Η ημερομηνία και ώρα που έγινε το κλικ.
- Το μοναδικό αναγνωριστικό του ερωτήματος (query id).
- Η IP διεύθυνση του χρήστη που έκανε το κλικ..
- Η URL διεύθυνση του αποτελέσματος.

Στο δεύτερο αρχείο (out2.log), με του οποίου τα περιεχόμενα θα δημιουργηθεί το ευρετήριο, καταγράφονται τα εξής:

- τα ερωτήματα του χρήστη,
- οι τίτλοι των αποτελεσμάτων
- οι περιλήψεις των αποτελεσμάτων,
- οι URL διευθύνσεις των αποτελεσμάτων.

4.1.3 Αλγόριθμος αναζήτησης

Όταν ο χρήστης πληκτρολογήσει το ερώτημα του προς αναζήτηση, εκτελούνται οι παρακάτω ενέργειες:

1. Αναζήτηση στο Google για τις λέξεις – κλειδιά που εισήγαγε ο χρήστης και με ζητούμενο αριθμό αποτελεσμάτων τον επιλεγμένο
2. Το Google επιστρέφει ένα string το οποίο περιλαμβάνει ουσιαστικά όλο τον HTML κώδικα που θα εμφάνιζε αν κάποιος είχε κάνει το ίδιο ερώτημα από την κανονική ιστοσελίδα της Google. Μέσα από το string αυτό πρέπει να βρούμε και να απομονώσουμε τους τίτλους, τις περιλήψεις και τις URL διευθύνσεις των αποτελεσμάτων.
3. Άνοιγμα των δύο log αρχείων που θα χρησιμοποιήσουμε.
4. Εγγραφή στο πρώτο log αρχείο (out.log) των παρακάτω στοιχείων:

    - Ημερομηνία και ώρα
    - Λέξεις – κλειδιά του ερωτήματος
    - Αναγνωριστικός αύξων αριθμός του ερωτήματος (query id)
    - Διεύθυνση IP του χρήστη που πραγματοποίησε το ερώτημα

5. Εκτέλεση του παρακάτω επαναληπτικού βρόχου

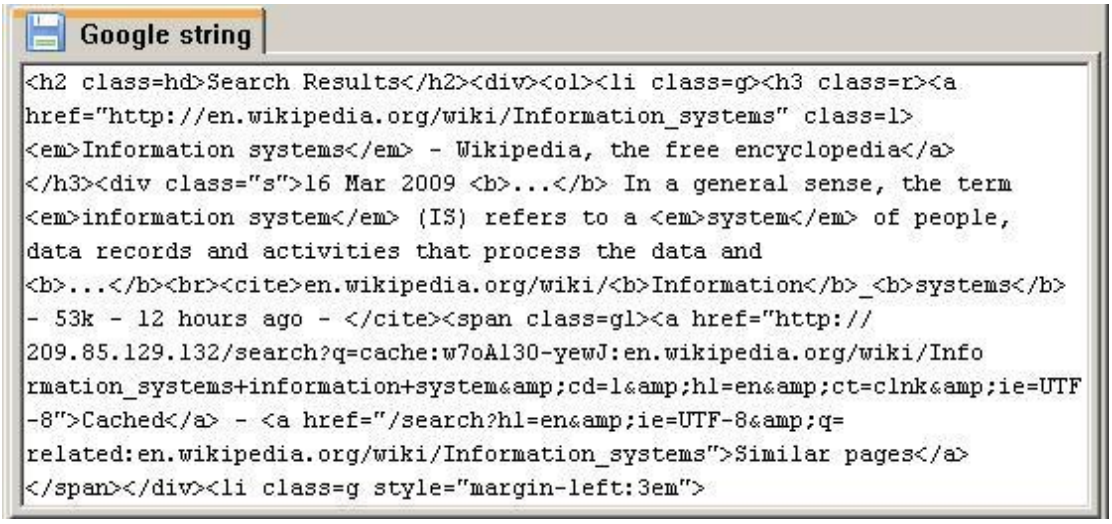
    - Εύρεση και αποθήκευση σε έναν πίνακα της πρώτης URL διεύθυνσης που βρίσκεται μέσα στο string που επέστρεψε το Google.
    - Αν δεν είναι κανονικό αποτέλεσμα κειμένου, αλλά αποτέλεσμα που παραπέμπει σε άλλες σελίδες αναζήτησης του Google, όπως είναι οι σελίδες του Google για εικόνες, βίντεο, βιβλία ή ειδήσεις, παραβλέπουμε αυτό το αποτέλεσμα και συνεχίζουμε στην επόμενη επανάληψη του βήματος 5.
    - Εύρεση στο string και αποθήκευση σε έναν πίνακα του τίτλου του αποτελέσματος.
    - Εύρεση στο string και αποθήκευση σε έναν πίνακα της περίληψης (abstract) του αποτελέσματος.
    - Αφαίρεση των HTML tags από τη διεύθυνση, τον τίτλο και την περίληψη, ώστε να μείνει μόνο καθαρό κείμενο, για να μπορεί να καταχωρηθεί σωστά στο ευρετήριο.

    Εγγραφή στο δεύτερο log αρχείο (out2.log) των παρακάτω στοιχείων:

- Ø Ερώτημα χρήστη
- Ø Τίτλος αποτελέσματος
- Ø Περίληψη αποτελέσματος
- Ø URL διεύθυνση αποτελέσματος

- Αφαίρεση από το αρχικό string των στοιχείων του αποτελέσματος που βρήκαμε, ώστε η επόμενη επανάληψη να συνεχίσει στα επόμενα αποτελέσματα.

6. Οι επαναλήψεις τελειώνουν όταν έχουν απομονωθεί τα στοιχεία όλων των αποτελεσμάτων που επέστρεψε το Google.
7. Εγγραφή στο πρώτο log αρχείο (out.log) των URL διευθύνσεων όλων των αποτελεσμάτων χωρισμένων μεταξύ τους με έναν αστερίσκο *.


Google string

```
<h2 class=hd>Search Results</h2><div><ol><li class=g><h3 class=r><a
href="http://en.wikipedia.org/wiki/Information_systems" class=l>
<em>Information systems</em> - Wikipedia, the free encyclopedia</a>
</h3><div class="s">16 Mar 2009 <b>...</b> In a general sense, the term
<em>information system</em> (IS) refers to a <em>system</em> of people,
data records and activities that process the data and
<b>...</b><br><cite>en.wikipedia.org/wiki/<b>Information</b>_<b>systems</b>
- 53k - 12 hours ago - </cite><span class=gl><a href="http://
209.85.129.132/search?q=cache:w7oAl30-yewJ:en.wikipedia.org/wiki/Info
rmation_systems+information+system&amp;cd=1&amp;hl=en&amp;ct=clnk&amp;ie=UTF
-8">Cached</a> - <a href="/search?hl=en&amp;ie=UTF-8&amp;q=
related:en.wikipedia.org/wiki/Information_systems">Similar pages</a>
</span></div><li class=g style="margin-left:3em">
```

8. Εμφάνιση των αποτελεσμάτων στο χρήστη.

4.1.4 Παράδειγμα εκτέλεσης αναζήτησης

Θα δείξουμε ένα παράδειγμα εκτέλεσης του παραπάνω αλγορίθμου βήμα προς βήμα.

1. Πληκτρολογούμε στη μηχανή αναζήτησης το ερώτημα “information system” και πατάμε το κουμπί Search, ζητώντας 10 αποτελέσματα από το Google.
2. Το Google επιστρέφει στην εφαρμογή ένα string με τα αποτελέσματα, όπως θα τα εμφάνιζε στη σελίδα του. Ακολουθεί ένα μικρό κομμάτι του string αυτού με την πληροφορία για το πρώτο αποτέλεσμα:

Εικόνα 9: Απόσπασμα από το string που επιστρέφει το Google για ένα ερώτημα.

2. Άνοιγμα των αρχείων καταγραφής.
3. Εγγραφή στο πρώτο log αρχείο

4.

- Εύρεση του URL του πρώτου αποτελέσματος

  http://en.wikipedia.org/wiki/Information_systems

- Εύρεση του τίτλου του πρώτου αποτελέσματος

  Information systems Wikipedia, the free encyclopedia

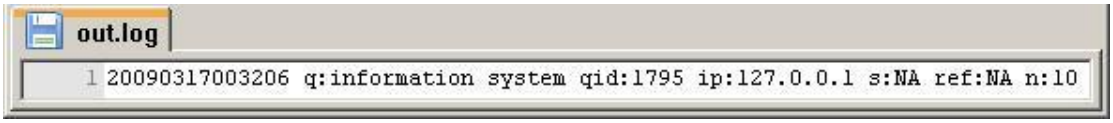
- Εύρεση της περίληψης του πρώτου αποτελέσματος

16 Mar 2009 ... In a general sense, the term information system (IS) refers to a system of people, data records and activities that process the data and ...

- Αφαίρεση των HTML tags από τον τίτλο και την περίληψη, που γίνονται:

Τίτλος: Information systems Wikipedia, the free encyclopedia

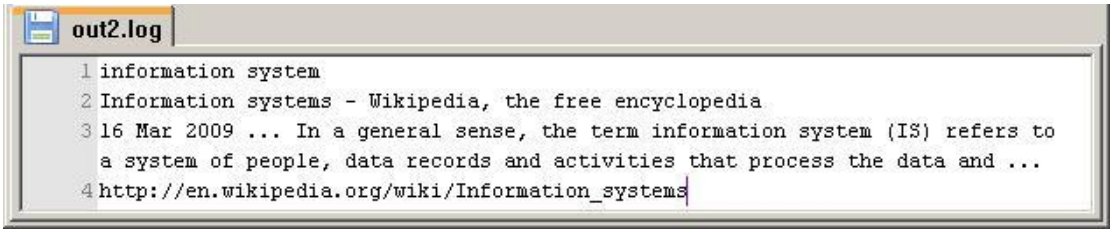
Περίληψη: 16 Mar 2009 ... In a general sense, the term information system (IS) refers to a system of people, data records and activities that process the data and…..



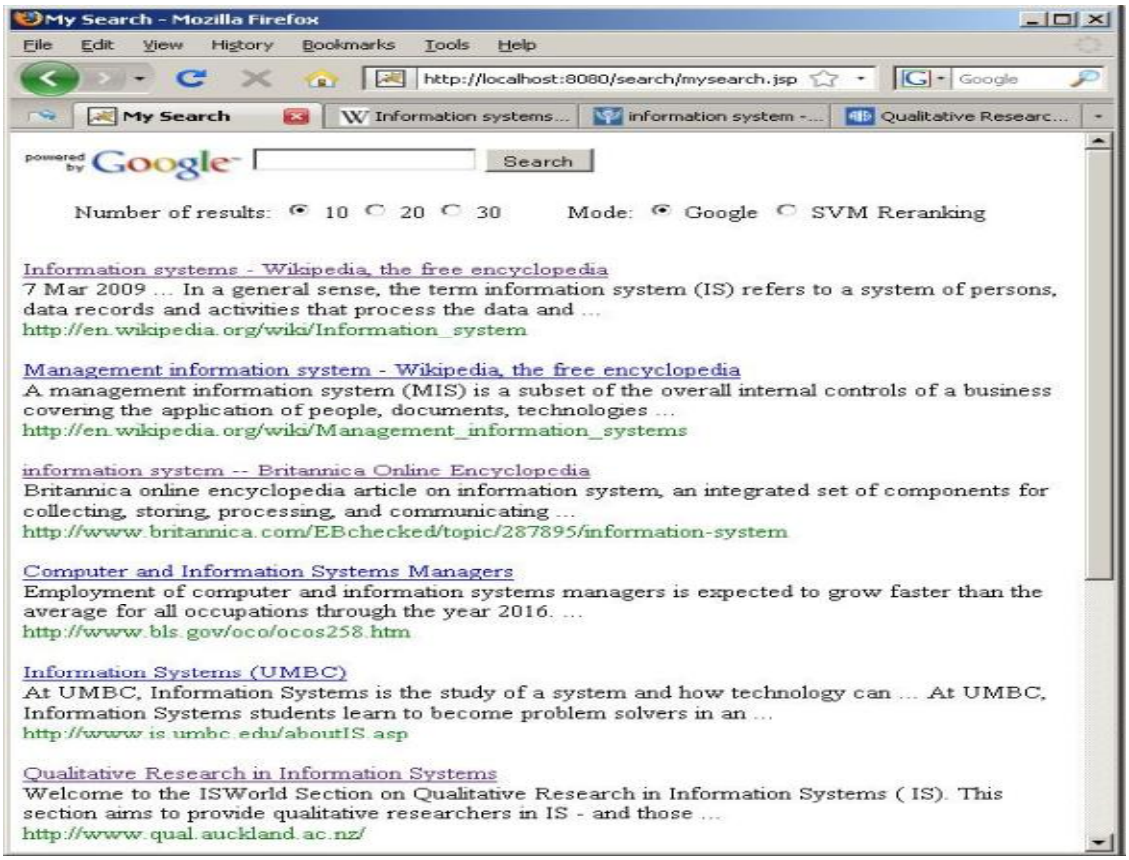
- Εγγραφή στο δεύτερο log αρχείο των παραπάνω, ως εξής:

- Αφαιρούμε από το συνολικό string το κομμάτι που επεξεργαστήκαμε και προχωράμε παρακάτω ομοίως

5. Οι επαναλήψεις τελειώνουν όταν κάνουμε τα παραπάνω βήματα για όλα τα αποτελέσματα
6. Γράφουμε στο πρώτο log αρχείο τα παρακάτω στοιχεία:



7. Εμφανίζουμε τα αποτελέσματα στο χρήστη, με τον τρόπο που φαίνεται στην παρακάτω εικόνα:



Έχουμε καταγράψει και εμφανίσει λοιπόν τα αποτελέσματα της αναζήτησης του χρήστη. Τώρα πρέπει να καταγραφούν και τα αποτελέσματα στα οποία επιλέγει να κάνει κλικ. Για να επιτευχθεί αυτό, οι σύνδεσμοι των αποτελεσμάτων δεν δείχνουν απ' ευθείας στο αποτέλεσμα, αλλά καλούν ένα Java Servlet που έχουμε δημιουργήσει. Το Servlet αυτό (LogRedirect.class) καταγράφει στο πρώτο log αρχείο (out.log) τα παρακάτω στοιχεία:

- Ημερομηνία και ώρα που έγινε το κλικ
- URL διεύθυνση του αποτελέσματος
- Αύξων αριθμός του ερωτήματος (query id)
- IP διεύθυνση του χρήστη

Κατόπιν ανοίγει ένα νέο παράθυρο στον περιηγητή και γίνεται ανακατεύθυνση στη σελίδα του αποτελέσματος που είχε ζητήσει ο χρήστης. Η διαδικασία της καταγραφής του κλικ είναι αόρατη στο χρήστη σε κανονικές συνθήκες και δεν καθυστερεί ιδιαίτερα τη φόρτωση της ζητούμενης σελίδας.

4.2 Μορφή log αρχείων

Όπως αναφέραμε και στην παραπάνω παράγραφο, χρησιμοποιούμε δύο αρχεία για την καταγραφή της δραστηριότητας των χρηστών της μηχανής μας αναζήτησης.

4.2.1 Αρχείο για εξαγωγή προτιμήσεων (out.log)

Το πρώτο αρχείο (out.log) που περιλαμβάνει τις λέξεις – κλειδιά του κάθε ερωτήματος, τις URL διευθύνσεις των αποτελεσμάτων που παρουσιάστηκαν στο χρήστη και των κλικ που έκανε. Το αρχείο αυτό θα χρησιμοποιηθεί για να εξάγουμε τις σχετικές προτιμήσεις του χρήστη, σύμφωνα με τις στρατηγικές που αναλύσαμε.

Η κάθε γραμμή του αρχείου αυτού μπορεί να είναι είτε γραμμή που να αναπαριστά ένα ερώτημα του χρήστη, είτε γραμμή που να αναπαριστά ένα κλικ του χρήστη. Οι γραμμές των ερωτημάτων αποθηκεύονται με την εξής μορφή:

<Ημερομηνία & ώρα> q:<Ερώτημα> qid:<Query ID> ip:<IP διεύθυνση> s:NA ref:NA n:<Αριθμός αποτελεσμάτων> <URL Result #1>*<URL Result #2>* … *<URL Result #n>

Οι γραμμές των κλικ αποθηκεύονται με την εξής μορφή:

<Ημερομηνία & ώρα> abs:<URL Click> qid:<Query ID> ip:<IP διεύθυνση> s:NA

Το αρχείο αυτό θα έχει δηλαδή M+N γραμμές οπού M ο αριθμός των ερωτημάτων που έχει θέσει ο χρήστης, και N ο αριθμός των κλικ σε συνδέσμους που έχει κάνει. Στην επόμενη εικόνα φαίνεται το αρχείο για το παράδειγμά μας, όπου η γραμμή 1 δείχνει το ερώτημα και τα αντίστοιχα αποτελέσματα, ενώ οι γραμμές 2,3 και 4 δείχνουν τα αποτελέσματα στα οποία έκανε κλικ ο χρήστης

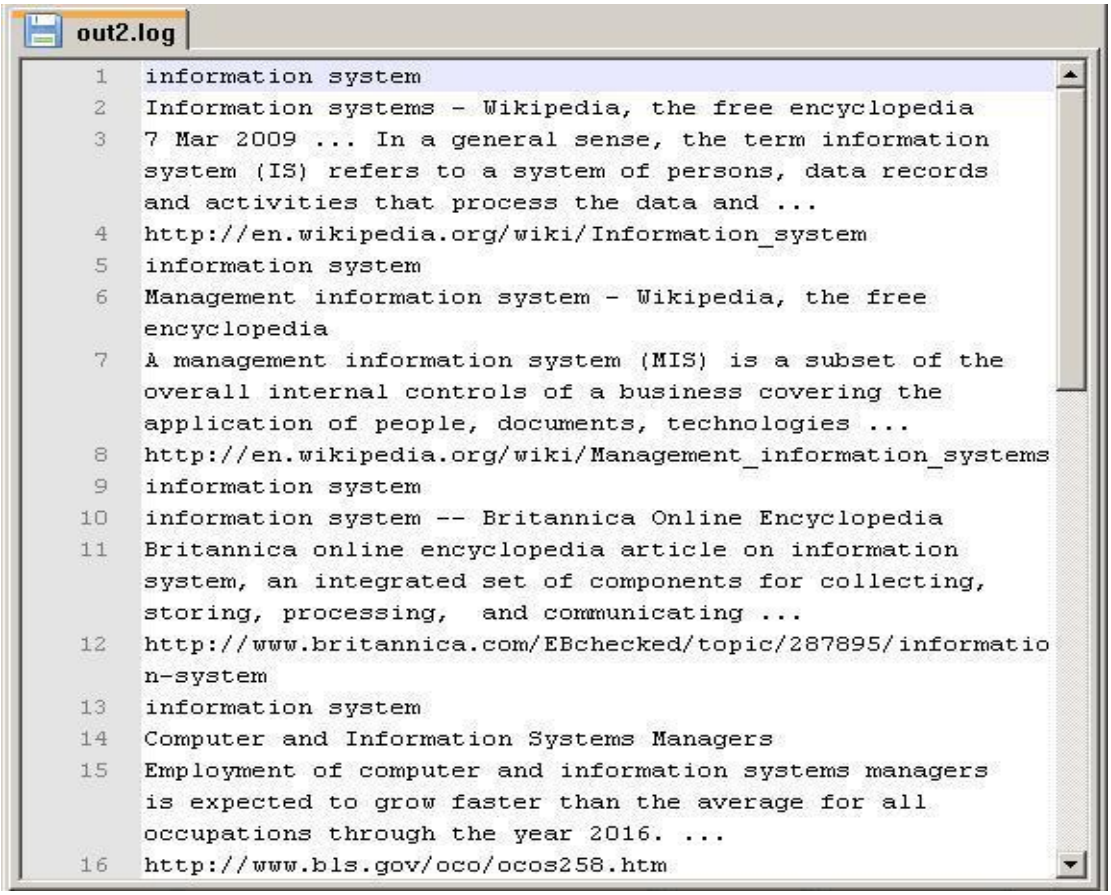

out.log
1 20090315182438 q:information system qid:1783 ip:127.0.0.1 s:NA ref:NA n:10 http://en.wikipedia.org/wiki/Information_system*http://en.wikipedia.org/wiki/Management_information_systems*http://www.britannica.com/EBchecked/topic/287895/information-system*http://www.bls.gov/oco/ocos258.htm*http://www.is.umbc.edu/aboutIS.asp*http://www.qual.auckland.ac.nz/*http://www.disa.mil/*http://www.wiley.com/bw/journal.asp?ref=1350-1917*http://www.aisnet.org/*http://egsc.usgs.gov/isb/pubs/gis_poster/
2 20090315182450 abs:http://en.wikipedia.org/wiki/Information_system qid:1783 ip:127.0.0.1 s:N/A
3 20090315182500 abs:http://www.britannica.com/EBchecked/topic/287895/information-system qid:1783 ip:127.0.0.1 s:N/A
4 20090315182510 abs:http://www.qual.auckland.ac.nz/ qid:1783 ip:127.0.0.1 s:N/A
5

4.2.2 Αρχείο για δημιουργία ευρετηρίου (out2.log)

Στο δεύτερο αρχείο (out2.log), με του οποίου τα περιεχόμενα θα δημιουργηθεί ευρετήριο, περιλαμβάνονται τα ερωτήματα του χρήστη, οι τίτλοι, οι περιλήψεις και οι URL διευθύνσεις των αποτελεσμάτων που του παρουσιάστηκαν. Για κάθε αποτέλεσμα αποθηκεύονται τέσσερις γραμμές στο αρχείο ως εξής:

3 Ερώτημα
4 Τίτλος
5 Περίληψη
6 URL διεύθυνση

Για ένα ερώτημα δηλαδή που έχουν εμφανιστεί N αποτελέσματα, θα αποθηκευτούν 4·N γραμμές στο αρχείο αυτό.



Στην παραπάνω εικόνα βλέπουμε την πληροφορία που αποθηκεύεται στο αρχείο για τα 4

πρώτα αποτελέσματα του ερωτήματος “information system”. Η πρώτη σειρά περιέχει τις λέξεις κλειδιά του ερωτήματος. Η δεύτερη σειρά έχει τον τίτλο του πρώτου αποτελέσματος, η τρίτη σειρά έχει την περίληψή του, ενώ η τέταρτη σειρά έχει τη διεύθυνσή του. Οι σειρές 58 έχουν την αντίστοιχη πληροφορία για το δεύτερο αποτέλεσμα, κ.ο.κ. Παρατηρούμε ότι η πληροφορία αυτή είναι ακριβώς αυτή που εμφανίζεται στο χρήστη.

Κεφαλαιο 5 : **Ανάλυση μοντέλου SVM και χρήση**

5.1 Ορισμός και ιδιότητες των SVMs

Τα Support Vector Machines (SVMs) είναι ένα σύνολο μεθόδων εκμάθησης που χρησιμοποιούνται για προβλήματα ταξινόμησης και παλινδρομικής ανάλυσης. Η κύρια ιδέα των SVM είναι να κατασκευαστεί ένα υπερεπίπεδο, έτσι ώστε η απόσταση του διαχωρισμού μεταξύ των θετικών και αρνητικών παραδειγμάτων να μεγιστοποιείται. Τα διανύσματα των πιο κοντινών στοιχείων στο υπερεπίπεδο αυτό είναι τα υποστηρικτικά διανύσματα (support vectors).

Αυτή η επιθυμητή ιδιότητα επιτυγχάνεται ακολουθώντας την αρχή της Ελαχιστοποίηση του Δομικού Ρίσκου (Structural Risk Minimization) από τη θεωρία της μηχανικής μάθησης. Η ιδέα της ελαχιστοποίησης του δομικού ρίσκου είναι να βρεθεί μια υπόθεση h για την οποία μπορούμε να εγγυηθούμε το χαμηλότερο πραγματικό σφάλμα. Το πραγματικό σφάλμα της h είναι η πιθανότητα της h να κάνει λάθος σε ένα τυχαία επιλεγμένο παράδειγμα το οποίο δεν έχει δει στο παρελθόν. Το πλεονέκτημα της τεχνικής αυτής είναι ότι επιτυγχάνονται καλές επιδόσεις στα προβλήματα ταξινόμησης χωρίς να ενσωματώνεται γνώση από τον τομέα του προβλήματος.

Βλέποντας τα δεδομένα εισόδου σαν δύο σύνολα διανυσμάτων σε ένα νδιάστατο χώρο, το SVM θα κατασκευάσει ένα διαχωριστικό υπερεπίπεδο σε αυτόν το χώρο, που θα μεγιστοποιεί την απόσταση μεταξύ των δύο συνόλων. Για τον υπολογισμό της απόστασης αυτής, κατασκευάζονται δύο παράλληλα υπερεπίπεδα, ένα σε κάθε πλευρά του διαχωριστικού υπερεπιπέδου, τα οποία

“σπρώχνονται” πάνω στα δύο σύνολα δεδομένων. Διαισθητικά, ένας καλός διαχωρισμός επιτυγχάνεται από το υπερεπίπεδο που έχει τη μεγαλύτερη απόσταση από τα γειτονικά σημεία δεδομένων και των δύο συνόλων, δεδομένου ότι σε γενικές γραμμές όσο μεγαλύτερη είναι η απόσταση τόσο καλύτερο είναι το λάθος γενίκευσης του ταξινομητή.

Η ταξινόμηση των δεδομένων είναι μια κοινή ανάγκη στο πεδίο της μηχανικής μάθησης. Ας υποθέσουμε ότι δίνονται κάποια σημεία δεδομένων που ανήκουν στα δύο σύνολα, και ο στόχος είναι να αποφασίσουμε σε ποιο σύνολο θα μπει ένα νέο σημείο δεδομένων. Στην περίπτωση των SVM, ένα σημείο δεδομένων θεωρείται σαν ένα διάνυσμα ρδιαστάσεων, και θέλουμε να ξέρουμε αν μπορούμε να χωρίσουμε αυτά τα σημεία με ένα p–1διάστατο υπερεπίπεδο. Αυτό ονομάζεται γραμμικός ταξινομητής. Υπάρχουν πολλά υπερεπίπεδα που θα μπορούσαν να ταξινομήσουν τα δεδομένα. Ωστόσο, ενδιαφερόμαστε επιπλέον να διαπιστώσουμε εάν μπορούμε να πετύχουμε το μέγιστο διαχωρισμό (απόσταση) μεταξύ των δύο κλάσεων. Με αυτό εννοούμε ότι διαλέγουμε το υπερεπίπεδο, έτσι ώστε η απόσταση από το υπερεπίπεδο στο πλησιέστερο σημείο δεδομένων να μεγιστοποιείται. Αυτό σημαίνει ότι η κοντινότερη απόσταση ανάμεσα σε ένα σημείο στο ένα διαχωρισμένο υπερεπίπεδο και σε ένα σημείο στο άλλο διαχωρισμένο υπερεπίπεδο μεγιστοποιείται. Αν υπάρχει ένα τέτοιο υπερεπίπεδο, είναι γνωστό ως το υπερεπίπεδο μέγιστουδιαχωρισμού, και ένας τέτοιος γραμμικός ταξινομητής είναι γνωστός ως ένας ταξινομητής μέγιστουδιαχωρισμού.

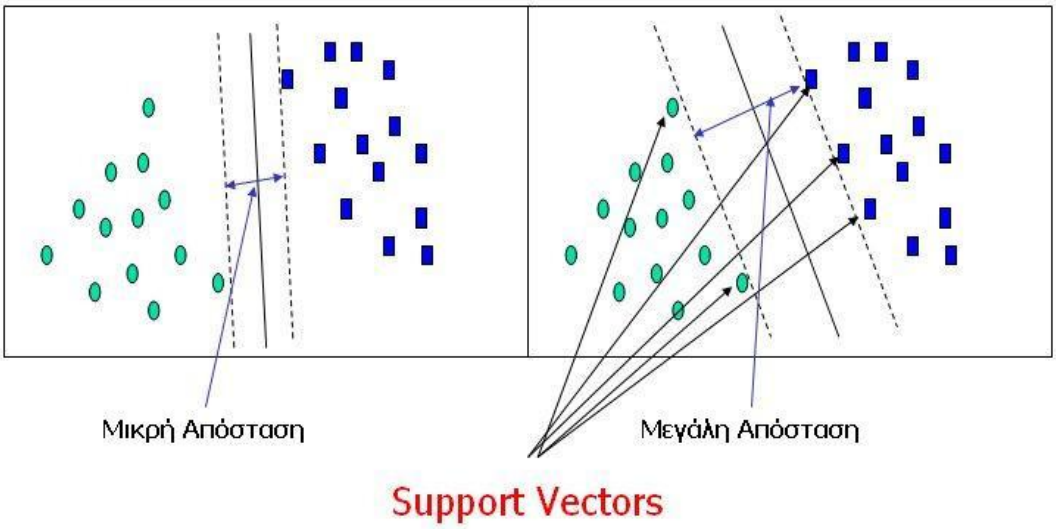
Τα SVMs ανήκουν στην κατηγορία των γενικευμένων γραμμικών ταξινομητών. Μια ειδική ιδιότητά τους είναι ότι ταυτόχρονα ελαχιστοποιούν το εμπειρικό σφάλμα ταξινόμησης και μεγιστοποιούν τη γεωμετρική απόσταση. Ως εκ τούτου, είναι ταξινομητές μέγιστουδιαχωρισμού.

Μία αξιοσημείωτη ιδιότητα των SVMs είναι ότι η ικανότητά τους να μαθαίνουν είναι ανεξάρτητη από τις διαστάσεις του χώρου χαρακτηριστικών. Τα SVMs μετράνε την πολυπλοκότητα των υποθέσεων με βάση την απόσταση που μπορούν να διαχωρίσουν τα στοιχεία, και όχι με βάση τον αριθμό των χαρακτηριστικών. Αυτό σημαίνει ότι μπορούμε να γενικεύσουμε ακόμη και με την παρουσία πάρα πολλών χαρακτηριστικών, αν τα στοιχεία μας μπορούν να διαχωριστούν με ένα ευρύ περιθώριο χρησιμοποιώντας συναρτήσεις από το χώρο υποθέσεων. Τα Support Vector Machines έχουν πολλά ελκυστικά χαρακτηριστικά. Είναι ένα σπάνιο παράδειγμα μεθοδολογίας όπου συνδυάζονται η γεωμετρική διαίσθηση, τα κομψά μαθηματικά, οι θεωρητικές εγγυήσεις και οι πρακτικοί αλγόριθμοι. Μπορούν να εφαρμοστούν αποτελεσματικά σε ένα ευρύ φάσμα προβλημάτων ταξινόμησης. Κλιμακώνονται σε τεράστια σύνολα δεδομένων και είναι ανεξάρτητα του τομέα του προβλήματος. Επιπλέον, μπορούν να αναπτυχθούν αποτελεσματικές συναρτήσεις πυρήνα για κάθε συγκεκριμένο πρόβλημα, προκειμένου να επιτευχθούν ακόμα καλύτερα αποτελέσματα. Τα SVMs έχουν πολλές επιτυχημένες εφαρμογές στον τομέα της βιοπληροφορικής (ταξινόμηση δεδομένων μικρό-συστοιχιών), της ανίχνευσης προσώπου και αναγνώρισης χειρογράφου κειμένου. Είναι επίσης πολύ καλά για την κατηγοριοποίηση κειμένου.

5.2 Διδιάστατο παράδειγμα SVM

Σε ένα ιδεατό διδιάστατο παράδειγμα, τα στοιχεία της μίας κατηγορίας βρίσκονται στο κάτω αριστερό άκρο ενώ τα στοιχεία της άλλης κατηγορίας βρίσκονται στο πάνω δεξιό άκρο, και έτσι είναι τελείως διαχωρισμένα. Προσπαθούμε να βρούμε ένα υπερεπίπεδο 1διάστασης (δηλαδή μια γραμμή) που να χωρίζει τα στοιχεία των δύο κατηγοριών. Οι πιθανές γραμμές που το επιτυγχάνουν αυτό είναι άπειρες, ο στόχος είναι όμως να προσδιορίσουμε την καλύτερη δυνατή. Ο SVM αλγόριθμος βρίσκει τη γραμμή (ή το υπερεπίπεδο – στη γενική περίπτωση) έτσι ώστε η απόσταση μεταξύ των στοιχείων των δύο κατηγοριών είναι η μέγιστη.

Στο παρακάτω σχήμα, οι διακεκομμένες γραμμές που είναι σχεδιασμένες παράλληλα στην διαχωριστική γραμμή δείχνουν την απόσταση μεταξύ της διαχωριστικής γραμμής και των πλησιέστερων διανυσμάτων στη γραμμή. Η απόσταση μεταξύ των διακεκομμένων γραμμών ονομάζεται περιθώριο. Τα διανύσματα (σημεία) που περιορίζουν το πλάτος του περιθωρίου είναι τα υποστηρικτικά διανύσματα (support vectors). Είναι φανερό ότι η γραμμή στο δεξί σχήμα διαχωρίζει πολύ καλύτερα τα στοιχεία από αυτήν στο αριστερό σχήμα.



Εικόνα 6 : SVM Margins & Support Vectors

5.3 Αλγόριθμος SVM

Με δεδομένα αρχεία καταγραφής της συμπεριφοράς των χρηστών σε μια δικτυακή μηχανή αναζήτησης, δείξαμε παραπάνω με ποια στρατηγική μετατρέπουμε τις εγγραφές του log αρχείου σε κρίσεις προτίμησης. Θα παρουσιάσουμε τώρα τον αλγόριθμο που χρησιμοποιεί το SVM για να μάθει από αυτές τις προτιμήσεις.

Θεωρούμε σαν είσοδο του αλγορίθμου προτιμήσεις της μορφής

$d_i >_q d_j$ (1)

όπου d_i, d_j έγγραφα για ένα δεδομένο ερώτημα q. Η παραπάνω σχέση δείχνει ότι το d_i προτιμάται σε σχέση με το d_j για ένα δεδομένο q. Για το μοντέλο ανάκτησης χρησιμοποιούμε μια γραμμική συνάρτηση ανάκτησης:

$rel(d_i,q) = w * \Phi(d_i,q)$ (2)

όπου $\Phi(d_i,q)$ είναι μια συνάρτηση που αντιστοιχίζει έγγραφα και ερωτήματα σε ένα διάνυσμα χαρακτηριστικών (feature vector). Διαισθητικά, μπορεί να θεωρηθεί σαν ένα διάνυσμα χαρακτηριστικών που περιγράφει την ποιότητα της αντιστοίχησης μεταξύ ενός εγγράφου d_i και του ερωτήματος q. Το w είναι ένα διάνυσμα βάρους που αναθέτει βάρη σε καθένα από τα χαρακτηριστικά στο Φ, και άρα δίνοντάς μας μια συνάρτηση ανάκτησης πραγματικής αξίας, όπου ένα μεγαλύτερο σκορ δηλώνει ότι ένα έγγραφο d_i είναι πιο σχετικό για το ερώτημα q. Το έργο της εκμάθησης μιας συνάρτησης κατάταξης ισοδυναμεί με την εκμάθηση ενός βέλτιστου w.

Ξαναγράφουμε τη σχέση (1) ως εξής:

$W* \Phi(d_i,q) > w* \Phi(d_j,q)$

Στη συνέχεια προσθέτουμε ένα περιθώριο και μηαρνητικές slack μεταβλητές ώστε να επιτρέψουμε κάποιους από τους περιορισμούς των προτιμήσεων να παραβιαστούν, όπως γίνεται και στα SVMs ταξινόμησης. Αυτό δείχνει ένα περιορισμό προτίμησης πάνω από το w

$W*\Phi(d_i,q) \geq w*\Phi(d_j,q) + 1\ \xi_{ij}$

Αν και δεν μπορούμε αποτελεσματικά να βρούμε ένα που να ελαχιστοποιεί τον αριθμό των παραβιασμένων περιορισμών, μπορούμε να ελαχιστοποιήσουμε ένα άνω όριο στον αριθμό των παραβιασμένων περιορισμών, $\sum \xi_{i,j}$. Η ταυτόχρονη μεγιστοποίηση του περιθωρίου οδηγεί στο παρακάτω δευτεροβάθμιο κυρτό πρόβλημα βελτιστοποίησης:

$$\min_{w,\xi_{ij}} \frac{1}{2} w \cdot w + C \sum_{ij} \xi_{ij}$$

$$\text{subject to } \forall (q,i,j) : \; w \cdot \Phi(d_i, q) \geq w \cdot \Phi(d_j, q) + 1 - \xi_{ij}$$

$$\forall i,j : \; \xi_{ij} \geq 0$$

Στην εργασία μας θα χρησιμοποιήσουμε τον αλγόριθμο SVM για την επίλυση του παραπάνω προβλήματος βελτιστοποίησης.

5.4 Εξαγωγή σχετικών προτιμήσεων χρήστη

Για να εκπαιδεύσουμε το μοντέλο που θα χρησιμοποιήσουμε για την ανακατάταξη των αποτελεσμάτων του Google, θα πρέπει να τροφοδοτήσουμε τον αλγόριθμο Support Vector Machine με τις σχετικές προτιμήσεις του χρήστη. Τις προτιμήσεις αυτές θα τις εξάγουμε έμμεσα από το αρχείο καταγραφής της δραστηριότητας του χρήστη (out.log).

Στο αρχείο αυτό καταγράφονται όλα τα αποτελέσματα που έχουν παρουσιαστεί στο χρήστη για τα ερωτήματα που έκανε, καθώς και τα κλικ που επέλεξε να κάνει. Αναλύοντας τα δεδομένα αυτά χρησιμοποιώντας τις στρατηγικές που περιγράψαμε παραπάνω, θα εξάγουμε τις σχετικές προτιμήσεις του χρήστη.

Για τη διαδικασία αυτήν της ανάλυσης του αρχείου καταγραφής και της εξαγωγής των προτιμήσεων του χρήστη από αυτό, θα χρησιμοποιήσουμε το Osmot. Το Osmot έχει έτοιμες συναρτήσεις για την ανάγνωση και ανάλυση log αρχείων καθώς και την εξαγωγή προτιμήσεων από αυτά. Χρησιμοποιώντας αυτές σαν βάση, κάναμε τις απαραίτητες αλλαγές και προσθήκες για τις ανάγκες τις εργασίας μας.

Η διαδικασία περιλαμβάνει τα εξής βήματα:

1 Ανάλυση του αρχείου καταγραφής και καταγραφή των ερωτημάτων και των κλικ.
2 Εξαγωγή προτιμήσεων για κάθε ερώτημα

Το πρώτο βήμα είναι να διαβάσουμε το log αρχείο και να μετατρέψουμε τα δεδομένα του σε μορφή που να είναι εύκολα επεξεργάσιμη. Χρησιμοποιώντας την συνάρτηση parseLog του

Osmot, αναλύουμε το log σε εγγραφές από ερωτήματα (QueryEntry) και σε εγγραφές από κλικ (ClickEntry).

Αν έχει γίνει κλικ σε κάποιο από τα αποτελέσματα ενός ερωτήματος, καλείται η συνάρτηση που εξάγει τις σχετικές προτιμήσεις του χρήστη. Στην περίπτωση που δεν έχει γίνει κλικ σε κάποιο αποτέλεσμα, δεν εξάγεται κάποια προτίμηση με βάση τις απλές στρατηγικές. Μπορούμε όμως να εξάγουμε προτιμήσεις από αυτό, αν το ερώτημα αποτελεί μέρος μιας αλυσίδας ερωτημάτων.

Στην απλή περίπτωση του καθενός ερωτήματος ξεχωριστά, σε κάθε αποτέλεσμα που παρουσιάστηκε στο χρήστη εκχωρούμε μια τιμή που δηλώνει την σχετική προτίμηση του χρήστη για το αποτέλεσμα. Οι τιμές αυτές προκύπτουν ακολουθώντας τις τεχνικές που αναλύσαμε στο κεφάλαιο 2, δηλαδή την στρατηγική “Click > Skip above” καθώς και τη στρατηγική για προσθήκη υπάρχουσας γνώσης. Ακολουθεί ο αλγόριθμος που χρησιμοποιούμε για να βρούμε την κατάλληλη αξία για κάθε αποτέλεσμα:

- Κάθε αποτέλεσμα στο οποίο δεν έγινε κλικ από το χρήστη παίρνει την τιμή 1.
- Για τα αποτελέσματα στα οποία έγινε κλικ:
  - Το αποτέλεσμα που βρίσκεται χαμηλότερα στην κατάταξη του Google παίρνει τιμή 2.
  - Το επόμενο αποτέλεσμα που βρίσκεται πιο πάνω στην κατάταξη παίρνει την τιμή 3.
  - Ομοίως συνεχίζουμε για τα υπόλοιπα αποτελέσματα στα οποία έγινε κλικ.

Για το παράδειγμα όπου ο χρήστης έκανε κλικ στους συνδέσμους l1, l3, και l6.

l1 l2 l3 l4 l5 l6 l7

Ακολουθώντας τα βήματα του παραπάνω αλγορίθμου έχουμε:

- Τα αποτελέσματα l2, l4, l5, και l7 παίρνουν όλα την τιμή προτίμησης 1, καθώς δεν έγινε κλικ σε αυτά.
- Το αποτέλεσμα l6 παίρνει τιμή προτίμησης 2, καθώς είναι το χαμηλότερο στην κατάταξη αποτέλεσμα στο οποίο έγινε κλικ.

- Το αποτέλεσμα l3 παίρνει τιμή προτίμησης 3, αφού είναι το επόμενο αποτέλεσμα στην κατάταξη στο οποίο έγινε κλικ.
- Το αποτέλεσμα l1 παίρνει τιμή προτίμησης 4, καθώς είναι το πρώτο αποτέλεσμα στην κατάταξη αποτέλεσμα στο οποίο έγινε κλικ.

Παρατηρούμε ότι τηρούνται όλες οι απαιτήσεις της στρατηγικής “Click > Skip above” :

value(l3) > value(l2), value(l6) > value(l2) , value(l6) > value(l4) , value(l6) > value(l5)

5.5 Επιλογή χαρακτηριστικών (features)

Για την εκπαίδευση της συνάρτησης ανάκτησης χρησιμοποιώντας τον αλγόριθμο Support Vector Machine, είναι απαραίτητο να σχεδιαστεί μια αντιστοίχηση χαρακτηριστικών μεταξύ ενός ερωτήματος q και ενός εγγράφου d. Έχοντας λοιπόν τα χαρακτηριστικά κάθε αποτελέσματος και τις προτιμήσεις του χρήστη για αυτά, το SVM μπορεί να εξάγει ένα μοντέλο. Ξέροντας ουσιαστικά τι χαρακτηριστικά έχουν τα αποτελέσματα που προτιμά ο χρήστης, μπορούμε να αλλάξουμε την κατάταξη των αποτελεσμάτων χρησιμοποιώντας το μοντέλο αυτό.

Παρακάτω ακολουθούν τα 11 χαρακτηριστικά που υλοποιήσαμε στην εργασία. Ο τρόπος υπολογισμού τους εξηγείται στην επόμενη ενότητα .

1 Ομοιότητα ερωτήματος – τίτλου εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου του Lucene.

2 Ομοιότητα ερωτήματος – περίληψης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου του Lucene.

3 Ομοιότητα ερωτήματος – URL διεύθυνσης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου του Lucene.

4 Ομοιότητα ερωτήματος – τίτλου εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου tf idf.

5 Ομοιότητα ερωτήματος – περίληψης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου tf idf.

6 Ομοιότητα ερωτήματος – URL διεύθυνσης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου tf idf.

7 Ομοιότητα ερωτήματος – τίτλου εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου BM25.

8 Ομοιότητα ερωτήματος – περίληψης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου BM25.

9 Ομοιότητα ερωτήματος – URL διεύθυνσης εγγράφου, υπολογισμένη με τη συνάρτηση ομοιότητας κειμένου BM25.

10 Βαθμολογία με βάση τη θέση κατάταξης του εγγράφου στο Google.

11 Domain της URL διεύθυνσης του εγγράφου

Η λίστα των χαρακτηριστικών που χρησιμοποιήσαμε σίγουρα έχει πολλά περιθώρια για βελτίωση, καθώς υλοποιήσαμε μόνο μερικά βασικά χαρακτηριστικά που είναι σημαντικά για την κατάταξη και σχετικά απλά να υλοποιηθούν. Το σύνολο αυτό των χαρακτηριστικών μπορεί εύκολα να επεκταθεί και το σημαντικό είναι ότι κάθε προσθήκη ενός χαρακτηριστικού αυξάνει ακόμα περισσότερο την ακρίβεια του εξαγόμενου μοντέλου.

5.6 Υπολογισμός των feature vectors

5.6.1 Συνάρτηση ομοιότητας κειμένου του Lucene

Η συνάρτηση tf idf αναλύθηκε στο κεφάλαιο 2, και πάνω σε αυτήν βασίζεται και η συνάρτηση του Lucene. Το Lucene χρησιμοποιεί ένα συνδυασμό του Μοντέλου Διανυσματικού Χώρου (Vector Space Model – VSM) και του Boolean μοντέλου για να καθορίσει πόσο σχετικό είναι ένα έγγραφο με το ερώτημα του χρήστη. Πρώτα χρησιμοποιεί το Boolean μοντέλο για να περιορίσει τα έγγραφα που πρέπει να βαθμολογηθούν. Η ιδέα πίσω από το VSM είναι ότι όσο περισσότερες φορές εμφανίζεται ένας όρος ερωτήματος σε ένα έγγραφο σε σχέση με τον αριθμό των φορών που εμφανίζεται σε όλα τα έγγραφα στη συλλογή, τόσο πιο σχετικό είναι το έγγραφο αυτό με το ερώτημα. Η βαθμολογία ενός ερωτήματος q για ένα έγγραφο d συσχετίζεται με το εσωτερικό γινόμενο μεταξύ των διανυσμάτων του εγγράφου και του ερωτήματος σε ένα Μοντέλο Διανυσματικού Χώρου. Ένα έγγραφο του οποίο το διάνυσμα βρίσκεται πιο κοντά στο διάνυσμα του ερωτήματος στο μοντέλο αυτό, βαθμολογείται υψηλότερα. Η βαθμολογίας ομοιότητας του Lucene υπολογίζεται λοιπόν ως εξής:

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t \text{ in } q} \left(tf(t \text{ in } d) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t,d) \right)$$

Όπου

1. tf (t in d) συσχετίζεται με τη συχνότητα του όρου, ορισμένη από τον αριθμό των φορών που ο όρος t εμφανίζεται στο έγγραφο d. Τα έγγραφα που έχουν περισσότερες εμφανίσεις ενός δεδομένου όρου λαμβάνουν υψηλότερη βαθμολογία. Ο προκαθορισμένος υπολογισμός για το tf (t in d) είναι:

   tf (t in d) = frequency $^{1/2}$

2. idf (t) είναι η αντίστροφη συχνότητα εγγράφου. Η τιμή αυτή συσχετίζεται με το αντίστροφο του docFreq (δηλαδή του

αριθμού των εγγράφων στα οποία εμφανίζεται ο όρος t). Αυτό σημαίνει ότι πιο σπάνιοι όροι δίνουν μια υψηλότερη συνεισφορά στη συνολική βαθμολογία. Ο προκαθορισμένος υπολογισμός για το idf (t) είναι:

$$idf(t) = 1 + \log(\frac{numDocs}{docFreq + 1})$$

3. coord(q, d) είναι ένας παράγοντας βασισμένος στο πόσοι από τους όρους του ερωτήματος βρίσκονται στο συγκεκριμένο έγγραφο. Τυπικά ένα έγγραφο που περιλαμβάνει περισσότερους από τους όρους του ερωτήματος λαμβάνει μια υψηλότερη βαθμολογία από ένα άλλο με λιγότερους όρους.
4. queryNorm(q) είναι ένας παράγοντας κανονικοποίησης που χρησιμοποιείται για να κάνει βαθμολογίες μεταξύ ερωτημάτων συγκρίσιμες. Αυτός ο παράγοντας δεν επηρεάζει την κατάταξη των εγγράφων (αφού όλα τα έγγραφα πολλαπλασιάζονται με τον ίδιο παράγοντα), αλλά απλά προσπαθεί να κάνει τις βαθμολογίες μεταξύ διαφορετικών ερωτημάτων συγκρίσιμες.
5. t.getBoost() και norm(t, d) είναι παράγοντες που μπορούν προαιρετικά να προάγουν ένα συγκεκριμένο όρο t για το ερώτημα q. Εμείς θα χρησιμοποιήσουμε τις προκαθορισμένες τιμές του Lucene, που είναι 1.

Μπορούμε να συνοψίσουμε τα παραπάνω ως εξης :

- Έγγραφα που περιλαμβάνουν όλους τους όρους του ερωτήματος είναι καλά
- Αντιστοιχίες σπάνιων λέξεων μεταξύ εγγράφου και ερωτήματος είναι καλύτερες από συνηθισμένες λέξεις.
- Τα μεγάλα σε έκταση έγγραφα δεν είναι τόσο καλά όσο τα μικρότερα.
- Έγγραφα που επαναλαμβάνουν πολλές φορές τους όρους του ερωτήματος είναι καλά.

5.6.2 Συνάρτηση ομοιότητας κειμένου BM25

Η συνάρτηση ομοιότητας BM25 υπολογίζει τη βαθμολογία ενός εγγράφου d σε σχέση με ένα ερώτημα q ως εξής:

$$R(q,d) = \sum_{t \text{ in } q} \frac{tf(t \text{ in } d)}{k_1\left((1-b) + b \cdot \frac{l_d}{avl_d}\right) + tf(t \text{ in } d)} \cdot idf(t)$$

Όπου τα tf και idf είναι όπως και πριν, ld είναι το μήκος του εγγράφου d, avld είναι το μέσο μήκος των εγγράφων στη συλλογή, k1 είναι μια ελεύθερη παράμετρος που συνήθως είναι 2 και b μια άλλη ελεύθερη παράμετρος που συνήθως είναι 0.75. Στην εργασία μας χρησιμοποιήσαμε την υλοποίηση της συνάρτησης BM25 του Joaquín PérezIglesias. Για να γίνει αυτό, καταγραφούμε το μήκος κάθε πεδίου του εγγράφου τη στιγμή που το προσθέτουμε στο ευρετήριο με το Lucene, έτσι ώστε στο τέλος να έχουμε και το μέσο μήκος, που χρειάζεται για τον υπολογισμό της ομοιότητας BM25.

5.6.3 Βαθμολογία θέσης κατάταξης στο Google

Για τη βαθμολογία με βάση τη θέση κατάταξης του αποτελέσματος στο Google, χρησιμοποιούμε τον ακόλουθο υπολογισμό:

$$G(q,d) = \begin{cases} 1 - r(q,d)/10, & r(q,d) < 10 \\ 0, & r(q,d) > 10 \end{cases}$$

Όπου r(q,d) είναι η θέση κατάταξη του αποτελέσματος d για το ερώτημα q, που δίνει το Google.

5.6.4 Domain του αποτελέσματος

Τέλος, το χαρακτηριστικό του domain είναι ουσιαστικά 73 χαρακτηριστικά που περιλαμβάνουν μερικά από τα υπάρχοντα Toplevel domains του διαδικτύου. Το θεωρούμε όμως σαν ένα χαρακτηριστικό, καθώς το πολύ ένα από αυτά τα χαρακτηριστικά θα έχει την τιμή 1 (αν το έγγραφο έχει διεύθυνση σε αυτό το domain), και τα υπόλοιπα θα έχουν την τιμή 0.Με το χαρακτηριστικό αυτό, μπορούμε να εξάγουμε εύκολα κάποια πιθανή προτίμηση του χρήστη για ιστοσελίδες κάποιου συγκεκριμένου domain. Για παράδειγμα, κάποιος που κάνει κλικ συχνά σε αποτελέσματα ιστοσελίδων από το domain .edu σημαίνει ότι τις προτιμάει και το μοντέλο μας θα δώσει μεγαλύτερο βάρος στο χαρακτηριστικό αυτό.

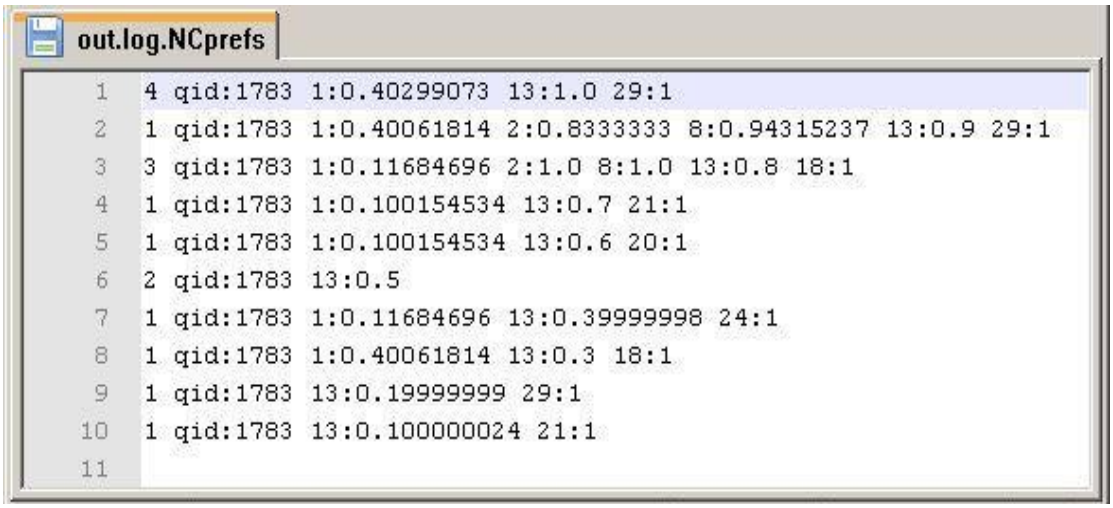
5.7 Μορφή αρχείου εισόδου SVM

Το αρχείο που θα δώσουμε σαν είσοδο στον SVM αλγόριθμο για να εκπαιδεύσει το ζητούμενο μοντέλο πρέπει να περιέχει τις προτιμήσεις του χρήστη για τα αποτελέσματα που του παρουσιάστηκαν και τα χαρακτηριστικά αυτών. Στις προηγούμενες παραγράφους του κεφαλαίου αναλύθηκε με ποια στρατηγική υπολογίζεται η τιμή προτίμησης για κάθε έγγραφο, καθώς και ο τρόπος υπολογισμού των χαρακτηριστικών των εγγράφων. Κάθε γραμμή του αρχείου αναπαριστά ένα έγγραφο και έχει την εξής μορφή:

<target> qid:<queryID> <feature>:<value> ... <feature>:<value>

όπου <target> είναι η τιμή προτίμησης του αποτελέσματος, <queryID> είναι ο αύξων αριθμός του ερωτήματος, <feature> είναι ο αύξων αριθμός του χαρακτηριστικού, και <value> είναι η τιμή για το αντίστοιχο χαρακτηριστικό. Τα ζευγάρια <feature>:<value> πρέπει να είναι καταταγμένα σε αύξουσα σειρά του αριθμού του χαρακτηριστικού. Χαρακτηριστικά τα οποία έχουν τιμή 0 μπορούν να παραλειφτούν.

Έτσι για το παράδειγμα μας, το αρχείο εισόδου του SVM θα πάρει την ακόλουθη μορφή:



Εικόνα 7 : Μορφή αρχείου εισόδου SVM

5.8 Εκπαίδευση SVM Μοντέλου

Έχοντας ετοιμάσει πλέον το αρχείο παραδειγμάτων που θα δώσουμε σαν είσοδο στον SVM αλγόριθμο, μπορούμε να τρέξουμε την κατάλληλη εφαρμογή του SVMlight που εκπαιδεύει το μοντέλο. Το SVMlight περιλαμβάνει μια εφαρμογή εκμάθησης (svm_learn) και μια εφαρμογή ταξινόμησης (svm_classify). Η εφαρμογή εκμάθησης έχει τρεις τρόπους λειτουργίας, για ταξινόμηση, για παλινδρόμηση και για κατάταξη, που είναι και αυτή που θα χρησιμοποιήσουμε εμείς. Η εφαρμογή ταξινόμησης μπορεί να χρησιμοποιηθεί για να εφαρμόσει το εκπαιδευμένο μοντέλο σε νέα έγγραφα.

Όταν χρησιμοποιείται η εφαρμογή εκμάθησης για ταξινόμηση, εξάγεται μια σχετική προτίμηση για κάθε ζευγάρι αποτελεσμάτων στο αρχείο εισόδου που έχουν διαφορετική τιμή προτίμησης (target value). Το ειδικό χαρακτηριστικό “qid” χρησιμοποιείται για να περιορίσει τις προτιμήσεις που θα δημιουργηθούν μόνο για αποτελέσματα που έχουν την ίδια τιμή “qid”.

Για παράδειγμα, αν το αρχείο έχει περιεχόμενο

```
3 qid:1 1:0.53 2:0.12
2 qid:1 1:0.13 2:0.1
7 qid:2 1:0.87 2:0.12
```

εξάγεται προτίμηση μόνο για τα πρώτα δύο παραδείγματα (ότι δηλαδή το πρώτο θα πρέπει να έχει θέση κατάταξης υψηλότερη από το δεύτερο), αλλά όχι για το τρίτο παράδειγμα, καθώς έχει διαφορετικό “qid”.

Τρέχοντας λοιπόν την εφαρμογή svm_learn με είσοδο το αρχείο που δημιουργήθηκε στην προηγούμενη ενότητα, θα εκπαιδευτεί το μοντέλο που φαίνεται στην παρακάτω εικόνα.

model.dat

```
1  SVM-light Version V6.02
2  0 # kernel type
3  3 # kernel parameter -d
4  1 # kernel parameter -g
5  1 # kernel parameter -s
6  1 # kernel parameter -r
7  empty# kernel parameter -u
8  29 # highest feature index
9  10 # number of training documents
10 10 # number of support vectors plus 1
11 0 # threshold b, each following line is a SV (starting with alpha*y)
12 2.2033490930117763 1:0.40299073 13:1 29:1 #
13 -1.3188353926446823 1:0.40061814 2:0.83333331 8:0.94315237 13:0.89999998 29:1 #
14 1.2800999763617817 1:0.11684696 2:1 8:1 13:0.80000001 18:1 #
15 -0.68985190795173889 1:0.10015453 13:0.69999999 21:1 #
16 -0.55501695711061338 1:0.10015453 13:0.60000002 20:1 #
17 1.7584471901929097 13:0.5 #
18 -0.43961179754822743 1:0.11684696 13:0.39999998 24:1 #
19 -1.3188353926446823 1:0.40061814 13:0.30000001 18:1 #
20 -0.91974481166652322 13:0.19999999 29:1 #
21
```

Εικόνα 8 : Μορφή αρχείου εκπαιδευμένου μοντέλου SVM

Οι πρώτες γραμμές του αρχείου του μοντέλου περιέχουν τις παραμέτρους της εκπαίδευσης. Οτιδήποτε είναι μετά τον χαρακτήρα “#” είναι επεξηγηματικό σχόλιο. Οι επόμενες γραμμές περιέχουν η κάθε μία από ένα διάνυσμα υποστήριξης (support vector), σε τυχαία σειρά.

Βιβλιογραφία

Βέγλης, Α., Πομπόρτσης, Α., Αβραάμ, Ε. (2004). *Έρευνα και συλλογή πληροφοριών στο διαδίκτυο*. Θεσσαλονίκη: Εκδόσεις Τζιόλα

Μήτρας, Μιχαήλ. Μηχανη Αναζητησης. Εκδοσεις Νεφελη. 2008

Κωνστανς Ζαφείρη. Μηχανές Αναζήτησης Και Directories. Αυτοέκδοση. 2005

Γεωργάκης, Κ. (2004) , Μελέτη Των Μηχανών Αναζήτησης Στο Διαδίκτυο Καθώς Και Των Τεχνικών Τους –Ανάπτυξη Ενός Μοντέλου –Προτύπου Ενιαίας Αναζήτησης , Πολυτεχνείο Κρήτης.

Μακριδάκης, Γ. (2003), Αξιολόγηση Μηχανών Αναζήτησης στο Διαδίκτυο και Ανάλυση Συμπεριφοράς των Χρηστών τους, Πολυτεχνείο Κρήτης.

Παναγής Ι. Περδικούρη, Α. Χριστοπούλου Ε., Θεοδωρίδης Ε (2007) Σημειώσεις Μαθήματος Ανάκτησης Πληροφορίας [online]. Πανεπιστήμιο Πάτρας. Τμήμα Μηχανικών Η/Υ και Πληροφορικής. Available from: http://mmlab.ceid.upatras.gr/ir/info/kef123.pdf

Παπαδόπουλος Α. (2005) Διαφάνειες Μαθήματος Ανάκτησης Πληροφορίας [online].Τμήμα Πληροφορικής. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης. Available from: http://delab.csd.auth.gr/courses/c_ir/

Στάθης Σταματάτος (2007) Διαφάνειες Μαθήματος Ανάκτησης Πληροφορίας [online]. Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων. Πανεπιστήμιο Αιγαίου. Available from: http://www.icsd.aegean.gr/lecturers/Stamatatos/courses/IR/index.htm

Franklin C. (2006) How Internet Search Engines Work [online] HowStuffWorks Inc. Available from:

<http://computer.howstuffworks.com/searchengine.htm>

Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search, ACM Transactions on Information Systems (TOIS), Vol. 25, No. 2 (April), 2007.

T. Joachims, SVMSupport Vector Machine, 1999.

Ingo,Steinwart, Andreas,Christmann. Support Vector Machines. Springer-varlag. 2008

Wikipedia, the free encyclopedia (2008) Metasearch Engine
[online]. Available from:
http://en.wikipedia.org/wiki/Metasearch_engine

http://www.searchenginemarketing.gr/blog/archives/53

Συμπεριφορά χρηστών

http://www.agelioforos.gr/default.asp?pid=7&ct=13&artid=133736

Support vector machine

http://en.wikipedia.org/wiki/Support_vector_machine

Information retrieval

http://el.wikipedia.org/wiki/Ανάκτηση_Πληροφοριών