ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΥΤΟΜΑΤΗ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΙΣΤΟΣΕΛΙΔΩΝ ΚΑΙ ΔΙΚΤΥΑΚΩΝ ΤΟΠΩΝ

Κανάκης Νικόλαος
ΑΜ 3107

Υπεύθυνος Καθηγητής:
Μπούρας Χρήστος,
Καθηγητής

ΠΑΤΡΑ,
ΣΕΠΤΕΜΒΡΙΟΣ 2009

ΠΡΟΛΟΓΟΣ

Λίγα γεγονότα στην ιστορία των υπολογιστών έχουν επηρεάσει τόσο βαθιά τη κοινωνία και τη καθημερινή ζωή των ανθρώπων, όσο η δημιουργία και η ανάπτυξη του Παγκόσμιου Ιστού. Ο Ιστός είναι ένα σύνολο από ηλεκτρονικές σελίδες (web pages) μεγάλης πολυπλοκότητας οι οποίες εισάγονται και εξάγονται από αυτό με μία διαδικασία εντελώς αποκεντρωμένη και χαοτική.

Ο καθένας μπορεί να φτιάξει μία σελίδα όπως θέλει χωρίς κάποια καθορισμένη δομή και πρότυπα περιεχομένου. Οι σελίδες του Ιστού μπορούν να έχουν γραφτεί σε διάφορες γλώσσες, διαλέκτους ή μορφές από άτομα με διαφορετικό υπόβαθρο, μόρφωση, κουλτούρα, ενδιαφέροντα και κίνητρα. Επίσης κάθε σελίδα μπορεί να διαφέρει στο μέγεθος, να περιέχει είτε αλήθειες, είτε ψέματα.

Καθημερινά στο διαδίκτυο προστίθενται περίπου ένα εκατομμύριο καινούριες σελίδες με αποτέλεσμα το μέγεθος του Παγκόσμιου Ιστού να αυξάνει διαρκώς και με ιδιαίτερα γρήγορους ρυθμούς. Κατά συνέπεια ο όγκος πληροφορίας που είναι αποθηκευμένος στο Παγκόσμιο Ιστό είναι πολύ μεγάλος και ιδιαίτερα χρήσιμος και διαρκώς αυξάνεται. Αυτό είναι και το βασικό προτέρημα του Παγκόσμιου Ιστού, όπως και οι τρομερές δυνατότητες που δίνονται από τον άμεσο τρόπο με τον οποίο προσθέτονται καινούριες σελίδες σε αυτόν.

Από την άλλη πλευρά όμως ο άναρχος τρόπος εισαγωγής των σελίδων είναι και το βασικό μειονέκτημά του. Η έλλειψη προτύπων και η παντελής έλλειψη δομής και οργάνωσης των σελίδων αυτών έχει οδηγήσει σε ένα χαοτικό σύνολο στο οποίο η πρόσβαση στη πληροφορία είναι ιδιαίτερα δυσχερής.

Σκοπός της παρούσας εργασίας είναι η βελτίωση των χαοτικών συνθηκών που επικρατούν στην ανεύρεση πληροφορίας από το περιεχόμενο του Παγκόσμιου Ιστού και η παροχή εργαλείων στο χρήστη ώστε να διευκολυνθεί η περιήγησή του στο Διαδίκτυο.

Θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή του τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών, κ. Χρήστο Μπούρα για τη πολύτιμη βοήθειά του, την υποστήριξη και τη καθοδήγησή του για την εκπόνηση της συγκεκριμένης εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον επιβλέποντα της παρούσας εργασίας κ. Βασίλη Πουλόπουλο για τις πολύτιμες συμβουλές του και τη καθοδήγησή του, χωρίς την οποία θα ήταν αδύνατη η συγγραφή της παρούσας διπλωματικής εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΛΙΣΤΑ ΕΙΚΟΝΩΝ ΚΑΙ ΠΙΝΑΚΩΝ

Κεφάλαιο

1

1 ΕΙΣΑΓΩΓΗ

Το Web 2.0 γίνεται πραγματικότητα στη ζωή μας και πλέον το διαδίκτυο τείνει να μετατραπεί από ένα εργαλείο αναζήτησης πληροφοριών σε ένα περιβάλλον παροχής υπηρεσιών. Ο όρος Web 2.0, χρησιμοποιείται για να περιγράψει τη νέα γενιά του Παγκόσμιου Ιστού η οποία βασίζεται στην όλο και μεγαλύτερη δυνατότητα των χρηστών του διαδικτύου να μοιράζονται πληροφορίες και να συνεργάζονται online. Μιλάμε για μία δυναμική διαδικτυακή πλατφόρμα στην οποία μπορούν να αλληλεπιδρούν χρήστες χωρίς εξειδικευμένες γνώσεις σε θέματα υπολογιστών και δικτύων. Η παρούσα διπλωματική εργασία αποτελεί μέρος της συνεχώς αυξανόμενης ανάπτυξης εφαρμογών οι οποίες έχουν στόχο την προσωποποίηση στον εκάστοτε χρήστη και την κάλυψη εξειδικευμένων αναγκών του. Τέτοιου είδους τεχνολογίες γνωρίζουν τρομερή άνθιση τα τελευταία χρόνια και πολλές ερευνητικές δραστηριότητες τείνουν να προσεγγίσουν από κάθε πλευρά, τεχνολογική, στατιστική, καθαρά μαθηματική ή και κοινωνική, την ολοένα αυξανόμενη και συνάμα παράξενη κοινότητα.

Στην «κοινωνία» αυτή του Διαδικτύου, οι χρήστες είναι πλέον ενεργά μέλη. Το αρχικό «μοντέλο» της εύρεσης κάθε είδους πληροφορίας στον παγκόσμιο ιστό έχει ξεπεραστεί προ πολλού. Η σημερινή μορφή του Παγκόσμιου Ιστού Πληροφοριών (Web) χαρακτηρίζεται σαν ένα περιβάλλον αχανές, ετερογενές, κατανεμημένο και πολύπλοκο με αποτέλεσμα να είναι δύσκολος ο αποδοτικός χειρισμός των δεδομένων των e-εφαρμογών με βάση παραδοσιακές μεθόδους και τεχνικές. Αυτό με τη σειρά του οδηγεί στην απαίτηση για σχεδιασμό, ανάπτυξη και υιοθέτηση «ευφυών» εργαλείων που θα επιλέξουν και θα εμφανίσουν στο χρήστη την κατάλληλη πληροφορία, στον κατάλληλο χρόνο και με την κατάλληλη μορφή. Η παρούσα διπλωματική εργασία ασχολείται με την ανάκτηση πληροφορίας από το διαδίκτυο, την επεξεργασία της και τη διαδικασία κατηγοριοποίησής της.

1.1 Προσδιορισμός του προβλήματος

Οι συνεχώς αυξανόμενοι χρήστες του διαδικτύου απαιτούν πλέον εξειδικευμένες υπηρεσίες και δεν αρκούνται απλώς σε μια περιήγηση στο διαδίκτυο αναζητώντας τις πληροφορίες που απαιτούν. Είναι γεγονός πως ένας απλός χρήστης αδυνατεί να εντοπίσει τις πληροφορίες που αναζητεί μέσα στο χαοτικό και αρκετές φορές ασαφές περιβάλλον του διαδικτύου με ταχύτητα και αξιοπιστία. Οι χρήστες γνωρίζοντας την ύπαρξη τεράστιου όγκου πληροφορίας επιθυμούν να μην αναλώνονται προσπαθώντας να κατανοήσουν το είδος της πληροφορίας που παρέχουν οι εκατομμύρια ιστότοποι του διαδικτύου, αλλά να είναι σε θέση να εντοπίζουν γρήγορα το είδος της πληροφορίας που αναζητούν.

Συνεπώς βασικό πρόβλημα για το μέσο χρήστη είναι πλέον πού μπορεί να εντοπίζει γρήγορα και αξιόπιστα συγκεκριμένο είδος πληροφορίας και όχι πώς. Τα τελευταία χρόνια παρατηρούμε πως η επεξεργασία και κατηγοριοποίηση πληροφορίας του διαδικτύου έχει κεντρίσει το ενδιαφέρον της επιστημονικής κοινότητας καθώς έχουν αναπτυχθεί αρκετοί μηχανισμοί που αναλαμβάνουν αυτή την εργασία.

1.2 Περιγραφή της εργασίας

Σύμφωνα με τα όσα αναφέραμε παραπάνω στόχος μας είναι να αναπτύξουμε ένα σύστημα που θα έχει ως σκοπό την ανάκτηση πληροφορίας από τους δικτυακούς τόπους, την κατάλληλη επεξεργασία αυτής και τέλος τη κατηγοριοποίηση των δικτυακών τόπων βάση του περιεχομένου τους. Το σύστημα που θα αναπτύξουμε θα αποτελείται από τρεις μηχανισμούς, καθένας από τους οποίους θα αναλαμβάνει την υλοποίηση των τριών εργασιών που μόλις αναφέραμε. Μεγαλύτερο βάρος θα δοθεί στο μηχανισμό επεξεργασίας της πληροφορίας, ο οποίος θα επεξεργάζεται σε διακριτά στάδια τη πληροφορία που παρέχεται σε ιστοσελίδες εκμεταλλευόμενο τη βασική δομή τους, με σκοπό το «φιλτράρισμά» τους ώστε να λάβουμε μόνο τη χρήσιμη και ωφέλιμη πληροφορία και στη συνέχεια με χρήση κατάλληλου μηχανισμού θα επιτυγχάνουμε τη κατηγοριοποίηση της ιστοσελίδας βάση της διαθέσιμης πληροφορίας.

Περιγράφοντας συνοπτικά τα τρία βασικά υποσυστήματα, θα θέλαμε να αναφέρουμε πως το πρώτο υποσύστημα που εκτελεί την ανάκτηση του κώδικα HTML από το διαδίκτυο δεν είναι ένας εξειδικευμένος crawler αλλά πρόκειται για ένα τετριμμένο υποσύστημα που αναπτύξαμε και αναλαμβάνει αυτή τη διαδικασία. Το μεγαλύτερο μέρος της εργασίας αναφέρεται στο δεύτερο υποσύστημα, που αναλαμβάνει την επεξεργασία κώδικα HTML και αποτελείται από τρία διακριτά στάδια, για την υλοποίηση των οποίων κάνουμε χρήση ποικίλων τεχνικών και τεχνολογιών. Για το τρίτο βασικό υποσύστημα θα χρησιμοποιήσουμε το μηχανισμό κατηγοριοποίησης που χρησιμοποιείται στο δικτυακό τόπο personal και αναπτύχθηκε από το κ. Βασίλη Πουλόπουλο.

1.3 Δομή της εργασίας

Η διαδικασία σχεδιασμού ενός υπολογιστικού συστήματος πρέπει να περιέχει μια συγκεκριμένη δομή, ούτως ώστε να τεκμηριώνει επαρκώς το περιεχόμενό του και να ανταποκρίνεται στις απαιτήσεις μιας τέτοιας εργασίας. Η δομή στην οποία βασίζεται η εργασία αυτή, παρουσιάζεται με τη μορφή κεφαλαίων. Στη παράγραφο αυτή, παρουσιάζουμε το περιεχόμενο κάθε κεφαλαίου.

Στο δεύτερο κεφάλαιο παρουσιάζονται γενικά στοιχεία που έχουν να κάνουν με ανάκτηση και διαχείριση πληροφορίας, με τη τεχνολογία δόμησης της πληροφορίας που προσφέρεται σε ιστοσελίδες και πιο συγκεκριμένα με τη γλώσσα HTML καθώς επίσης και με αλγορίθμους κατηγοριοποίησης.

Στο τρίτο κεφάλαιο παρουσιάζονται κάποιες σχετικές εργασίες πάνω σε πολλές από τις οποίες στηριχτήκαμε προκειμένου να κατασκευάσουμε το μηχανισμό ανάκτησης πληροφορίας και το μηχανισμό επεξεργασίας.

Στο τέταρτο κεφάλαιο περιγράφονται στοιχεία που αφορούν την αρχιτεκτονική, τις προδιαγραφές και τη λειτουργικότητα του συστήματος που

κατασκευάσαμε. Προσδιορίζονται οι βασικές αρχές πάνω στις οποίες θα στηριχτεί η λειτουργία του συστήματος.

Στο πέμπτο κεφάλαιο αναφέρονται οι τεχνολογίες που μπορούν να χρησιμοποιηθούν προκειμένου να κατασκευαστεί συνολικά το σύστημα αλλά και αυτές που επελέγησαν τελικά προκειμένου να δημιουργηθεί ένα σταθερό, ισχυρό και ευέλικτο σύστημα.

Στο έκτο κεφάλαιο αναπτύσσουμε τους βασικούς μηχανισμούς του συστήματος που χρησιμοποιούμε για την επίτευξη καλύτερης και ορθότερης λειτουργίας του συστήματος.

Στο έβδομο κεφάλαιο αναφέρονται λεπτομέρειες που αφορούν την υλοποίηση και πιο συγκεκριμένα παρουσιάζονται η δομή της βάσης δεδομένων του συστήματος και οι κυριότεροι αλγόριθμοι που υλοποιήθηκαν.

Στο όγδοο κεφάλαιο γίνεται ανασκόπηση της εργασίας και προτάσεις για μελλοντικές εργασίες και βελτιώσεις του συστήματος.

Τέλος στο παράρτημα αυτής της εργασίας παρατίθενται τα σημαντικότερα κομμάτια κώδικα που υλοποιήσαμε.

Κεφάλαιο

2

2 ΤΑ ΘΕΜΑΤΑ ΠΟΥ ΜΑΣ ΑΠΑΣΧΟΛΟΥΝ

Στο παρόν κεφάλαιο θα παρουσιάσουμε τα θέματα με τα οποία θα ασχοληθεί η συγκεκριμένη εργασία καθώς επίσης και μία μικρή ανάλυση καθενός από αυτά προκειμένου να δημιουργηθεί το κατάλληλο υπόβαθρο για να είναι εφικτή η κατανόηση των όρων που θα χρησιμοποιηθούν στα επόμενα κεφάλαια.

Συγκεκριμένα τα θέματα που μας απασχολούν χωρίζονται σε τρεις ενότητες βάση των μηχανισμών που αναπτύξαμε για την υλοποίηση του συστήματός μας. Η πρώτη ενότητα αναφέρεται στην ανάκτηση πληροφορίας από το διαδίκτυο, η δεύτερη στην επεξεργασία της διαθέσιμης πληροφορίας και η τρίτη στη διαδικασία κατηγοριοποίησής της.

Για να γίνει σαφές για την υλοποίηση του μηχανισμού αυτόματης κατηγοριοποίησης δικτυακών τόπων αρκεί η εκτέλεση των παρακάτω βημάτων:

- Ανάκτηση σελίδων από το διαδίκτυο
- Διαχείριση του διαθέσιμου κώδικα κάθε σελίδας
- Επεξεργασία σε διακριτά στάδια του κώδικα, προκειμένου να εξαχθεί ωφέλιμη πληροφορία
- Αποθήκευση και διαχείριση του κειμένου που αποτελεί την ωφέλιμη πληροφορία
- Κατηγοριοποίηση της πληροφορίας βασισμένη σε συγκεκριμένες κατηγορίες που αντιπροσωπεύουν το σύνολο της πληροφορίας

Στη συνέχεια θα προσπαθήσουμε να εισάγουμε βασικές έννοιες που θεωρούμε τη γνώση τους απαραίτητη για τη κατανόηση και περιγραφή των μηχανισμών που αναπτύχθηκαν για την ολοκλήρωση της παρούσας εργασίας.

2.1 Εξόρυξη πληροφορίας από το Διαδίκτυο

Εξόρυξη πληροφορίας από το Διαδίκτυο ονομάζεται κάθε διαδικασία που έχει σαν αποτέλεσμα ανάκτηση πληροφορίας (Information Retrieval) από τον παγκόσμιο ιστό. Στο εξής θα αναφερόμαστε στον όρο ανάκτηση πληροφορίας ως IR για συντομία. Η ανακτώμενη πληροφορία δεν περιορίζεται απλώς σε σελίδες HTML, αλλά μπορεί να είναι και αρχεία πολυμέσων ή οποιοδήποτε είδος αρχείου μπορεί να μεταφερθεί πάνω από το Διαδίκτυο. Η ανάγκη για ανάκτηση πληροφορίας πηγάζει από τις αρχές της δεκαετίας του 50 όταν ο Mooers [1] εξέφρασε ανοιχτά σε δημοσίευσή του την ανάγκη για ανάκτηση πληροφορίας. Αργότερα, στη δεκαετία του 60, το IR είχε γίνει πλέον ένα πολύ δημοφιλές θέμα καθώς πολλοί ερευνητές

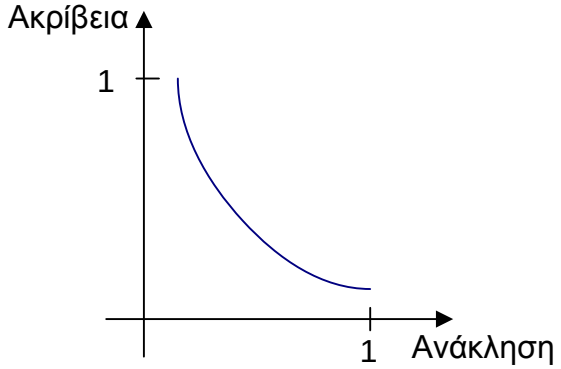
πίστευαν ότι μπορούν να αυτοματοποιήσουν μέχρι τότε χειροκίνητες διαδικασίες όπως η δεικτοδότηση και η αναζήτηση [2; 3].

Προκειμένου να πετύχει το στόχο της η κοινότητα IR όρισε δύο βασικές ενέργειες που έχουν γίνει αντικείμενα έρευνας για πολλά χρόνια και είναι: η δεικτοδότηση και η αναζήτηση. Η δεικτοδότηση αναφέρεται στον τρόπο με τον οποίο αναπαρίσταται η πληροφορία για τους σκοπούς της ανάκτησης. Η αναζήτηση αναφέρεται στον τρόπο με τον οποίο δομείται η πληροφορία όταν πραγματοποιείται ένα ερώτημα. Παρόλο που οι δύο αυτές διαδικασίες αποτελούν τον πυρήνα ενός συστήματος IR, άλλες διαδικασίες είναι αυτές που κερδίζουν έδαφος, όπως τεχνικές αναπαράστασης της πληροφορίας, με σκοπό να βελτιωθεί η αποτελεσματικότητα της ανάκτησης [4].

Στην παρούσα φάση το IR αντιμετωπίζει μία σειρά από θέματα. Αρχικά, εφαρμόστηκε σε ΒΔ βιβλιοθηκών, όπου σε ένα αρχείο αποθηκεύονταν γενικά χαρακτηριστικά κάθε εγγράφου, όπως ο τίτλος και ο συγγραφέας, και η αναζήτηση γινόταν βάσει αυτών των στοιχείων. Στη συνέχεια, και εξ αιτίας της αύξησης του μεγέθους των αποθηκευτικών μέσων, ολόκληρο το κείμενο αποθηκευόταν σε αρχείο και η αναζήτηση ήταν εφικτή σε ολόκληρες συλλογές από κείμενα. Έτσι μέχρι ενός σημείου το IR αντιπροσώπευε την ανάκτηση κειμένων. Αργότερα και έως σήμερα, δίνεται περισσότερη σημασία στον όρο πληροφορία (Information). Άλλωστε σήμερα δεν έχουμε μόνο έγγραφα πάνω στα οποία γίνεται η αναζήτηση αλλά και αρχεία πολυμέσων. Ωστόσο το βασικό κλειδί στην υπόθεση του IR είναι ανάκτηση κειμένων ή πληροφορίας που προσεγγίζουν περισσότερο τις ανάγκες του χρήστη που πραγματοποιεί την αναζήτηση.

Ένα από τα βασικά στοιχεία του IR είναι η μέτρηση του κατά πόσο τα ανακτημένα κείμενα είναι σχετικά με το ερώτημα που κάνουμε. [5]. Έτσι λοιπόν, ένα βασικό στοιχείο στο οποίο εστιάζουμε είναι η εύρεση μετρικών που θα μπορούν να αναπαραστήσουν αριθμητικά τη σχετικότητα των αποτελεσμάτων ενός συστήματος IR. Πολλές μετρικές έχουν αναπτυχθεί με τις δύο πιο γνωστές να είναι η ανάκληση και η ακρίβεια. Η ακρίβεια μας δίνει το ποσοστό (%) των σχετικών κειμένων εν συγκρίσει με αυτά που ανακτήθηκαν ενώ η ανάκληση μας δίνει το ποσοστό (%) των κειμένων που ανακτήθηκαν εν συγκρίσει με μία συλλογή που γνωρίζουμε ότι περιέχει όλα τα σχετικά.

Η συνηθισμένη απόκριση που έχει ένα σύστημα IR είναι αυτή που φαίνεται στο παρακάτω σχήμα στο οποίο φαίνεται ότι τα μεγέθη ακρίβεια και ανάκληση είναι αντιστρόφως ανάλογα. Αυτό σημαίνει πως για αν αυξήσουμε την ανάκληση θα μειωθεί η ακρίβεια. Φυσικά ισχύει και το αντίστροφο [6].



Εικόνα 1. Σχεδιάγραμμα ακρίβειας – ανάκλησης

Ένα σύστημα IR μπορεί να πετύχει κατά μέσο όρο περίπου 30% ανάκληση και 30% ακρίβεια. Οι τιμές αυτές δεν έχουν καμία σύγκριση με ένα σύστημα DBMS που τα ποσοστά αυτού προσεγγίζουν το 100%. Ωστόσο θα μπορούσε κανείς να πει πως και τα δύο συστήματα πραγματοποιούν την ίδια διαδικασία, δηλαδή ανάκτηση πληροφορίας. Αυτό βέβαια έχει να κάνει με τον τρόπο με τον οποίο δομείται ένα σύστημα DBMS και ο οποίος είναι τέτοιος ώστε να εξυπηρετεί απόλυτα τις ανάγκες ενός χρήστη.

Αυτή η δυσκολία που αντιμετωπίζουν τα συστήματα IR (μικρές τιμές ανάκλησης και ακρίβειας) γεννούν ένα άλλο επιστημονικό πεδίο το οποίο υπάρχει παράλληλα με το IR και είναι το IF (Information Filtering). Σε ένα κλασσικό άρθρο οι Belkin και Croft παρουσίασαν δύο διαφορετικούς ορισμούς για τα δύο παραπάνω θέματα οι οποίοι έχουν κοινές τεχνικές αλλά διαφέρουν σε τρία βασικά στοιχεία [7]. Πρώτον, στο IR όταν ο χρήστης κάνει ένα ερώτημα περιμένει άμεση απόκριση. Στο IF ο χρήστης μπορεί να περιμένει, εν γνώσει του, για μεγάλο χρονικό διάστημα μέχρι να του παρουσιαστεί μία απάντηση. Επιπρόσθετο το IF χειρίζεται και θέματα που από τη φύση του είναι δυναμικά και εντάσσει στο μηχανισμού του στοιχεία εκμάθησης σύμφωνα με τα κείμενα που προσθέτει στη συλλογή του. Τέλος, το βασικότερο είναι πως το IR αναζητά παραπλήσια κείμενα από μία μεγάλη συλλογή κειμένων σε αντίθεση με το IF το οποίο προσπαθεί να αφαιρέσει από μία συλλογή τα εισερχόμενα κείμενα που δεν είναι σχετικά.

Παρ' όλες τις διαφορές που έχουν τα δύο αυτά πεδία δεν πρέπει να αμελούμε πως έχουν παραπλήσιο σκοπό: να εξασφαλίσουν ότι τα κείμενα που θα παρουσιαστούν στο χρήστη είναι σχετικά με το ερώτημά του.

Τα διαγράμματα ακρίβειας/ανάκλησης είναι χρήσιμα εφόσον μελετούμε την απόδοση ανάκτησης διαφορετικών αλγορίθμων σε ένα σύνολο από πρότυπες πληροφοριακές ανάγκες. Ωστόσο υπάρχουν περιπτώσεις στις οποίες θα θέλαμε να συγκρίνουμε την απόδοση αλγορίθμων ανάκτησης για ατομικές πληροφοριακές ανάγκες. Οι λόγοι για να το κάνουμε αυτό είναι δύο:

1. η χρήση μέσων τιμών που προκύπτουν από την εκτέλεση διαφόρων ερωτημάτων μπορεί να αποκρύπτει σημαντικές ανωμαλίες στον αλγόριθμο ανάκτησης
2. όταν συγκρίνουμε δύο αλγορίθμους μπορεί να θέλουμε να μελετήσουμε κατά πόσο ο ένας είναι καλύτερος του άλλου για κάθε μία από τις πληροφοριακές ανάγκες που έχουμε και όχι συνολικά.

Σε τέτοιες περιπτώσεις υπολογίζουμε μία μόνο τιμή ακρίβειας για κάθε ερώτημα, η οποία θα μπορούσε να θεωρηθεί σαν σύνοψη του συνολικού διαγράμματος ακρίβειας/ανάκλησης. Συνήθως αυτή η τιμή είναι η ακρίβεια σε κάποιο συγκεκριμένο επίπεδο ανάκλησης. Φυσικά αυτές είναι λίγες από τις πολλές προσεγγίσεις που μπορούν να γίνουν.

2.1.1 Μοντέλα ανάκτησης πληροφορίας

Τα τρία κλασσικά μοντέλα στην Ανάκτηση Πληροφορίας είναι το Boolean, το Vector Space και το Πιθανοτικό. Στο μοντέλο Boolean, τόσο τα κείμενα όσο και τα ερωτήματα αντιμετωπίζονται ως ένα σύνολο από όρους δεικτοδότησης. Κατά συνέπεια το μοντέλο μπορεί να θεωρηθεί ως συνολοθεωρητικό. Στο Vector Space, τα κείμενα και τα ερωτήματα αναπαρίστανται ως διανύσματα σε έναν t-διάστατο χώρο. Έτσι λέμε ότι το μοντέλο είναι αλγεβρικό. Το Πιθανοτικό μοντέλο εισάγει έναν τρόπο αναπαράστασης, ο οποίος βασίζεται στην πιθανοθεωρία. Κατά συνέπεια το μοντέλο είναι πιθανοτικού χαρακτήρα. Το πιθανοτικό μοντέλο και Με τον καιρό προτάθηκαν διάφορες νέες προσεγγίσεις σε καθεμιά από τις κατηγορίες βασικών μοντέλων. Έτσι

έχουμε στο συνολοθεωρητικό πεδίο τα μοντέλα, ασαφές (fuzzy) Boolean και επεκταμένο Boolean. Στα αλγεβρικά μοντέλα έχουμε το γενικευμένο vector space, την λανθάνουσας σημασιολογικής δεικτοδότησης (LSI) και το μοντέλο των νευρωνικών δικτύων. Στον πιθανοτικό τομέα εμφανίστηκαν τα δίκτυα εξαγωγής συμπεράσματος (inference networks) και τα δίκτυα πεποίθησης (belief networks). Εκτός από την χρήση του περιεχομένου των κειμένων, ορισμένα μοντέλα εκμεταλλεύονται και την εσωτερική δομή που φυσιολογικά υπάρχει στο γραπτό λόγο.

Σε αυτή την περίπτωση λέμε ότι έχουμε ένα δομημένο μοντέλο. Για τη δομημένη ανάκτηση κειμένου, συναντούμε δύο μοντέλα, τις μη επικαλυπτόμενες λίστες (non-overlapping lists) και τους κοντινούς κόμβους (proximal nodes).

2.1.1.1 Τυπικός ορισμός των μοντέλων

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

1) D είναι ένα σύνολο από λογικές αναπαραστάσεις για τα κείμενα της συλλογής

2) Q είναι ένα σύνολο από λογικές αναπαραστάσεις για τις πληροφοριακές ανάγκες του χρήστη. Αυτές οι αναπαραστάσεις καλούνται ερωτήματα

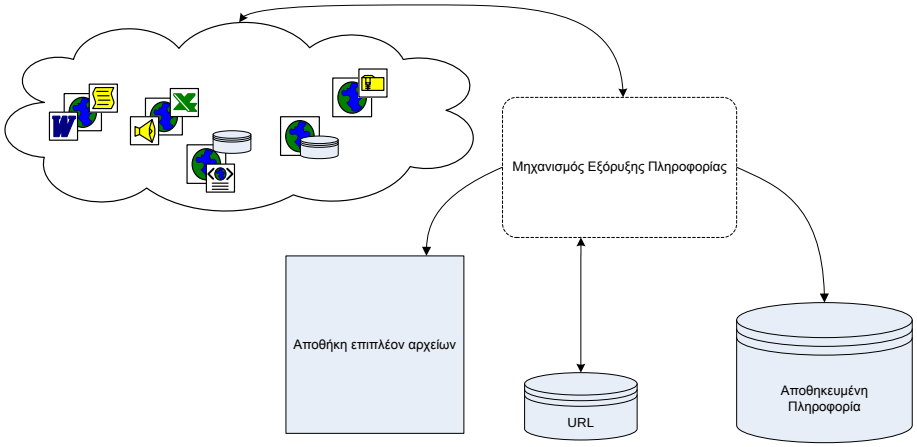
3) F είναι ένα υπόβαθρο για την μοντελοποίηση της αναπαράστασης των κειμένων, των ερωτημάτων και των σχέσεων μεταξύ τους

4) R(qi, dj) είναι μια συνάρτηση κατάταξης, η οποία συνδέει έναν πραγματικό αριθμό με ένα ερώτημα qi ∈ Q και μια αναπαράσταση κειμένου dj ∈ D. Μια τέτοια κατάταξη ορίζει μια διάταξη πάνω στα κείμενα πάντα με βάση το ερώτημα. qi.

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

2.1.2 Αρχιτεκτονική μηχανισμών εξόρυξης

Όλες οι μηχανές αναζήτησης πραγματοποιούν ανάκτηση πληροφορίας προκειμένου να μπορούν να εξυπηρετούν τους χρήστες τους. Έτσι, μέχρι σήμερα έχει κατασκευαστεί πληθώρα προγραμμάτων τα οποία είτε λειτουργώντας σαν αυτόνομες μονάδες είτε σε συνεργασία μεταξύ τους πραγματοποιούν εξόρυξη πληροφορίας. Η γενική ιδέα ενός μηχανισμού εξόρυξης πληροφορίας είναι εξαιρετικά απλή και φαίνεται στο παρακάτω σχήμα.



Εικόνα 2. Μηχανισμός Εξόρυξης Πληροφορίας

Ένας τέτοιος μηχανισμός μπορεί να είναι ένας απλός υπολογιστής ή ακόμα και μερικές χιλιάδες υπολογιστές που λειτουργούν κάτω από την επίβλεψη ενός. Ο μηχανισμός ξεκινά να λειτουργεί περιδιαβαίνοντας σελίδες του Διαδικτύου. Οι HTML σελίδες αποθηκεύονται σε μία βάση δεδομένων μαζί με επιπρόσθετες πληροφορίες για αυτές οι οποίες μπορεί να περιλαμβάνουν: το URL, την ώρα που ανακτήθηκε η σελίδα, το μέγεθός της και άλλα. Σε μία ξεχωριστή (συνήθως) βάση δεδομένων αποθηκεύονται όλα τα URL που έχουν ανακτηθεί και τα οποία ανακτώνται ανά τακτά χρονικά διαστήματα. Παράλληλα κάθε σελίδα αναλύεται προκειμένου να εξαχθούν από αυτή όλα τα links που περιέχει (σύμβολο <a> στην HTML). Τα links που «διαβάζει» ο μηχανισμός συγκρίνονται με αυτά που υπάρχουν αποθηκευμένα στη βάση δεδομένων URL και γίνονται οι κατάλληλες προσθήκες. Τέλος, κάποια επιπλέον αρχεία (doc, css, xml, scripts, πολυμέσα) αποθηκεύονται συνήθως σε καταλόγους που ονομάζονται κατάλληλα από τον μηχανισμό, έτσι ώστε να είναι σε θέση να τα προσπελάσει ανά πάσα στιγμή.

Μερικοί από τους πιο γνωστούς μηχανισμούς που πραγματοποιούν εξόρυξη πληροφορίας είναι οι crawlers, τα bots, τα spiders κ.α. Η λειτουργία τους είναι ουσιαστικά ίδια και βασίζεται στην αρχιτεκτονική που φαίνεται στο παραπάνω σχήμα.

2.2 Επεξεργασία Πληροφορίας

Προτού αναπτύξουμε τη περιγραφή του μηχανισμού επεξεργασίας πληροφορίας που κατασκευάσαμε θεωρούμε πως είναι απαραίτητο να αναφέρουμε το τυπικό ορισμό της πληροφορίας και στη συνέχεια να περιγράψουμε τη δομή μίας ιστοσελίδας στην οποία περιέχεται πληροφορία και πως αυτή δημιουργείται, καθώς αυτό αφορά το πιο βασικό κομμάτι της εργασίας μας.

2.2.1 Τι είναι Πληροφορία

Η πληροφορία, έχει να κάνει με τα στοιχεία εκείνα που μεταδίδονται από μια πηγή προς κάποιον δέκτη. Έτσι, ανάλογα με τη σκοπιά κάτω από την οποία

προσεγγίζει κανείς την πληροφορία, αυτή μπορεί να έχει να κάνει με την ενημέρωση αν η σκοπιά είναι επικοινωνιακή, για τη διάκριση μεταξύ σημαίνοντος και σημαινόμενου αν την εξετάσουμε υπό το πρίσμα της γλωσσολογίας ή ακόμα και για ένα τηλεφωνικό ειδοποιητήριο όπως συχνά συμβαίνει στην καθημερινή ζωή. Ουσιαστικά, προσεγγίζοντας την έννοια της πληροφορίας διαισθητικά, διαπιστώνουμε πως γενικά παραπέμπει σε καινούργια γνώση για κάτι. Επιπλέον, αν και η «σημαίνουσα» αξία της πληροφορίας υφίσταται αναλλοίωτα, το «σημαινόμενο» αυτής είναι δυνατόν να μεταβάλλεται κάθε φορά με τρόπο καθοριστικό για την ίδια την πληροφορία. Ο τρόπος που το «σημαινόμενο» μεταβάλλεται καθορίζεται από τους τρεις παράγοντες που αφενός στοιχειοθετούν και αφετέρου δίνουν υπόσταση στην πληροφορία: την πηγή, το δέκτη και το μέσο διάδοσης. Δηλαδή, αν μια πληροφορία υφίσταται κατ' αρχήν έχει νόημα μόνο αν την γνωρίσει κάποιος, επιπλέον, ενώ η πληροφορία υφίσταται (σημαίνον) το νόημά της (ή περιεχόμενό της ή σημαινόμενο) εξαρτάται από το πώς θα το μεταδώσει η πηγή, από το πόσο καλά (ή αξιόπιστα) θα το μεταφέρει το μέσο προς το δέκτη και τέλος από το πώς θα το αντιληφθεί ο δέκτης. Όπως γίνεται αντιληπτό σκοπός του μηχανισμού επεξεργασίας που αναπτύξαμε είναι να επιτύχουμε να αντλήσουμε από τη δομή μίας ιστοσελίδας μόνο την ωφέλιμη πληροφορία γαι το χρήστη.

2.2.2 Η γλώσσα HTML

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η html δεν είναι μια γλώσσα προγραμματισμού. Είναι μια περιγραφική γλώσσα (*markup language*), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Ο φυλλομετρητής (browser) αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Αξίζει να σημειωθεί ότι η html είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Η html χρησιμοποιεί τις ειδικές ετικέτες (τα tags) για να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων < και >. Π.χ. <BODY> Οι οδηγίες είναι case insensitive, δηλαδή δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

2.2.2.1 Η δομή μίας ιστοσελίδας με τη γλώσσα HTML

Βασικό στοιχείο της γλώσσας αυτής είναι η έννοια του "tag" (ετικέτα). Κάθε στοιχείο των σελίδων HTML εμφανίζεται ανάμεσα σε ετικέτες και οι ετικέτες αυτές καθορίζουν τη τοποθεσία μέσα στη σελίδα των στοιχείων αυτών και τη μορφή με την οποία θα εμφανίζονται και θα φαίνονται.

Όλες οι σελίδες HTML ξεκινούν με την ετικέτα <html> και τελειώνουν με την αντίστοιχη ετικέτα τέλους </html>. Επίσης κάθε σελίδα HTML αποτελείται από δύο τμήματα. Το πρώτο τμήμα καθορίζεται από τις ετικέτες <head> και </head> και το δεύτερο από τις <body> και </body>.

Το πρώτο τμήμα αποτελεί και τη κεφαλή του κειμένου και καθορίζει διάφορες παραμέτρους της συγκεκριμένης σελίδας. Για παράδειγμα καθορίζει το τίτλο της, τη σχέση της με άλλες σελίδες, τη μορφή που μπορεί να έχει ή ακόμα και τη scripting

γλώσσα που μπορεί να χρησιμοποιεί. Από τα παραπάνω στοιχεία αυτό που χαρακτηρίζει κατά κάποιο τρόπο τη συγκεκριμένη σελίδα είναι ο τίτλος της ο οποίος εμφανίζεται ανάμεσα στις ετικέτες <title> και </title>.

Το δεύτερο τμήμα αποτελεί το σώμα της σελίδας και είναι αυτό που περιέχει όλη τη πληροφορία που επιθυμεί ο δημιουργός της ιστοσελίδας να παρουσιάσει, είτε με τη μορφή κειμένου, είτε εικόνων, είτε ακόμα και με διασυνδέσεις προς άλλες σελίδες δικές του ή άλλες ατόμων. Από τα στοιχεία που μπορούν να τοποθετηθούν στο σώμα μίας σελίδας HTML τα περισσότερα μπορούν να ενταχθούν σε δύο κατηγορίες.

Η πρώτη αποτελείται από τα στοιχεία ορισμού περιοχής τα οποία είναι οι επικεφαλίδες, οι παράγραφοι και οι οριζόντιες γραμμές. Σημαντικό ενδιαφέρον παρουσιάζουν οι επικεφαλίδες, οι οποίες καθώς τονίζονται από το δημιουργό της σελίδας υποδηλώνουν ότι περιέχουν κάποια πληροφορία η οποία πρέπει να προσεχτεί. Υπάρχουν έξι επίπεδα από επικεφαλίδες, το H1 είναι το πιο σημαντικό και το H6 το λιγότερο σημαντικό. Το κείμενο που εμφανίζεται σαν επικεφαλίδα περιέχεται ανάμεσα σε ετικέτες της μορφής <h1> και </h1>, δηλαδή ανάλογα με την επικεφαλίδα καθορίζεται και η ετικέτα.

Η δεύτερη βασική κατηγορία αποτελείται από τα στοιχεία ορισμού κειμένου τα οποία ορίζουν τύπους χαρακτήρων στο κείμενο. Βασική υποκατηγορία αυτών των στοιχείων αποτελούν τα στοιχεία τύπου γραμματοσειράς. Ανάλογα με τα στοιχεία που επιλέγει ο δημιουργός της σελίδας μπορεί να παρουσιάσει κάποιο κείμενο, είτε με πιο έντονα γράμματα, είτε με πλάγιους χαρακτήρες, είτε να είναι υπογραμμισμένο. Ένα κείμενο για να εμφανίζεται με έντονα γράμματα πρέπει να βρίσκεται είτε ανάμεσα στις ετικέτες και , είτε ανάμεσα στις και . Για να εμφανίζεται με πλάγιους χαρακτήρες πρέπει να βρίσκεται ανάμεσα στις ετικέτες και . Τέλος για να είναι υπογραμμισμένο πρέπει να βρίσκεται ανάμεσα στις ετικέτες <u> και </u>.

Όπως αναφέρθηκε και παραπάνω ο δημιουργός της σελίδας μπορεί να εμφανίζει στη σελίδα του διάφορες εικόνες. Για να το επιτύχει αυτό χρειάζεται ένα άλλο στοιχείο της γλώσσας HTML, το img, το οποίο ανήκει στα στοιχεία ορισμού κειμένου. Το στοιχείο αυτό καθορίζεται από την ετικέτα <img> και δεν έχει ετικέτα τέλους. Για να εμφανιστεί μία εικόνα σε μία σελίδα HTML πρέπει να υπάρχει στο κώδικα της σελίδας το στοιχείο img με τη παρακάτω μορφή: <img src=”imagepath/imagename.jpeg” alt=”comment”>. Στο πεδίο «src» του στοιχείου img δίνεται πρώτα το μονοπάτι που δείχνει το κατάλογο στον οποίο είναι αποθηκευμένη η εικόνα που θα εμφανιστεί και μετά δίνεται το όνομα της εικόνας με τη κατάληξη του τύπου της. Στο πεδίο «alt» δίνεται το κείμενο που επιθυμεί ο δημιουργός να φαίνεται μέχρι να εμφανιστεί η εικόνα, είναι εμφανές ότι το κείμενο αυτό περιγράφει κατά κάποιο τρόπο την εικόνα και το θέμα της.

Τέλος ένα ακόμα πολύ χρήσιμο και θεμελιώδες στοιχείο της γλώσσας HTML είναι οι υπερδεσμοί. Οι υπερδεσμοί περιέχονται ανάμεσα στις ετικέτες <a> και και έχουν την ακόλουθη μορφή: hyperlink – text. Στο πεδίο «href» του στοιχείου a δίνεται η διεύθυνση της σελίδας προς την οποία δείχνει ο συγκεκριμένος υπερδεσμός. Ανάμεσα στις δύο ετικέτες δίνεται το κείμενο που παρουσιάζει ο υπερδεσμός και η σελίδα προς την οποία δείχνει, επομένως κατά κάποιο τρόπο παρουσιάζει και το περιεχόμενο της σελίδας αυτής.[14]

Ένα απλοϊκό παράδειγμα μίας πλήρους σύνταξης σε γλώσσα HTML ακολουθεί παρακάτω:

```
<html>
<head>
<title>Ένα απλό παράδειγμα</title>
</head>
<body>
<h1>Η HTML </h1>
<p> Η πρώτη παράγραφος ενός κώδικα HTML </p>
<p> Και η δεύτερη παράγραφος.</p>
</body>
</html>
```

Αναλυτικά οι εντολές που χρησιμοποιήθηκαν στη σύνταξη του παραπάνω κώδικα είναι οι παρακάτω:

- **<head>** Η ετικέτα αυτή περιέχει τον τίτλο και άλλες σημαντικές πληροφορίες για το έγγραφο.
- **<title>** Μέσα σε αυτήν την ετικέτα περικλείεται ο τίτλος του εγγράφου. Ο τίτλος εμφανίζεται στην μπάρα του browser
- **<p>** Η ετικέτα αυτή ορίζει μία νέα παράγραφο.
- **<h1>** Η ετικέτα αυτή καθορίζει το μέγεθος των γραμμάτων. Ο αριθμός μπορεί να ανέλθει ως το 6 (δηλαδή <h2> , <h3>). Το <h1> είναι το μεγαλύτερο μέγεθος και το <h6> είναι το μικρότερο.

2.2.3 Κανονικές Εκφράσεις (Regular Expressions)

Θεωρούμε απαραίτητο σημείο την αναφορά στις Κανονικές Εκφράσεις, καθώς η χρήση τους αποτέλεσε ένα πολύ σημαντικό παράγοντα στην ολοκλήρωση του μηχανισμού επεξεργασίας κειμένου.

Οι ρίζες των regular expressions βρίσκονται στη θεωρία αυτομάτων και είναι μέρος της θεωρίας των πρώτων γλωσσών προγραμματισμού. Στη δεκαετία του '50, ο μαθηματικός Stephen Cole Kleene περιέγραψε αυτά τα πρότυπα χρησιμοποιώντας τη μαθηματική τυποποίηση την οποία αποκαλούσε κανονικά σύνολα (regular sets). Η γλώσσα SNOBOL ήταν μια πρόωρη εφαρμογή των regular sets, αλλά όχι ίδια με τις κανονικές εκφράσεις. Μεγάλη εξάπλωση γνώρισαν οι κανονικές εκφράσεις από τη στιγμή που ενσωματώθηκαν σε συστήματα Unix η οποία οδήγησε τελικά στη δημοφιλή χρήση εργαλείων αναζήτησης βάση κανονικών εκφράσεων (grep). Μέχρι σήμερα πολλές γλώσσες προγραμματισμού έχουν ενσωματώσει τη χρήση κανονικών εκφράσεων.

2.2.3.1 Τυπικός ορισμός της Κανονικής Έκφρασης

Η R λέγεται κανονική έκφραση εάν είναι της μορφής:

1. A, όπου α ένα σύμβολο ενός αλφαβήτου Σ
2. E
3. Ø
4. (R1 ⊔ R2), όπου R1 και R2 δύο κανονικές εκφράσεις

5. (R1 ◦ R2) , όπου R1 και R2 δύο κανονικές εκφράσεις
6. R1* , όπου R1 μία κανονική έκφραση

Στα σκέλη 1 και 2, οι κανονικές εκφράσεις α και ε αναπαριστούν τις γλώσσες {α} και {ε}, αντίστοιχα. Στο σκέλος 3, η κανονική έκφραση Ø αναπαριστά τη κενή γλώσσα. Στα σκέλη 4,5 και 6, οι εκφράσεις αναπαριστούν αντίστοιχα τις γλώσσες που προκύπτουν από την ένωση ή τη συναρμογή των R1 και R2 και από τη σώρευση της γλώσσας R1.

Οι κανονικές εκφράσεις ε και Ø δεν θα πρέπει να συγχέονται μεταξύ τους. Η γλώσσα ε αναπαριστά τη γλώσσα που περιέχει μία μόνο λέξη –τη κενή λέξη-, ενώ η Ø αναπαριστά τη γλώσσα που δε περιέχει καμία λέξη.

Χάριν απλότητας, εισάγουμε επίσης την έκφραση R^+ ως συντομογραφία της έκφρασης RR^*. Έτσι, ενώ η γλώσσα R^* περιέχει όλες τις λέξεις που προκύπτουν από τη συναρμογή 0 ή περισσότερων λέξεων της R, η R^+ περιέχει όλες τις λέξεις που προκύπτουν από τη συναρμογή 1 ή περισσότερων λέξεων της R.[15]

Ακολουθεί ένα παράδειγμα της σύνταξης των κανονικών εκφράσεων θεωρώντας ότι το αλφάβητο Σ είναι το {0,1} :

$1^*(01^+)^*$ = {w| κάθε 0 στη w ακολουθείται από τουλάχιστον ένα 1}

2.3 Κατηγοριοποίηση Πληροφορίας

Η κατηγοριοποίηση της πληροφορίας είναι ένα θέμα που απασχολεί ολοένα και περισσότερο τα τελευταία χρόνια την ακαδημαϊκή κοινότητα. Βασική αρχή στην κατηγοριοποίηση αποτελούν τα μοντέλα μάθησης. Είναι ουσιαστικά η βάση ενός μηχανισμού κατηγοριοποίησης. Ένας μηχανισμός κατηγοριοποίησης υλοποιεί μία διαδικασία μέσω της οποίας προβάλλεται το διάνυσμα του κειμένου εισόδου στο χώρο και μέσω συγκρίσεων προκύπτει η κλάση στην οποία πιθανώς ανήκει το κείμενο εισόδου. Στην περίπτωση της κατηγοριοποίησης κειμένου τα χαρακτηριστικά είναι λέξεις του κειμένου και οι κλάσεις είναι κατηγορίες κειμένου (π.χ. πολιτικά, αθλητικά, πολιτισμός κλπ). Συχνά, οι μηχανισμοί κατηγοριοποίησης είναι πιθανοτικοί όσον αφορά τη διαδικασία με την οποία κατηγοριοποιούν, η οποία είναι πιθανοτική κατανομή.

Ο κυρίαρχος στόχος της κατηγοριοποίησης πληροφορίας είναι να πραγματοποιήσει διαδικασία μάθησης στους μηχανισμούς κατηγοριοποίησης χρησιμοποιώντας επαγωγικές διαδικασίες. Προκειμένου να γίνει αντιληπτό αυτό θα αναλύσουμε στην πορεία μια σειρά από διαφορετικούς αλγορίθμους κατηγοριοποίησης. Όλοι οι αλγόριθμοι απαιτούν μόνο ένα μικρό σύνολο από «πληροφορία εκπαίδευσης» σαν είσοδο. Η «πληροφορία εκπαίδευσης» χρησιμοποιείται για να αρχικοποιήσει τις παραμέτρους του μοντέλου κατηγοριοποίησης. Στη διαδικασία δοκιμών και αποτίμησης, μπορούμε να προσδιορίσουμε την αποδοτικότητα κάθε αλγορίθμου.

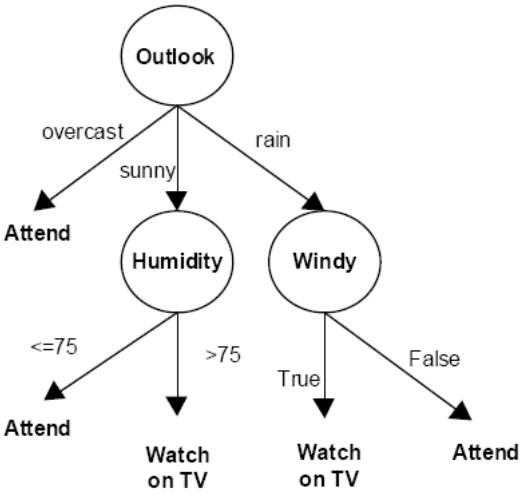
Ένα κοινό χαρακτηριστικό στις διαφορετικές εκδόσεις των αλγορίθμων είναι η αναπαράσταση των κειμένων με ένα διάνυσμα από λέξεις, το οποίο είναι δημοφιλέστατο και στα συστήματα IR. Οι τιμές της συχνότητας των λέξεων και η ανάστροφη συχνότητα κειμένων υπολογίζονται και ανάλογα με την τεχνική εκμάθησης που χρησιμοποιείται, μερικά ή όλα από τα στοιχεία εισόδου εισάγονται στην πληροφορία εκπαίδευσης.

2.3.1 Αλγόριθμοι για κατηγοριοποίηση πληροφορίας

Υπάρχει πληθώρα αλγορίθμων που πραγματοποιούν αυτόματη κατηγοριοποίηση κειμένων βασισμένοι στο περιεχόμενο του κειμένου. Παρακάτω παρουσιάζονται οι πιο σημαντικοί από αυτούς.

2.3.1.1 Δέντρα απόφασης (Decision Trees)

Σε αυτό τον αλγόριθμο αρχικά έχουμε μια σειρά εγγραφών. Κάθε εγγραφή έχει την ίδια δομή, ένα ζευγάρι με χαρακτηριστικό/τιμή. Ένα από αυτά τα ζευγάρια αντιπροσωπεύει την κατηγορία της εγγραφής. Ο στόχος είναι να προσδιοριστεί ένα δέντρο το οποίο με βάση απαντήσεις σε ερωτήματα σε ότι αφορά χαρακτηριστικά που δεν αφορούν κάποια κατηγορία να προβλεφθεί η κατηγορία στην οποία θα ενταχθεί το χαρακτηριστικό. Ένας μηχανισμός προσδιορισμού κατηγορίας μέσω δέντρου δημιουργείται για κάθε ξεχωριστή κατηγορία χρησιμοποιώντας την προσέγγιση του Quinlan [8]. Αλγόριθμοι όπως ο ID3, C4.5 ή ο C5 είναι απλώς παραδείγματα που προκύπτουν από πρότυπα δέντρα απόφασης. Συνήθως τα χαρακτηριστικά κατηγοριών έχουν δυαδικές τιμές (0 ή 1). Στο παρακάτω γράφημα βλέπουμε ένα παράδειγμα δέντρου απόφασης που δύναται να αποφασίσει αν κάποιος πρέπει να πάει να δει έναν ποδοσφαιρικό αγώνα ή να τον παρακολουθήσει από την τηλεόρασή του, βασισμένο στις καιρικές συνθήκες.



Εικόνα 3. Δέντρο απόφασης

Εκτός από δυαδικές τιμές για την κατηγοριοποίηση, μπορεί να χρησιμοποιηθεί μέθοδος που χρησιμοποιεί κλάση πιθανοτήτων όπου η έξοδος είναι η πιθανότητα να ανήκει ένα αντικείμενο σε μια συγκεκριμένη κατηγορία. Ένα πιο αναλυτικό άρθρο για τη συγκεκριμένη τεχνική μπορεί να βρεθεί στο [9].

2.3.1.2 Naïve Bayes

Ένας μηχανισμός κατηγοριοποίησης βασισμένος στην τεχνική Naïve Bayes δημιουργείται χρησιμοποιώντας πληροφορία εκπαίδευσης για να ευρεθεί η πιθανότητα κάθε κατηγορίας δεδομένου ενός κειμένου προς κατηγοριοποίηση. Το θεώρημα του Bayes μπορεί να χρησιμοποιηθεί για να υπολογιστεί η πιθανότητα:

$$P(C = c_k \mid \vec{x}) = \frac{P(\vec{x} \mid C = c_k)P(C = c_k)}{P(\vec{x})}$$

Ο πρώτος όρος του αριθμητή είναι συνήθως δύσκολο να υπολογιστεί χωρίς να απλοποιηθεί η παράσταση. Για το συγκεκριμένο μηχανισμό κατηγοριοποίησης, υποθέτουμε πως τα χαρακτηριστικά X1 ... Xn είναι ανεξάρτητα υπό όρους, δεδομένης μίας μεταβλητής κατηγορίας C. Αυτή η υπόθεση απλοποιεί την παραπάνω παράσταση στην:

$$P(\vec{x} \mid C = c_k) = \prod_i P(x_i \mid C = C_k)$$

Παρά το γεγονός ότι η θεώρηση της ανεξαρτησίας είναι γενικά αναληθής όσον αφορά την εμφάνιση κειμένων μέσα σε ένα έγγραφο, ο παραπάνω αλγόριθμος είναι αποτελεσματικός.

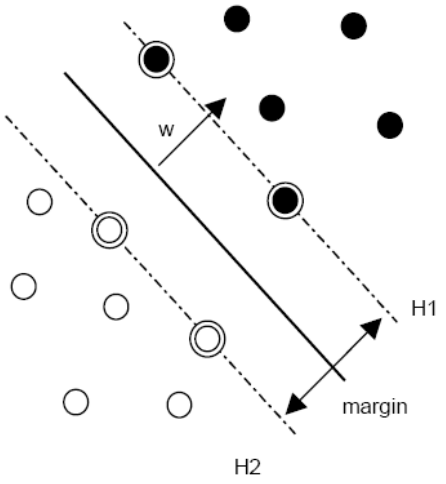
2.3.1.3 k-Nearest Neighbor (κοντινότερος γείτονας)

Ο αλγόριθμος kNN, είναι μία ακόμα στατιστική προσέγγιση στην αναγνώριση μοτίβου και κατηγοριοποίηση πληροφορίας [10]. Ο συγκεκριμένος αλγόριθμος, για ένα δοκιμαστικό κείμενο βρίσκει του κ κοντινότερους γείτονες ανάμεσα στα κείμενα εκπαίδευσης με την προσέγγιση να υπολογίζεται σαν μια ομοιότητα, και χρησιμοποιεί τις κατηγορίες των κ αυτών γειτόνων για να υπολογίσει τα βάρη με τα οποία θα συμμετέχει το κείμενο στην προσπάθεια ένταξης σε μία κατηγορία. Το αποτέλεσμα που εξάγεται υπολογίζοντας όλα τα βάρη, δίνει ένα αποτέλεσμα για την κατηγοριοποίηση του κειμένου.

2.3.1.4 Support Vector Machine

Το SVM είναι μια καινούρια μέθοδος κατηγοριοποίησης η οποία προτάθηκε από τον Vapnik [11; 12], και έχει ήδη αποκτήσει μεγάλη δημοσιότητα.

Στην πιο απλή του μορφή, ένα SVM ορίζεται σαν έναν υπερεπίπεδο που δύναται να διαχωρίσει ένα σύνολο θετικών από ένα σύνολο αρνητικών στοιχεία που αφορούν μια συγκεκριμένη κατηγορία. Αυτό φαίνεται και στο παρακάτω σχήμα όπου υποθέτοντας ότι οι μαύρες κουκίδες αφορούν τα θετικά στοιχεία και οι άσπρες τα αρνητικά στοιχεία ορίζεται με τη βοήθεια του SVM ένα μέγιστο υπερεπίπεδο που αποτελεί το διαχωριστικό ανάμεσα στα στοιχεία.



Εικόνα 4. Γραμμικά χωρισμένα υπερεπίπεδα

Στη γραμμική μορφή του αλγορίθμου, το περιθώριο μεταξύ των στοιχείων μπορεί να οριστεί σαν η απόσταση του υπερεπιπέδου από τα κοντινότερα θετικά και αρνητικά στοιχεία. Η μεγιστοποίηση αυτού του περιθωρίου μπορεί να αποτελέσει ένα πρόβλημα βελτιστοποίησης. Φυσικά τα περισσότερα παραδείγματα δε μπορούν να διαχωριστούν με τη χρήση της γραμμικής μορφής του αλγορίθμου γι' αυτό χρησιμοποιούνται πίνακες προκειμένου να υπολογιστούν τα περιθώρια και οι αποστάσεις.

Οι αλγόριθμοι για SVM έχουν αποδειχθεί ότι έχουν καλή γενικά απόδοση ακόμα και σε δύσκολα προβλήματα κατηγοριοποίησης μερικά από τα οποία είναι η αναγνώριση γραφικού χαρακτήρα, η αναγνώριση προσώπου, η κατηγοριοποίηση κειμένων. Η απλή γραμμική μορφή έχει πολύ καλή απόδοση, υφίσταται γρήγορη εκμάθηση και παράλληλα μπορεί να κατηγοριοποιεί εξαιρετικά γρήγορα.
Περισσότερα στοιχεία για το SVM μπορούν να βρεθούν στο [13].

2.4 Σημασιολογικός Ιστός και Μεταδεδομένα

Το Διαδίκτυο σήμερα αποτελεί τη μεγαλύτερη πηγή πληροφοριών. Μεγάλοι όγκοι δεδομένων αναζητούνται, ανταλλάσσονται και επεξεργάζονται μέσω του Παγκόσμιου Ιστού. Επειδή, όμως ο όγκος των δεδομένων του Ιστού έχει πάρει μεγάλες διαστάσεις χωρίς να υπάρχει ενιαίος τρόπος οργάνωσης, η ανταλλαγή και η επεξεργασία τους είναι πολύ δύσκολη. Ο Σημασιολογικός Ιστός έρχεται ακριβώς να εξυπηρετήσει την ανάγκη για ενιαία οργάνωση των δεδομένων, ώστε το Διαδίκτυο να γίνει μια αποδοτική παγκόσμια πλατφόρμα ανταλλαγής και επεξεργασίας από ετερογενείς πηγές πληροφορίας. Ένας γενικός ορισμός μας λέει ότι ο Σημασιολογικός Ιστός δίνει δομή, οργάνωση και σημασιολογία στα δεδομένα, ώστε να είναι, σε μεγάλο βαθμό, κατανοητά από μηχανές (machine understandable).

Ο όρος Σημασιολογικός Ιστός (Semantic Web) χρησιμοποιήθηκε για πρώτη φορά το 1998 από το δημιουργό του πρώτου φυλλομετρητή ιστοσελίδων και εξυπηρετητή διαδικτύου, Tim Berners-Lee. Από τότε καταβάλλεται μεγάλη προσπάθεια από την επιστημονική κοινότητα για την υλοποίησή του πάνω από τον Παγκόσμιο Ιστό. Στο βασικότερο επίπεδό του, ο Σημασιολογικός Ιστός αποτελεί μία συλλογή από συνοπτική πληροφορία για τη διακινούμενη πληροφορία, τα μεταδεδομένα, η οποία δεν είναι ορατή στον τελικό χρήστη. Τα μεταδεδομένα χρησιμοποιούνται για να περιγράψουν υπάρχοντα έγγραφα, ιστοσελίδες, βάσεις δεδομένων, προγράμματα που βρίσκονται στο διαδίκτυο. Οι εφαρμογές λογισμικού που κάνουν χρήση μεταδεδομένων αποκτούν καλύτερη κατανόηση της σημασιολογίας του περιεχομένου τους και άρα μπορούν να τα επεξεργαστούν με πιο αποδοτικό τρόπο. Η κατανόηση των μεταδεδομένων από τις μηχανές είναι δυνατή μέσω της χρήσης ειδικών λεξικών (των οντολογιών) τα οποία παρέχουν κοινούς κανόνες και λεξιλόγια για την ερμηνεία των δεδομένων. Με αυτό τον τρόπο είναι δυνατή η κοινή κατανόηση όρων και εννοιών από εφαρμογές που προέρχονται από διαφορετικά πληροφοριακά συστήματα. Απώτερος στόχος της όλης προσπάθειας είναι η ικανοποίηση των απαιτήσεων των συμμετεχόντων στην Κοινωνία της Πληροφορία για αυξημένη ποιότητα υπηρεσιών. Αυτό συνίσταται κυρίως στη βελτιωμένη αναζήτηση, εκτέλεση σύνθετων διεργασιών μέσω του Διαδικτύου και στην εξατομίκευση της πληροφορίας σύμφωνα με τις ανάγκες του εκάστοτε χρήστη.

Ένα από τα σημαντικότερα προβλήματα που καλείται να λύσει ο Σημασιολογικός Ιστός είναι η πρόσβαση στην πληροφορία. Σύμφωνα με πρόσφατες μελέτες, η ανθρωπότητα έχει παράγει από το 1999 μέχρι το 2003, τόσες νέες πληροφορίες όσες παρήγαγε όλα τα προηγούμενα χρόνια της ιστορίας της. Σε αυτό

το διάστημα των τριών τελευταίων ετών παρήχθησαν 12 exabytes πληροφορίας υπό τη μορφή έντυπου, οπτικού ή και ηχητικού υλικού. Η αυξανόμενη αυτή παραγωγή και η συνεχής βελτίωση των μεθόδων ψηφιοποίησης συμβάλλουν στην παραγωγή ενός ωκεανού ψηφιακών δεδομένων που προφανώς δύναται να δημιουργήσει μεγάλο αριθμό προβλημάτων. Το πιο σημαντικό ίσως από αυτά είναι ο τρόπος με τον οποίο θα μπορεί κανείς να διαχειριστεί όλη αυτή την πληροφορία. Δε θα πρέπει φυσικά να αμελούμε το γεγονός πως η ικανότητα παραγωγής, αποθήκευσης και μετάδοσης της πληροφορίας έχει ξεπεράσει κατά πολύ τις δυνατότητες αναζήτησης, πρόσβασης και παρουσίασης.

Λόγω του αυξανόμενου όγκου της πληροφορίας και των προβλημάτων αποτελεσματικής πρόσβασης, έχει γίνει τα τελευταία χρόνια ξεκάθαρο προς την επιστημονική κοινότητα ότι για την αύξηση της απόδοσης χρειάζονται νέες μέθοδοι υπολογισμού ικανές να προσαρμοστούν σε μία πληθώρα παραμέτρων τόσο αντικειμενικών όσο και υποκειμενικών. Η απόδοση ενός συστήματος πρόσβασης στην πληροφορία εκτιμάται μέσα από την ανάκληση και την ακρίβεια.

Η αναφορά στα προβλήματα που αντιμετωπίζουν τα σύγχρονα συστήματα πρόσβασης στην πληροφορία έχει άμεση σχέση με τον τύπο των ερωτήσεων που δέχονται ως είσοδο. Υπάρχουν δύο διαφορετικά είδη ερωτημάτων, οι ερωτήσεις γενικού περιεχομένου και ειδικού περιεχομένου. Το μέγεθος της απάντησης σε ερωτήσεις γενικού περιεχομένου είναι μεγάλο και παρουσιάζει εξαιρετικά μεγάλες αποκλίσεις ως προς τη σχετικότητα της ίδιας της ερώτησης. Το πρόβλημα εστιάζεται στην επιλογή ενός μικρού συνόλου από τις πιο σχετικές απαντήσεις, είναι δηλαδή πρόβλημα ακρίβειας. Αντίθετα, για τις ερωτήσεις ειδικού περιεχομένου, το διαθέσιμο σύνολο σχετικών απαντήσεων είναι μικρό και το πρόβλημα που προκύπτει είναι πρόβλημα ανάκτησης.

Εκτός από τα κλασσικά προβλήματα που αντιμετωπίζουν τα ΠΣ στον τομέα της πρόσβασης στην πληροφορία, αναδύονται και άλλα άμεσα συνδεδεμένα με το είδος της ίδιας της πληροφορίας:

- Συνωνυμία: ανάκτηση μη σχετικών απαντήσεων που περιέχουν όρους συνώνυμους με αυτούς της ερώτησης
- Ασάφεια / Διφορούμενες έννοιες: ανάκτηση μη σχετικών λόγω ασάφειας της ερώτησης ή λόγω ύπαρξης διφορούμενων εννοιών.
- Πειθώ των μηχανών αναζήτησης (search engine persuasion): ταξινόμηση των ανακτημένων εγγράφων με βάση το βαθμό σχετικότητας τους προς την ερώτηση έχοντας υπόψη τα προβλήματα της συνωνυμίας και της ασάφειας.

Τα τελευταία χρόνια, μια νέα ερευνητική προσπάθεια έχει επικεντρωθεί σε αυτό το πεδίο το οποίο ανήκει στην περιοχή που ονομάζεται Προσαρμοσμένη Πρόσβαση στην Πληροφορία. (Adaptive Information Access). Η πρόσβαση στην πληροφορία περιλαμβάνει αρκετές ερευνητικές περιοχές που θα μπορούσαν να συνδυαστούν για την κατασκευή συστημάτων ικανών να ανταποκριθούν στις σύγχρονες ανάγκες. Τέτοιες περιοχές είναι η έξυπνη αναζήτηση πληροφορίας, μάθηση μηχανής και αλληλεπίδραση ανθρώπου υπολογιστή. Στην παρούσα διπλωματική θα ασχοληθούμε με ζητήματα που έχουν να κάνουν τόσο με έξυπνη ανάκτηση πληροφορίας, με μάθηση μηχανής όσο και με αλληλεπίδραση χρηστών με τον υπολογιστή.

Κεφάλαιο

3

3 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Η αναζήτηση για σχετικές εργασίες μας φέρνει αντιμέτωπους με μία σειρά από συστήματα που έχουν αναπτυχθεί προκειμένου να διευκολύνουν τους χρήστες κατά την προσπάθεια εξόρυξης πληροφορίας και επεξεργασίας κειμένου. Τα συστήματα αυτά έχουν, το καθένα, ένα διαφορετικό τρόπο προσέγγισης του θέματος. Σε άλλα σημεία συγκλίνουν και σε άλλα αποκλίνουν ενώ η γενική ιδέα εντοπίζεται κυρίως στους μηχανισμούς κατηγοριοποίησης.

3.1 Συλλογή Δεδομένων από το Διαδίκτυο

Για τη συλλογή δεδομένων από το διαδίκτυο χρησιμοποιούνται οι ευρέως γνωστοί crawlers. Το πλήθος τους είναι αμέτρητο ενώ, αν εξαιρέσουμε τους εξειδικευμένους crawlers (focused crawlers) παρατηρούμε πως οι περισσότεροι έχουν σαν σκοπό να συλλέξουν όλες τις HTML σελίδες από τις οποίες απαρτίζεται ένας δικτυακός τόπος μαζί με τα βοηθητικά αρχεία (pdf, εικόνες, video, css, javascript) και ουσιαστικά να δημιουργήσουν ένα offline-instance του δικτυακού τόπου τον οποίο προσπελαύνουν. Οι crawlers που έχουν κατασκευαστεί για το διαδίκτυο αγγίζουν σε αριθμό τις μερικές χιλιάδες καθώς η κατασκευή τους είναι σχεδόν τετριμμένη. Στη συνέχεια θα παρουσιάσουμε συγκεκριμένου crawlers που αξίζουν προσοχής για τα ιδιαίτερα χαρακτηριστικά που παρουσιάζουν.

3.1.1 RBSE

Ο RBSE ήταν ο πρώτος web crawler που δημοσιεύτηκε. Βασίστηκε σε δύο προγράμματα: το πρώτο πρόγραμμα, "`spider`" διατηρεί μια ουρά σε μια σχεσιακή βάση δεδομένων, και το δεύτερο πρόγραμμα "`mite`" είναι ένας τροποποιημένος ASCII browser που μεταφορτώνει τις σελίδες από τον Ιστό.

3.1.2 World Wide Web Worm

Ήταν ένας crawler που χρησιμοποιήθηκε για την κατασκευή ενός απλού πίνακα που περιείχε τίτλους και URLs κάποιων εγγράφων. Ο πίνακας θα μπορούσε να αναζητηθεί με τη χρησιμοποίηση της εντολής `grep` στο Unix.

3.1.3 Internet Archive Crawler

Είναι ένας crawler που σχεδιάστηκε με σκοπό την αρχειοθέτηση περιοδικών στιγμιότυπων ενός μεγάλου τμήματος του Web. Χρησιμοποιεί αρκετές διεργασίες και ένας καθορισμένος αριθμός από ιστοσελίδες ανατίθεται σε κάθε μία από αυτές. Η διεργασία που ανταλλάσσει URLs μεταφέρεται ανά ομάδες με μεγάλο χρονικό

διάστημα μεταξύ των ανταλλαγών, καθώς αυτή είναι μία πολυδάπανη διαδικασία. Ο Internet Archive Crawler πρέπει ακόμα να αντιμετωπίσει και το πρόβλημα της αλλαγής των DNS records, γι' αυτό διατηρεί ένα ιστορικό αρχείο του hostname στις χαρτογραφήσεις του IP.

3.1.4 Webcrawler

Πρόκειται για έναν από τους πρώτους crawlers που κατασκευάστηκαν από τον Pinkerton το 1994. Βασίστηκε στη βιβλιοθήκη WWW προκειμένου να είναι σε θέση να κατεβάζει σελίδες από το διαδίκτυο ενώ χρησιμοποιούσε ένα δεύτερο πρόγραμμα προκειμένου να διαβάζει τα URL τα οποία πρέπει να προσπελάσει. Ο αλγόριθμος προσπέλασης ήταν κατά πλάτος αναζήτηση του γραφήματος μίας ιστοσελίδας σε συνδυασμό με αποφυγή των σελίδων που έχει ήδη επισκεφθεί. Ένα αξιοσημείωτο στοιχείο ήταν η δυνατότητα να ακολουθεί συγκεκριμένα μόνο links σε ένα δικτυακό τόπο – και όχι όλα – βάση του ερωτήματος που έθετε ο χρήστης. Ήταν κάτι σαν ένας crawler πραγματικού χρόνου που φυσικά μπορούσε να ανταποκριθεί πλήρως λόγω του μικρού μεγέθους που είχε το διαδίκτυο.

3.1.5 Google Crawler

Ένας από τους πιο σημαντικούς crawlers που κατασκευάστηκαν και διατηρούνται ακόμα και σήμερα, με σημαντικές βέβαια βελτιώσεις είναι ο Google Crawler των Brin και Page, 1998 . Βασίζεται στις γλώσσες προγραμματισμού C++ και Python και παρουσιάζει εξαιρετικά μεγάλη πολυπλοκότητα. Επειδή η χρήση των σελίδων που κατέβαζε ο crawler προοριζόταν για εκτενή αναζήτηση μέσα σε σειρές από κείμενα, ο συγκεκριμένος crawler βασίστηκε στη διαδικασία indexing. Στο μηχανισμό υπάρχει ένας URL εξυπηρετητής που αποστέλλει λίστες με URL προς τους crawlers του συστήματος οι οποίοι λειτουργούν παράλληλα. Οι crawlers εξάγουν από τις σελίδες το κείμενο αλλά και όσα URLs εντοπίζουν. Αυτά στέλνονται πίσω στον URL εξυπηρετητή για έλεγχο και σε περίπτωση που δεν τα έχει επισκεφθεί ποτέ ο crawler προστίθενται στη λίστα του εξυπηρετητή.

3.1.6 Mercator

Ο Mercator είναι ένας κατανεμημένος τμηματοποιημένος web crawler γραμμένος εξ' ολοκλήρου σε γλώσσα προγραμματισμού Java. Η τμηματοποίηση του προκύπτει από τη χρήση δύο διαφορετικών πρωτοκόλλων. Τα τμήματα πρωτοκόλλων είναι υπεύθυνα για την ομαλή σύνδεση του μηχανισμού στις σελίδες και για την εξασφάλιση πως ο μηχανισμός θα είναι σε θέση να κατεβάσει τη σελίδα. Και από την άλλη μεριά τα τμήματα επεξεργασίας είναι αυτά που αφορούν την ανάλυση της σελίδας και την εξαγωγή του κειμένου και συνδέσμων από αυτή. Η απλή διαδικασία επεξεργασίας περιλαμβάνει ανάλυση της σελίδας και εξαγωγή των συνδέσμων που αυτή περιέχει ενώ σε μία πιο σύνθετη μορφή της περιλαμβάνει αλγορίθμους για την αποτελεσματική εξαγωγή του κειμένου.

3.1.7 WebFountain

Πρόκειται για έναν κατανεμημένο τμηματικό crawler παραπλήσιο του mercator, με τη διαφορά ότι είναι γραμμένος σε C++. Περιλαμβάνει ένα κεντρικό μηχανισμό και μία σειρά από "ant" (μερμύγκι) μηχανισμούς. Πρόκειται δηλαδή για το ρυθμιστή της κατάστασης και τους εργάτες. Ο μηχανισμός αυτός περιέχει στοιχεία

που τον κάνουν πολύ φιλικό προς τις σελίδες που επισκέπτεται. Σκοπός του είναι η διατήρηση ενός off-line instance του διαδικτύου. Αυτό έχει σαν αποτέλεσμα, μία από τις μετρικές τις οποίες προσμετρά ο συγκεκριμένος μηχανισμός να είναι το κατά πόσο η σελίδες που διαθέτει ανταποκρίνονται στις πραγματικές σελίδες που βρίσκονται on-line στους δικτυακούς τόπους και όχι απλά μία παλαιότερη έκφανσή τους. Για να πετύχει μεγαλύτερο freshness όπως ονομάζεται η συγκεκριμένη μετρική χρησιμοποιεί διαφορετική συχνότητα επίσκεψης στις σελίδες που έχει αποθηκευμένες στη βάση δεδομένων του.

3.1.8 WebRACE

Πρόκειται για έναν crawler ο οποίος είναι γραμμένος σε Java και αποτελεί ένα κομμάτι ενός γενικότερου συστήματος που ονομάζεται eRACE. Το συγκεκριμένο σύστημα λαμβάνει εντολές από τους τελικούς χρήστες για να ξεκινήσει να κατεβάσει σελίδες και συμπεριφέρεται σαν proxy server. Το σύστημα μπορεί να εξυπηρετήσει και αιτήσεις για αλλαγές στοιχείων σε σελίδες: μόλις μία σελίδα αλλάξει, τότε ο crawler την ξανακατεβάζει και ειδοποιεί τον τελικό χρήστη που ενδιαφέρεται πως η σελίδα έχει αλλάξει και πως πλέον στον proxy είναι αποθηκευμένη μία νέα σελίδα. Το πιο σημαντικό στοιχείο του συγκεκριμένου crawler είναι η χαρακτηριστική διαφορά που παρουσιάζει συγκριτικά με όσους crawlers έχουμε δει. Στο συγκεκριμένο crawler δεν υπάρχει ένα feed URL από το οποίο θα ξεκινήσει να αναζητά σελίδες. Το URL feed είναι δυναμικό και διαμορφώνεται από τα ερωτήματα των χρηστών. Μετά τη χρήση του καταστρέφεται και ο μηχανισμός βρίσκεται σε αναμονή μέχρι να του δοθεί κάποιο νεότερο ερώτημα.

3.1.9 Ubicrawler

Ο Ubicrawler είναι ένας κατανεμημένος crawler γραμμένος σε Java και δε διαθέτει κεντρικοποιημένη διαδικασία. Είναι κατασκευασμένος από έναν αριθμό από όμοιους “agents” και μία συνάρτηση ανάθεση που αναθέτει σε κάθε agent κάποια εργασία. Οι agents δεν επικοινωνούν μεταξύ τους άμεσα αλλά όλες οι διαδικασίες διευθετούνται από την κεντρική συνάρτηση ανάθεσης. Καμία σελίδα δεν προσπελαύνεται διπλή φορά καθώς κάθε agent φροντίζει να ενημερώσει για τις σελίδες που έχει επισκεφτεί εκτός και αν κάποιος από τους agents καταστραφεί. Πρόκειται για έναν πολύ σταθερό crawler, σχεδιασμένο με τέτοιον τρόπο ώστε να πετυχαίνει μέγιστη κλιμάκωση και μικρή ευαισθησία σε σφάλματα.

3.1.10 FAST Crawler

Πρόκειται για ένα crawler που χρησιμοποιήθηκε από την μηχανή αναζήτησης FAST και η γενική περιγραφή της αρχιτεκτονικής του είναι διαθέσιμη. Είναι μία διανεμημένη αρχιτεκτονική στην οποία κάθε μηχανή κρατά έναν "χρονοπρογραμματιστή εγγραφών" που διατηρεί μια ουρά από εγγραφές, που πρέπει να γίνουν download από έναν "επεξεργαστή εγγράφων", ο οποίος τις αποθηκεύει σε ένα υποσύστημα τοπικής αποθήκευσης. Κάθε crawler επικοινωνεί με άλλους crawlers μέσω ενός μοντέλου "διανομέων" που ανταλλάσσει πληροφορίες υπερσυνδέσμων.

3.1.11 WIRE

Είναι ένας Web crawler γραμμένος σε γλώσσα C++, που περιλαμβάνει διάφορες πολιτικές για το σχεδιασμό των downloads μίας σελίδας και ένα μοντέλο για την παραγωγή εκθέσεων και στατιστικών πάνω στις σελίδες που έγιναν download έτσι ώστε να χρησιμοποιηθούν για το χαρακτηρισμό του Web.

3.1.12 Crawlers Ανοιχτού Κώδικα

Μία σειρά από crawlers ανοιχτού κώδικα διανέμονται ελεύθερα στο διαδίκτυο. Κυρίως είναι προϊόντα κάποιου ιδιώτη που κατασκευάζονται για να καλύψουν συγκεκριμένες ανάγκες που έχουν οι τελικοί χρήστες, ανάγκες που συχνά δεν καλύπτονται από τους εμπορικούς crawlers. Η χρήση τους είναι συνήθως ως εξής. Κάποιος χρήστης που δεν καλύπτεται από έναν εμπορικό crawler λαμβάνει το κώδικα ενός open source συστήματος και το αλλάζει με σκοπό να το φέρει στα μέτρα του. Συνήθως οι open source crawlers δεν έχουν εξειδικευμένες λειτουργικότητες ωστόσο προσφέρονται στους τελικούς χρήστες οι οποίοι μπορούν να τους τροποποιήσουν ελεύθερα.

3.2 Μηχανισμοί Επεξεργασίας Πληροφορίας

3.2.1 Ad hoc ανάκτηση και φιλτράρισμα

Στα περισσότερα συστήματα ΑΠ, η συλλογή των κειμένων παραμένει σχεδόν στατική (ακόμα και αν αλλάζει π.χ. μια φορά τη μέρα θεωρείται στατική), ενώ συνέχεια υποβάλλονται καινούρια ερωτήματα. Αυτός ο τρόπος λειτουργίας έχει ονομαστεί ad hoc ανάκτηση πληροφορίας και είναι η πιο κοινή μορφή διαδικασίας χρήστη. Μια δεύτερη παρόμοιας μορφής διαδικασία χρήστη είναι το φιλτράρισμα πληροφορίας (information filtering). Σε αυτή τη διαδικασία, τα ερωτήματα παραμένουν σχεδόν σταθερά, ενώ η συλλογή των κειμένων μεταβάλλεται με καινούρια κείμενα να φτάνουν συνεχώς στο σύστημα. Παράδειγμα της τελευταίας διαδικασίας είναι η λίστα αλληλογραφίας ή μία υπηρεσία πληροφόρησης για το χρηματιστήριο.

Στο φιλτράρισμα κατασκευάζεται ένα προφίλ χρήστη, το οποίο περιγράφει τις προτιμήσεις του. Το προφίλ συγκρίνεται με κάθε εισερχόμενο κείμενο για να αποφασίσει το σύστημα αν είναι σχετικό ή όχι το κείμενο με τις προτιμήσεις του χρήστη. Με άλλα λόγια το προφίλ είναι μία εναλλακτική μορφή ερωτήματος προς το σύστημα. Μια πιθανή εφαρμογή είναι το φιλτράρισμα ειδήσεων που φθάνουν δεκάδες, on-line, έτσι ώστε να παρέχονται στο χρήστη αυτές που πιθανόν τον ενδιαφέρουν.

3.2.2 Stemming

Στην ΑΠ η σχέση μεταξύ ενός ερωτήματος χρήστη και ενός κειμένου καθορίζεται κυρίως από το πλήθος των όρων που έχουν κοινούς. Δυστυχώς, οι λέξεις έχουν πολλές μορφολογικές παραλλαγές οι οποίες δεν αναγνωρίζονται από αλγόριθμους που βασίζονται στο ταίριασμα όρων χωρίς να προηγηθεί κάποιας μορφής επεξεργασία φυσικής γλώσσας (Natural Language Processing). Στις

περισσότερες των περιπτώσεων, αυτές οι παραλλαγές έχουν παρόμοιες εννοιολογικές ερμηνείες και μπορούν να αντιμετωπιστούν ως ισοδύναμες στα πλαίσια εφαρμογών ΑΠ. Ως εκ τούτου, ένα πλήθος αλγορίθμων κατάλληλων για τη διαδικασία του stemming έχουν αναπτυχθεί ώστε να περιορίσουν τις μορφολογικές παραλλαγές στην αρχική τους ρίζα.

Το πρόβλημα του stemming έχει προσεγγιστεί από μία μεγάλη ποικιλία μεθόδων που περιγράφονται στο [16] και περιλαμβάνουν αφαίρεση της κατάληξης, τμηματοποίηση λέξης και λεξιλογική μορφοποίηση.

3.2.3 HTML Parsers

Με τον όρο HTML parsers αναφερόμαστε σε μηχανισμούς που έχουν αναπτυχθεί σε διάφορες γλώσσες προγραμματισμού, κυρίως αντικειμενοστραφείς, οι οποίοι σαν στόχο έχουν τον έλεγχο του κώδικα HTML για συντακτικά λάθη και την αναπαράσταση του κώδικα σε δενδρική κυρίως δομή (parse tree, abstract syntax tree). Συναντούμε διάφορες κατηγορίες HTML parsers, όπως TP (Top-Down) parsers, BU (Bottom-Up) parsers, LL (Left-to-Right) parsers κ.α.

3.3 Κατηγοριοποίηση Πληροφορίας

Η αυτόματη κατηγοριοποίηση κειμένων είναι η διαδικασία ανάθεσης ετικετών κατηγορίας (προκαθορισμένων) σε νέα κείμενα που καταφθάνουν, στηριζόμενη στην πιθανότητα η οποία προτείνεται από τη βάση γνώσης που προϋπάρχει. Η διαδικασία έχει εγείρει ορισμένες προκλήσεις για τις στατιστικές μεθόδους που συνήθως χρησιμοποιούνται, και την αποτελεσματικότητά τους στην επίλυση πραγματικών προβλημάτων, τα οποία συχνά είναι πολλών διαστάσεων και έχουν μη σαφώς καθορισμένη κατανομή μεταξύ των κειμένων προς κατηγοριοποίηση. Η ανίχνευση του θέματος ενός κειμένου, για παράδειγμα, είναι η πιο κοινή εφαρμογή της κατηγοριοποίησης κειμένων. Ένας ολοένα και αυξανόμενος αριθμός μεθόδων αντιμετώπισης του προβλήματος προτείνονται, μεταξύ των οποίων μοντέλα παλινδρόμησης, κατηγοριοποίηση κοντινότερων γειτόνων, πιθανοτικές προσεγγίσεις με μεθόδους Bayes, επαγωγική εκμάθηση κανόνων, νευρωνικά δίκτυα, on-line εκμάθηση και Support Vector Machines. Παρότι η πλούσια βιβλιογραφία που υπάρχει πάνω στον τομέα της κατηγοριοποίησης κειμένων, ασφαλείς εκτιμήσεις και συγκρίσεις μεταξύ των μεθόδων είναι συνήθως δύσκολες.

Άλλες επαναστατικές τεχνικές, όπως η χρήση κωδικών ελέγχου, η χρήση αιτιολογικών δικτύων έχουν προταθεί και αποτελούν ουσιαστικά μια τροποποιημένη έκδοση του Bayes αλγορίθμου του που αποδίδουν καλά σε εργασίες κατηγοριοποίησης κειμένων. Καμία από τις προηγούμενες τεχνικές δεν αντιμετωπίζει τα σημασιολογικά θέματα.

Κεφάλαιο

4

4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ, ΠΡΟΔΙΑΓΡΑΦΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Τα συστήματα ανάκτησης και διαχείρισης πληροφορίας του Διαδικτύου γνωρίζουν μεγάλη άνθιση τον τελευταίο καιρό. Ο λόγος που συμβαίνει αυτό είναι για να παρέχεται στους χρήστες του Διαδικτύου ποιοτική πληροφορία καθώς η πληροφορία που διακινείται στον παγκόσμιο ιστό είναι υπερβολική. Έτσι πολλοί χρήστες είτε γιατί δεν έχουν το χρόνο, είτε γιατί δεν έχουν τη γνώση, δε μπορούν να «φτάσουν» στην επιθυμητή γι' αυτούς πληροφορία. Οι πιο γνωστές μηχανές αναζήτησης όπως είναι το Google[17] και το Altavista[18] σε ερωτήματα όπως : "politics", "sports", "economy", "local news (Greece)" κ.α. συνήθως έχουν ως απαντήσεις εκατομμύρια ιστοσελίδες, αρκετές από τις οποίες δε περιλαμβάνουν πληροφορία σχετική με την αναζήτησή τους, έχοντας σαν συνέπεια ο χρήστης να μη μπορεί να αποφασίσει ποια είναι η απάντηση που ταιριάζει περισσότερο με αυτό που προσπαθεί να βρει.

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός συστήματος κατηγοριοποίησης των εκατομμυρίων ιστοσελίδων που παρέχονται στο διαδίκτυο, ενός μηχανισμού που ουσιαστικά θα βγάζει το χρήστη από τη χρονοβόρα και συχνά αδιέξοδη διαδικασία της ανεύρεσης χρήσιμης πληροφορίας.

Συνεπώς πρωταρχικός μας στόχος είναι η βελτίωση των χαοτικών πλέον συνθηκών που επικρατούν στο διαδίκτυο.

Στο παρόν κεφάλαιο θα ασχοληθούμε με θέματα που έχουν να κάνουν με τους στόχους, τη γενική αρχιτεκτονική και τις προδιαγραφές του συστήματος που κατασκευάσαμε καθώς επίσης και με τα γενικά χαρακτηριστικά του.

4.1 Στόχοι του συστήματος

Το σύστημα έχει σχεδιαστεί με στόχο να παρέχει στην έξοδό του ένα σύνολο με ιστοσελίδες που θα έχουν κατανεμηθεί σε διακριτές κατηγορίες βάση του περιεχομένου τους.

Σκοπός μας είναι να κατασκευάσουμε τους κατάλληλους μηχανισμούς και τη κατάλληλη υποδομή ώστε αρχικά να δημιουργούμε μία σύνδεση με το server που βρίσκεται η υπό ανάκτηση ιστοσελίδα με σκοπό να ανακτήσουμε το κώδικα HTML της ιστοσελίδας ως κείμενο, στη συνέχεια «φιλτράροντας» το κώδικα με διάφορους μηχανισμούς επεξεργασίας κειμένου να ανακτούμε το καθαρό κείμενο που αποτελεί την ωφέλιμη πληροφορία κάθε ιστοσελίδας και τέλος κάνωντας χρήση του μηχανισμού κατηγοριοποίησης να κατηγοριοποιούμε τη κάθε ιστοσελίδα σύμφωνα με το περιεχόμενό της.

Έτσι στο τέλος θα έχουμε στη διάθεσή μας ένα σύνολο διευθύνσεων ιστοτόπων που θα ανήκουν σε διάφορες κατηγορίες, έτοιμο να διατεθεί στους χρήστες που ενδιαφέρονται να πραγματοποιήσουν μία εξειδικευμένη περιήγηση στο διαδίκτυο.

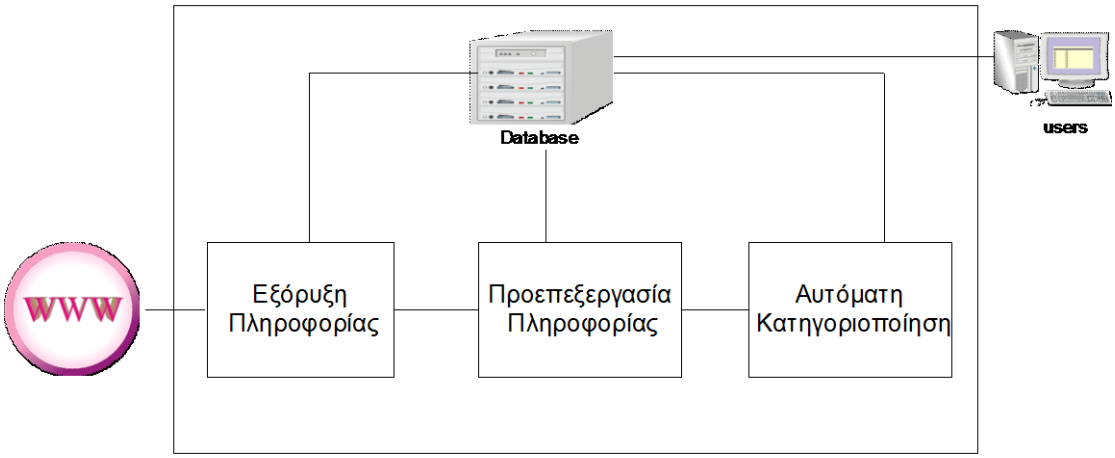
4.2 Αρχιτεκτονική του συστήματος

Η αρχιτεκτονική του συστήματος είναι αρκετά πολύπλοκη αν σκεφτεί κανείς πως πρόκειται για ένα σύστημα που περιλαμβάνει υποσυστήματα.

4.2.1 Γενική Αρχιτεκτονική

Γνωρίζουμε πως πολλές προσπάθειες του παρελθόντος έχουν καταλήξει στη δημιουργία συστημάτων ανεύρεσης και ανάλυσης πληροφορίας με σκοπό να παρέχουν στον τελικό χρήστη ποιότητα στα δεδομένα με τα οποία έρχεται σε επαφή. Τα συστήματα που έχουν αναπτυχθεί κατά καιρούς χρησιμοποιούν ένα γενικό μοντέλο που χαρακτηρίζει όλα σχεδόν τα συστήματα που έχουν τέτοιο σκοπό. Αξίζει να τονίσουμε πως η πλειοψηφία των συστημάτων αυτών βασίζεται σε παράλληλες τεχνολογίες, κατανεμημένα συστήματα, τεχνολογίες grid κλπ. γιατί η υπολογιστική ισχύς που απαιτείται για την επεξεργασία μεγάλου όγκου κειμένων είναι υπερβολική για να την «αντέξει» ένα απλός επιτραπέζιος υπολογιστής και να την πραγματοποιήσει σε πραγματικό χρόνο.

Στη παρακάτω εικόνα μπορεί κάποιος να δει τη γενική δομή της αρχιτεκτονικής που ακολουθήσαμε για την υλοποίηση του συστήματός μας.



Εικόνα 5. Η αρχιτεκτονική του συστήματός μας

Αρχικά στο πρώτο στάδιο, υπάρχει ένας μηχανισμός που έχει ως σκοπό να ανακτά σελίδες από το διαδίκτυο. Τα στοιχεία που λαμβάνει ο συγκεκριμένος μηχανισμός βρίσκονται αποθηκευμένα στη βάση δεδομένων, τα οποία είναι τα διαθέσιμα URL των ιστοσελίδων που επιθυμούμε να επεξεργαστούμε. Ο κώδικας που ανακτάται αποθηκεύεται δυναμικά και όχι στη βάση δεδομένων του συστήματος. Πρόκειται για ένα υποσύστημα που ανακτά το κώδικα HTML των ιστοσελίδων, τον οποίο διοχετεύει ως είσοδο στον επόμενο μηχανισμό, το υποσύστημα που αναλαμβάνει την επεξεργασία της διαθέσιμης πληροφορίας. Πρόκειται για ένα

μηχανισμό που αναλαμβάνει να εξάγει χρήσιμο κείμενο από την πληροφορία που έχει ανακτήσει ο προηγούμενος μηχανισμός, το οποίο και αποθηκεύει στη βάση δεδομένων του συστήματος.

Μετά το πέρας των παραπάνω διαδικασιών στη βάση δεδομένων είναι αποθηκευμένες οι λέξεις μήκους μεγαλύτερου του τέσσερα από κάθε ιστοσελίδα, καθώς στη πλειονότητα αυτές είναι οι αντιπροσωπευτικότερες λέξεις κάθε κειμένου. Πάνω σε αυτές τις λέξεις κάθε κειμένου δρα ο τελευταίος μηχανισμός του συστήματός μας που πραγματοποιεί τη διαδικασία της κατηγοριοποίησης προκειμένου να ευρεθεί η κατηγορία στην οποία ανήκει κάθε κείμενο. Έτσι μετά τη πλήρη εκτέλεση του συστήματός μας παίρνουμε στη βάση δεδομένων ένα σύνολο από διευθύνσεις ιστοσελίδων, τη κατηγορία που ανήκει η κάθε μία καθώς και τη συσχέτιση, που μας δίνει τη συνάφεια του περιεχομένου της κάθε ιστοσελίδας με τη κατηγορία που εντάχθηκε.

4.3 Προδιαγραφές του συστήματος

Οι προδιαγραφές του συστήματος αναφέρονται περισσότερο στο πως πρέπει και πως είναι δημιουργημένο κάθε κομμάτι του μηχανισμού προκειμένου να επιτευχθεί ορθή διαδικασία και συνεπώς ορθό αποτέλεσμα.

4.3.1 Εξόρυξη πληροφορίας

Για την εξόρυξη της πληροφορίας από τους διάφορους δικτυακούς τόπους, παρότι υπήρχε η δυνατότητα να χρησιμοποιήσουμε έναν υπάρχον crawler καθώς υπάρχουν χιλιάδες τέτοιοι μηχανισμοί στο διαδίκτυο, είτε γενικού σκοπού, είτε εστιασμένοι (focused crawlers) όπως αναφέρεται στη παράγραφο 3.1, προτιμήσαμε να αναπτύξουμε ένα σχετικά απλό και αυτόνομο υποσύστημα, το οποίο δεν επιβαρύνει ιδιαίτερα τη λειτουργία του συστήματος, βασιζόμενοι σε υπάρχοντες μηχανισμούς τους οποίους παραμετροποιήσαμε στις ανάγκες της εργασίας μας.

4.3.2 Επεξεργασία πληροφορίας

Όπως αναφέραμε στην προηγούμενη ενότητα σκοπός μας είναι να παρέχουμε στο μηχανισμό που πραγματοποιεί κατηγοριοποίηση πληροφορίας καθαρό κείμενο απαλλαγμένο από κάθε κομμάτι κώδικα που μπορεί να καταστρέψει τη διαδικασία. Για το λόγο αυτό, η πληροφορία προτού φτάσει στο μηχανισμό κατηγοριοποίησης θα πρέπει να υποστεί κάποιου είδους έλεγχο και επεξεργασία. Αυτές τις λειτουργίες, αναλαμβάνει να πραγματοποιήσει ο μηχανισμός επξεργασίας.

Γνωρίζουμε πως από το μηχανισμό εξόρυξης θα λαμβάνουμε σελίδες HTML. Συνεπώς το στοιχείο στο οποίο θα πρέπει να δοθεί ιδιαίτερο βάρος είναι οι σελίδες HTML και η δομή τους. Ο μηχανισμός επεξεργασίας θα πρέπει να έχει τη δυνατότητα με έξυπνες διαδικασίες, να εντοπίζει το ωφέλιμο κείμενο μιας HTML σελίδας και να το εξάγει από αυτή προκειμένου να το παρέχει σαν είσοδο στον μηχανισμό κατηγοριοποίησης. Αφού γίνει ανάγνωση ενός αρχείου HTML εν συνεχεία το περιεχόμενό του θα πρέπει να αποθηκεύεται σε μία μεταβλητή προκειμένου να είναι εφικτή η επεξεργασία και εύρεση κειμένου. Στη συγκεκριμένη περίπτωση θα γίνει χρήση των συναρτήσεων που παρέχει έτοιμες η γλώσσα προγραμματισμού που χρησιμοποιείται (Java), οι οποίες είναι σε θέση να

πραγματοποιήσουν άμεση επεξεργασία σε ένα κείμενο και να εξάγουν στοιχεία από αυτό. Σκοπός μας λοιπόν είναι να συνδυάσουμε συναρτήσεις των βιβλιοθηκών της Java ώστε να απομονώσουμε το σώμα ενός κώδικα HTML και στη συνέχεια να το απαλλάξουμε από κώδικα που περιέχεται σε αυτό και υλοποιεί διάφορες λειτουργίες της ιστοσελίδας που δεν αποτελούν ωφέλιμη πληροφορία. Στη συνέχεια θα αφαιρέσουμε από το κείμενο τις λέξεις με μήκος μικρότερο ή ίσο του τέσσερα. Ιδιαίτερη μνεία σε αυτό το σημείο θα πρέπει να κάνουμε στη χρήση κανονικών εκφράσεων (regular expressions), μηχανισμός ο οποίος ενσωματώνεται στη γλώσσα Java, τον οποίο χρησιμοποιούμε για την υλοποίηση της παραπάνω διαδικασίας στο «καθαρό» πλέον κείμενο.

Η διαδικασία αυτή μοιάζει απλοϊκή για κάποιον που σκέφτεται μία αφαιρετική κατάσταση όπου κάποιος αποφασίζει για το τι είναι κείμενο και τι όχι σε μία σελίδα. Ωστόσο, θα πρέπει να αναλογιστούμε πως είναι μια αυτοματοποιημένη διαδικασία την οποία καλείται να πραγματοποιήσει μια μηχανή. Επιπρόσθετα θα πρέπει να ληφθεί υπόψη πως οι σελίδες που θα υποστούν προεπεξεργασία θα είναι ανομοιόμορφες μεταξύ τους με αποτέλεσμα να είναι απαραίτητη η δημιουργία ενός πολύ ευέλικτου μηχανισμού.

4.3.3 Κατηγοριοποίηση πληροφορίας

Μέχρι το συγκεκριμένο σημείο το σύστημα έχει ανακτήσει σελίδες από τον παγκόσμιο ιστό και έχει εξάγει από αυτές το χρήσιμο περιεχόμενο ούτως ώστε να είναι σε θέση να λειτουργήσει ο μηχανισμός κατηγοριοποίησης της πληροφορίας.

Η λειτουργία του συγκεκριμένου μηχανισμού θα πρέπει να βασίζεται στον αλγόριθμο κατηγοριοποίησης SVM και πιο συγκεκριμένα να πραγματοποιεί LSI. Στο σύστημα θα εφαρμοστεί μία πρωτότυπη μέθοδος κατηγοριοποίησης που θα βασίζεται στην ανάλυση προτάσεων και όχι ολόκληρων των παραγράφων των κειμένων. Πιο συγκεκριμένα, η ανάλυση θα είναι διαφορετική για τις διαφορετικές ομάδες χρηστών. Οι ομάδες χρηστών θα περιλαμβάνουν ουσιαστικά ομάδες όμοιων χαρακτηριστικών και ομάδες ίδιας αφαίρεσης πληροφορίας. Όσο μεγαλύτερη είναι η αφαίρεση πληροφορίας σε τόσο λιγότερες προτάσεις ενός κειμένου πραγματοποιείται κατηγοριοποίηση του κειμένου και συνεπώς η κατηγορία στην οποία εντάσσεται ένα κείμενο είναι πιο γενική.

Η παραπάνω διαδικασία θα έχει σαν αποτέλεσμα να δημιουργηθεί πολλαπλού είδους κατηγοριοποίηση στα κείμενα τα οποία θα διαθέτει το σύστημα με αποτέλεσμα να είναι διαφορετικά τα αποτελέσματα για κάθε χρήστη ανάλογα με τη λεπτομέρεια της αναζήτησης που πραγματοποιούν. Το ένα είδος κατηγοριοποίησης θα είναι καθαρά αλγοριθμικό ενώ το δεύτερο κομμάτι θα βασίζεται κυρίως στις ομάδες χρηστών που δημιουργούν κατηγορίες αφαίρεσης πληροφορίας.

4.4 Λειτουργικότητα του συστήματος

Το σύστημα που υλοποιήθηκε βασίζεται σε γενικές αρχιτεκτονικές και ανοικτά πρότυπα. Ωστόσο είναι πολύ εύκολο να οριστούν τα βασικά στοιχεία λειτουργικότητας τα οποία μπορούν να χωριστούν σε τρεις διακριτές κατηγορίες: ο μηχανισμός ανάκτησης πληροφορίας από το διαδίκτυο, ο μηχανισμός επεξεργασίας και ο μηχανισμός αυτόματης κατηγοριοποίησης.

4.4.1 Μηχανισμός Ανάκτησης Πληροφορίας

Όπως αναφέραμε και στις προδιαγραφές του μηχανισμού ανάκτησης πληροφορίας, η υλοποίησή του βασίστηκε στις βασικές λειτουργίες ενός τέτοιου μηχανισμού γενικού σκοπού. Αρχικά ο μηχανισμός αυτός αφού ανακτήσει από τη βάση δεδομένων του συστήματος τις επιθυμητές διευθύνσεις των ιστοτόπων ελέγχει εάν υπάρχει η δυνατότητα σύνδεσης στο server που βρίσκεται αποθηκευμένος ο κώδικας HTML της εκάστοτε ιστοσελίδας. Αφού εγκαθιδρύσει την απαιτούμενη σύνδεση, ο μηχανισμός ανακτά από το server το διαθέσιμο κώδικα, τον οποίο αποθηκεύει δυναμικά ως κείμενο χωρίς να παρεμβάλλεται σε αυτό το σημείο η βάση δεδομένων.

4.4.2 Μηχανισμός Επεξεργασίας

Ο μηχανισμός επεξεργασίας πληροφορίας υλοποιήθηκε σε τρία στάδια, τα οποία λειτουργούν σειριακά καθώς το καθένα δέχεται είσοδο από το προηγούμενο. Στο πρώτο στάδιο επεξεργασίας διατηρούμε το κυρίως σώμα του κώδικα της ιστοσελίδας. Στο δεύτερο στάδιο αφαιρούμε όσα κομμάτια κώδικα υλοποιούν κάποιες λειτουργίες της ιστοσελίδας. Ιδιαίτερη προσοχή σε αυτό το σημείο θα πρέπει να δώσουμε στη σύνταξη ενός κώδικα HTML, καθώς όπως παρατηρήσαμε είναι πολύ εύκολο να δημιουργηθεί σύγχυση σχετικά με το τι αφορά ωφέλιμη πληροφορία και το τι όχι. Για παράδειγμα ένα κομμάτι κώδικα που υλοποιεί ένα script, δεν αποτελεί σε καμία περίπτωση ωφέλιμη πληροφορία, όμως ένα κομμάτι κώδικα που εισάγει μία φωτογραφία κατά πάσα πιθανότητα φέρει πληροφορία, όπως ο τίτλος της φωτογραφίας. Σε επόμενο κεφάλαιο θα περιγράψουμε αναλυτικά αλγορίθμους που χρησιμοποιήσαμε για να ελαχιστοποιήσουμε τη πιθανότητα αφαίρεσης χρήσιμης πληροφορίας. Επίσης να αναφέρουμε πως ο υπομηχανισμός αυτός δε προχωρά στη δενδρική δομή (links) μιας ιστοσελίδας, αλλά επεξεργάζεται μόνο τη πληροφορία που περιέχεται στη κεντρική σελίδα ενός ιστότοπου. Τέλος στο τρίτο στάδιο έχοντας το καθαρό κείμενο αφαιρούμε από αυτό όσες λέξεις έχουν μήκος μικρότερο ή ίσο του τέσσερα.

4.4.3 Μηχανισμός Κατηγοριοποίησης

Έχει ήδη αναφερθεί αρκετές φορές ποια είναι η λειτουργία του συγκεκριμένου μηχανισμού. Αξίζει όμως να τονίσουμε κάποια βασικά στοιχεία της λειτουργίας αυτού του μηχανισμού. Ο μηχανισμός αυτός από τη στιγμή που θα αρχικοποιηθεί με ένα σύνολο πρότυπων κειμένων για τη δημιουργία μίας κατηγορίας μπορεί να λειτουργεί ανεξάρτητα από το υπόλοιπο σύστημα κατηγοριοποιώντας συνεχώς κείμενα. Είναι πολύ βασικό για την καλή λειτουργία του συστήματος να υπάρχουν συνεχώς κείμενα προς κατηγοριοποίηση προκειμένου να μη μένει ο μηχανισμός αδρανής. Επιπρόσθετα ο συγκεκριμένος μηχανισμός είναι σε θέση να αναγνωρίσει ένα κείμενο σαν πρότυπο κείμενο κατηγορίας και να το εντάξει στην αλγοριθμική διαδικασία μέσω της οποίας γίνεται η κατηγοριοποίηση.

4.4.4 Γενικά στοιχεία λειτουργικότητας

Προκειμένου να επιτευχθεί η λειτουργικότητα χρησιμοποιήθηκαν για τη διαδικασία εκπαίδευσης πραγματικά κείμενα από το διαδίκτυο και πιο συγκεκριμένα από τα μεγαλύτερα ειδησεογραφικά πρακτορεία. Τα κείμενα αυτά θεωρήθηκαν σαν πρότυπα και αξιόπιστα ώστε να μπορούν να αντιπροσωπεύουν μία κατηγορία. Μετά το πέρας της δημιουργίας των πρότυπων κατηγοριών, ξεκίνησε η ανάκτηση κειμένων προς κατηγοριοποίηση βασισμένη στις πρότυπες κατηγορίες που δημιουργήθηκαν με τη βοήθεια του μοντέλου LSI, σε μία πολύ απλοϊκή του μορφή.

Τα συμπεράσματα που επιθυμούμε από την κατηγοριοποίηση κειμένων είναι τα εξής:

- Να βρούμε σε ποιες κατηγορίες ΔΕΝ ανήκει κάποιο κείμενο.
- Να βρούμε σε ποιες κατηγορίες ανήκει.
- Να βρούμε αν μπορεί κάποιο κείμενο να είναι τόσο αντιπροσωπευτικό για μία κατηγορία ώστε να μπορέσουμε να το χρησιμοποιήσουμε στη διαδικασία εκμάθησης.

Είναι πολύ σημαντικό να μπορέσουμε να αποκλείσουμε κείμενα από κάποιες κατηγορίες. Είναι εξίσου σημαντικό με το να εντάξουμε κάποιο κείμενο σε κάποια κατηγορία. Σε οποιαδήποτε διαδικασία κατηγοριοποίησης είναι ενδιαφέρον να διαθέτουμε θετικά και αρνητικά παραδείγματα ανάλογα με την ένταξη ή όχι ενός κειμένου σε κάποια κατηγορία. Τέλος, σημαντικό κομμάτι της όλης διαδικασίας είναι και η διαδικασία βελτιστοποίησης της απόδοσης του συστήματος που γίνεται μέσω της διαδικασίας μάθησης. Θα πρέπει το σύστημα να είναι σε θέση να βελτιώνεται κάτι το οποίο μπορεί να επιτευχθεί με τη χρήση κειμένων που προσεγγίζουν σε μεγάλο βαθμό μία κατηγορία.

Στη συνέχεια θα παρουσιάσουμε τις τεχνολογίες που έχουμε στη διάθεσή μας και μπορούν να μας φανούν χρήσιμες ώστε να κατασκευάσουμε το μηχανισμό που περιγράφουμε καθώς και την τελική επιλογή μας.

Κεφάλαιο

5

5 ΕΠΙΛΟΓΗ ΤΕΧΝΟΛΟΓΙΩΝ

Η επιλογή της τεχνολογίας που θα ακολουθηθεί κατά την κατασκευή ενός σύνθετου συστήματος είναι εξαιρετικά σημαντική προκειμένου να δημιουργηθεί ένα καθολικό σύστημα το οποίο να είναι ευέλικτο, να υποστηρίζει εύκολα αλλαγές και αναβαθμίσεις, να αποτελείται από υποσυστήματα και τέλος να βασίζεται σε ανοιχτά πρότυπα. Το σύστημα που υλοποιήθηκε είναι σύνθετο καθότι έχει βάση το διαδίκτυο αλλά ένα σημαντικό κομμάτι του, ίσως ο πυρήνας, κρύβεται στο μηχανισμό που πραγματοποιεί προεπεξεργασία κειμένου και γενικότερα διαχείριση πληροφορίας. Οι δύο τελευταίοι μηχανισμοί επεξεργασίας και κατηγοριοποίησης ουσιαστικά δεν έχουν καμία επαφή με το διαδίκτυο και φυσικά δεν είναι και απαραίτητο να έχουν. Βέβαια, τα δεδομένα που δέχονται προέρχονται από εξόρυξη πληροφορίας στο διαδίκτυο (HTML σελίδες).

5.1 Βάση Δεδομένων

Όσον αφορά τη βάση δεδομένων θα πρέπει να επιλεγεί μία τεχνολογία η οποία να είναι πλήρως συμβατή με τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για την υλοποίηση των μηχανισμών ανάκτησης και επεξεργασίας καθώς επίσης και με το μηχανισμό που θα κατηγοριοποιεί τη πληροφορία.

5.1.1 Γιατί MySQL

Η MySQL είναι η δημοφιλέστερη Βάση Δεδομένων ανοιχτού κώδικα που προσφέρεται από το Δίκτυο MySQL. Η αρχιτεκτονική της την κάνουν να είναι εξαιρετικά γρήγορη και πολύ εύκολη σε αλλαγές και αναβαθμίσεις. Επιτρέπει επαναχρησιμοποίηση κώδικα όπου αυτό είναι αναγκαίο και παρέχει ένα μινιμαλιστικό τρόπο δημιουργίας στοιχείων διαχείρισης βάσης δεδομένων τέτοια ώστε να κάνουν τη MySQL ασύγκριτη σε ταχύτητα, σε κατάληψη χώρου, σταθερότητα και ευκολία. Ο μοναδικός στο είδος του διαχωρισμός του κεντρικού πυρήνα του server από το μηχανισμό αποθήκευσης κάνει δυνατή την ύπαρξη αυστηρού ελέγχου σε συναλλαγές και μείωση ταχύτητας ή ύπαρξη θεαματικά μεγάλης ταχύτητας με απευθείας προσπέλαση των δεδομένων στοιχεία που μπορεί να χρησιμοποιηθούν ανάλογα με τις ανάγκες των χρηστών.

Η MySQL περιλαμβάνει αποθήκευση σε μηχανή InnoDB, η οποία υποστηρίζει ασφάλεια στις συναλλαγές και ACID-συμβατή μηχανή αποθήκευσης με commit, rollback, crash recovery και low-level locking δυνατότητες.

Η έκδοση της MySQL που βρίσκεται αυτή τη στιγμή σε σταθερή κατάσταση είναι η 5.1.36 και υποστηρίζει πολλά στοιχεία που αφορούν την απόδοση, τη διεθνοποίηση και τη δυνατότητα ένταξης του MySQL server σε άλλα στοιχεία υλικού και λογισμικού. Τα πιο βασικά στοιχεία που χαρακτηρίζουν τη MySQL είναι:

- Υποερωτήματα, που επιτρέπουν στους χρήστες να κάνουν σύνθετα ερωτήματα με μεγάλη ευκολία και αποδοτικά.
- Γρήγορη επικοινωνία μεταξύ server και client μέσα από ένα καινούριο πρωτόκολλο
- Μικρότερη κατανάλωση πόρων από το server μέσα από βελτιστοποίηση στις βιβλιοθήκες
- Υποστήριξη Unicode, διεθνείς χαρακτήρες και υποστήριξη αποθήκευσης στην πλειοψηφία των συνόλων χαρακτήρων
- Υποστήριξη τύπων GIS για ερωτήματα που αφορούν χάρτες και γεωγραφικά δεδομένα

Τα παραπάνω στοιχεία κάνουν τη MySQL ένα υπερπολύτιμο εργαλείο στα χέρια κάποιου χρήστη και τη θέτουν στην 1η θέση για επιλογή ως βάση δεδομένων του συστήματός μας. [19]

5.1.2 Γιατί PostgreSQL

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων βασισμένη στα αντικείμενα. Ουσιαστικά προέρχεται από την POSTGRES, V 4.2, που έχει δημιουργηθεί στο πανεπιστήμιο της Καλιφόρνια στο τμήμα Επιστήμης των Υπολογιστών του Μπέρκλεϋ. Μάλιστα το συγκεκριμένα σύστημα υλοποίησε πολλές λειτουργικότητες πολλά χρόνια πριν εφαρμοστούν στα πιο γνωστά από τα σημερινά συστήματα βάσεων δεδομένων.

Η PostgreSQL είναι ένας ανοιχτού κώδικα απόγονος του αρχικού κώδικα που γράφηκε στο Μπέρκλεϋ. Υποστηρίζει SQL92 και SQL99 και προσφέρει πολλά στοιχεία που υποστηρίζουν οι περισσότερες βάσεις δεδομένων τελευταίας τεχνολογίας όπως:

- Σύνθετα ερωτήματα
- Foreign Keys
- Triggers
- Διαφορετικές όψεις
- Ακεραιότητα στις συναλλαγές
- Συνεργασία ταυτόχρονων πολλαπλών εκδόσεων

Επιπρόσθετα, η PostgreSQL μπορεί να εμπλουτιστεί σε στοιχεία από κάποιον έμπειρο χρήστη με πολλούς τρόπους ώστε να υποστηρίζει νέα:

- Τύπους δεδομένων
- Συναρτήσεις
- Διαχειριστές
- Συναθροιστικές συναρτήσεις
- Μεθόδους ευρετηρίου
- Διαδικασιακές γλώσσες

Τέλος, αξίζει να τονιστεί η γενναιοδωρία της άδειας κάτω από την οποία βρίσκεται η PostgreSQL σύμφωνα με την οποία μπορεί να χρησιμοποιηθεί, αλλαχθεί και διακινηθεί από τον καθένα χωρίς κανένα κόστος. [20]

5.2 Τεχνολογία Μηχανισμού Ανάκτησης και Επεξεργασίας

Οι μηχανισμοί ανάκτησης και επεξεργασίας είναι τα δύο συστήματα τα οποία αναλαμβάνουν τη πιο σημαντική εργασία του συστήματός μας. Βασικές τους λειτουργίες είναι η επικοινωνία με τον εξυπηρετητή (server) που βρίσκεται ο κώδικας κάθε ιστοσελίδας και η διαχείριση και επεξεργασία κειμένου. Το βασικό ερώτημα εδώ είναι αν θα χρησιμοποιηθεί κάποια αντικειμενοστραφής γλώσσα ή μία γλώσσα διαδικαστική. Εδώ να παρατηρήσουμε πως τα περισσότερα σύγχρονα σχετικά συστήματα έχουν υλοποιηθεί σε αντικειμενοστραφή γλώσσα προγραμματισμού.

5.2.1 Γιατί C

Η επιλογή της C μπορεί να γίνει για ένα σύνολο από λόγους μεταξύ των οποίων είναι οι εξής: Η C μπορεί να χρησιμοποιηθεί σαν χαμηλού επιπέδου γλώσσα προγραμματισμού επιτρέποντας άμεση πρόσβαση στους πόρους του υπολογιστή και άρα στην αποτελεσματική και χωρίς overhead αξιοποίησή τους. Εξάλλου, είναι η καθιερωμένη γλώσσα για χαμηλού επιπέδου προγραμματισμό που ένας μηχανικός θα απαιτηθεί να κάνει για την καλύτερη αξιοποίηση του υλικού που σχεδιάζει και αναπτύσσει. Ταυτόχρονα, μπορεί να χρησιμοποιηθεί και σαν γλώσσα υψηλού επιπέδου καθώς η πληθώρα των διαθέσιμων βιβλιοθηκών υπερκαλύπτουν τις απαιτήσεις ανάπτυξης λογισμικού επιπέδου εφαρμογής (Application Layer Software). Επίσης είναι σχετικά μικρή και εύκολη στην εκμάθηση, υποστηρίζει top-down και modular σχεδιασμό, υποστηρίζει δομημένο (structured) προγραμματισμό και είναι αποτελεσματική (efficient) αφού παράγει συμπαγή και γρήγορα στην εκτέλεση προγράμματα. Ακόμα είναι φορητή (portable), ευέλικτη (flexible), ισχυρή (powerful), δε βάζει περιορισμούς, γεγονός που συχνά αποβαίνει σε βάρος της και αποτελεί με τη C++ την ευρύτερα χρησιμοποιούμενη γλώσσα σε ερευνητικά και αναπτυξιακά προγράμματα. Να αναφέρουμε ακόμα ότι υπάρχει μία πολλή μεγάλη εγκατεστημένη βάση εφαρμογών που αναπτύχθηκαν με τη γλώσσα αυτή και πρέπει να συντηρούνται και να εξελίσσονται και τέλος η γνώση της C αποτελεί ένα πολύ καλό εφόδιο για την εκμάθηση της Java καθώς αυτή υιοθετεί το μεγαλύτερο ποσοστό των δομικών στοιχείων της C. [21]

5.2.2 Γιατί C++

Πρόκειται μία γλώσσα προγραμματισμού που δημιουργήθηκε ως κύριος αντίπαλος της Java και προφανώς υποστηρίζει αντικειμενοστραφή προγραμματισμό. Από το 1998 το C++ Standard αποτελείται από δύο κομμάτια: ο πυρήνας και οι βασικές βιβλιοθήκες. Η τελευταία έκδοση περιέχει βασικές βιβλιοθήκες της C++ και ένα μεγάλο κομμάτι από τις βασικές βιβλιοθήκες της C. Παράλληλα υπάρχουν πολλές βιβλιοθήκες που έχουν συγκεκριμένους σκοπούς και επικεντρώνονται σε συγκεκριμένα στοιχεία και δεν περιλαμβάνονται στις Standard βιβλιοθήκες.

Αξιοσημείωτο είναι και το γεγονός ότι είναι σχετικά απλό να ενταχθούν βιβλιοθήκες της C μέσα σε προγράμματα γραμμένα σε C++.

Είναι πολύ σημαντικό να γίνει κατανοητό, πως δεν υπάρχει πλέον μία μοναδική γλώσσα που να ονομάζεται C++. Ο όρος αντιπροσωπεύει μία οικογένεια παρόμοιων γλωσσών οι οποίες είναι συχνά υπό- ή υπέρ- σύνολα μεταξύ τους.

Βασικά στοιχεία της C++ περιλαμβάνουν δηλώσεις, function-like casts, inline functions, function overloading, classes, exception handling κ.α. Η C++ συνήθως πραγματοποιεί μεγαλύτερο έλεγχο τύπων σε μεταβλητές απ' ότι η C. Πολλά στοιχεία της C++ τα υιοθέτησε και η C ωστόσο η C99 παρουσίασε πολλά στοιχεία που δεν υιοθετήθηκαν ούτε και υπάρχουν στην C++. Μία πολύ συνηθισμένη πηγή σύγχυσης είναι το ζήτημα ορολογίας: εξαιτίας της παραγωγής από τη C, στη C++ ο όρος αντικείμενο σημαίνει περιοχή μνήμης, όπως και στη C, και όχι ένα class instance, κάτι το οποίο συμβαίνει στις περισσότερες γλώσσες προγραμματισμού. [22]

5.2.3 Γιατί Java

Αντίστοιχα, η επιλογή της Java μπορεί να γίνει για ένα σύνολο από λόγους μεταξύ των οποίων είναι οι εξής: Αναπτύχθηκε κατ' αρχήν ως γλώσσα για ανάπτυξη ενσωματωμένου λογισμικού (embedded software) και καλύπτει τις αντίστοιχες ανάγκες ενός Μηχανικού συστημάτων. Είναι φορητή, γεγονός που διασφαλίζει τη δυνατότητα εκτέλεσης των Java προγραμμάτων ανεξάρτητα πλατφόρμας υλικού και λογισμικού. Επίσης διαθέτει πολύ μεγάλη βιβλιοθήκη έτοιμων κλάσεων, οι οποίες διευκολύνουν σε μεγάλο βαθμό τη γρήγορη ανάπτυξη αξιόπιστων εφαρμογών και γνωρίζει ραγδαία εξάπλωση σε ερευνητικά και αναπτυξιακά προγράμματα. Ακόμα μπορεί να χρησιμοποιηθεί για προγραμματισμό στο διαδίκτυο και όσον αφορά την υποστήριξη της Αντικειμενοστραφούς Προσέγγισης είναι πολύ πιο καθαρή από τη C++ και έτσι θα μπορούσε να θεωρηθεί σαν λογική συνέχεια της C. Τέλος υιοθετεί μεγάλο μέρος της C.

Η Java παρουσιάστηκε σαν μία γλώσσα που είχε αφαιρέσει τα «βρώμικα» στοιχεία της C++ και είχε εισάγει ένα σύνολο από καλά στοιχεία άλλων γλωσσών όπως η Smalltalk. Η ιστορία της γλώσσας ξεκίνησε όταν μία ομάδα ερευνητών στην προσπάθειά της να αναπτύξει ενσωματωμένο λογισμικό (embedded software) για έξυπνες καταναλωτικές συσκευές στα πλαίσια του project Green, αποφάσισε να αναπτύξει μία νέα γλώσσα μετά τη διαπίστωσή της ότι η C και η C++ δεν ανταποκρίνονται στις απαιτήσεις της. Έτσι τον Αύγουστο του 1991 εμφανίστηκε μία νέα αντικειμενοστραφής γλώσσα με το όνομα OAK, που είναι το ακρωνύμιο του Object Application Kernel. Η γλώσσα απλά προστέθηκε στον κατάλογο των καλών γλωσσών προγραμματισμού με ουσιαστική υποστήριξη σε εφαρμογές τύπου πελάτη-εξυπηρετητή (client-server) και τίποτα παραπάνω.

Μόλις τον Απρίλιο του 1993 έκανε την εμφάνισή του το NCSA MOSAIC 1.0 ως πρώτο γραφικό πρόγραμμα πλοήγησης στο διαδίκτυο (Web browser) και έτσι η γλώσσα άρχισε να κάνει τα πρώτα της βήματα στο χώρο του διαδικτύου με πολύ θετικά αποτελέσματα. Το στοιχείο αυτό ώθησε τη Sun, μετά από μία αποτυχημένη προσπάθειά της να πουλήσει τη γλώσσα (Αύγουστος 93), να χρηματοδοτήσει την ανάπτυξή της για το 1994, αν και το προηγούμενο έτος είχε διακόψει ως μη επιτυχημένο το αντίστοιχο project. Στα μέσα του 1994, αναπτύχθηκε το πρώτο πειραματικό πρόγραμμα πλοήγησης με Java κάτω από το όνομα του WebRunner. Το

φθινόπωρο του ίδιου έτους, ο Van Hoff υλοποιεί με Java τον πρώτο Java διερμηνευτή.

Μόλις τον Ιανουάριο του 1995, η γλώσσα πήρε τη σημερινή της ονομασία και εμφανίστηκε η πρώτη επίσημη τεκμηρίωσή της με τη μορφή ενός "white paper". Το Μάιο του ίδιου έτους, η Sun παρουσιάζει επίσημα τη Java και το HotJava. Ταυτόχρονα, η Netscape αγόρασε άδεια χρήσης της Java και ενσωμάτωσε τη γλώσσα στη δεύτερη έκδοση του Netscape, του γνωστού προγράμματος πλοήγησης. Στη συνέχεια, ο ένας μετά τον άλλο, οι μεγάλοι κατασκευαστές λογισμικού ανακοίνωσαν την απόφασή τους να χρησιμοποιήσουν τη Java, με αποκορύφωμα την απόφαση της Microsoft το Δεκέμβρη του 1995. Η Java καθιερώθηκε πια ως η γλώσσα που θα πρωτοστατήσει στην ερχόμενη δεκαετία. Μία αναλυτική αναφορά στο χρονικό της εξέλιξης της γλώσσας μπορείτε να βρεθεί στο [23].

5.2.4 Γιατί Perl

Η Perl είναι μια γενικού σκοπού γλώσσα προγραμματισμού που αρχικά δημιουργήθηκε για την επεξεργασία κειμένου και τώρα χρησιμοποιείται σε μια πλειάδα συστημάτων, συμπεριλαμβανομένων των συστημάτων διαχείριση, ανάπτυξη συστημάτων δικτύου, δικτυακός προγραμματισμούς, ανάπτυξη GUI και άλλα.

Η γλώσσα αυτή σκοπεύει να είναι απλή, αποδοτική και τέλεια παρά «όμορφη». Τα κύρια στοιχεία της είναι η ευκολία στη χρήση, η υποστήριξη διαδικασιακού και αντικειμενοστραφή προγραμματισμού και παράλληλα υποστηρίζει πολύ ισχυρούς μηχανισμούς επεξεργασίας κειμένου.

Η γενικότερη δομή της προέρχεται κυρίως από τη γλώσσα προγραμματισμού C. Είναι μια διαδικασιακή γλώσσα προγραμματισμού που χρησιμοποιεί μεταβλητές, παραστάσεις, αποδόσεις, μπλοκ κώδικα, συναρτήσεις ελέγχου και υπορουτίνες.

Λαμβάνει υπόψη της τον προγραμματισμό σε shell και τα προγράμματα σε perl είναι μεταφραζόμενα. Όλες οι μεταβλητές διαχωρίζονται με ένα συγκεκριμένο χαρακτηριστικό που προηγείται αυτών, επιτρέποντας έτσι καλύτερη σύνταξη. Όπως και το shell του UNIX, η Perl έχει πολλές έτοιμες συναρτήσεις οργανωμένες σε βιβλιοθήκες που αναλαμβάνουν τις περισσότερες απλές εργασίες όπως ταξινόμηση ή διασύνδεση με λειτουργίες του συστήματος.

Η Perl χρησιμοποιεί συσχετιζόμενους πίνακες από το awk και «κανονικές εκφράσεις» από το sed. Αυτά τα στοιχεία απλοποιούν την ανάλυση λέξεων, τη διαχείριση κειμένου και τη διαχείριση δεδομένων.

Στην έκδοση 5 της perl, προστέθηκαν στοιχεία για να υποστηρίζουν σύνθετους τύπους δεδομένων και δομές δεδομένων καθώς επίσης και μοντέλα αντικειμενοστραφούς προγραμματισμού.

Σε όλες τις εκδόσεις της perl ο τύπος δεδομένων μίας μεταβλητής βρίσκεται αυτόματα, ενώ αυτόματη είναι και η διαχείριση της μνήμης. Ο μεταφραστής γνωρίζει τον τύπο και τις απαιτήσεις σε αποθηκευτικό χώρο για κάθε τύπο του προγράμματος. Καθορίζει το χώρο που θα καταλαμβάνει κάθε πρόγραμμα και απελευθερώνει πόρους όποτε αυτό είναι εφικτό. Επιτρεπόμενες μετατροπές μεταξύ τύπων γίνονται αυτόματα.

Τα παραπάνω βέβαια σημαίνουν ότι δεν επιτρέπονται διαρροές στη μνήμη, σταμάτημα του μεταφραστή ή να διακοπεί η αναπαράσταση των εσωτερικών δεδομένων [24].

5.3 Τελική επιλογή τεχνολογιών

Η τελική επιλογή τεχνολογιών όπως αναφέρθηκε και στην αρχή του κεφαλαίου βασίζεται στο γεγονός ότι η υλοποίηση βασίζεται σε συνδυασμό πολλών μηχανισμών οι οποίοι είτε επικοινωνούν άμεσα μεταξύ τους είτε μέσω της βάσης δεδομένων. Για αυτό το λόγο θα πρέπει να εξασφαλίσουμε τη πλήρη συμβατότητα της γλώσσας προγραμματισμού με τη τεχνολογία της βάσης δεδομένων, λαμβάνοντας όμως υπόψη και τις λειτουργίες που μας προσφέρει η κάθε γλώσσα προγραμματισμού για επεξεργασία κειμένου και επικοινωνία με το διαδίκτυο.

Συνεπώς καταλήγουμε σε γλώσσα βάσης δεδομένων MySQL γιατί επιθυμούμε απλότητα, σταθερότητα και αξιοπιστία και σε αντικειμενοστραφή γλώσσα προγραμματισμού Java με υποστήριξη βάσης δεδομένων MySQL προκειμένου να γίνονται όλες οι διαδικασίες που χρειάζονται εκτενείς αναλύσεις και υπολογισμούς.

Κεφάλαιο

6

6 ΜΕΛΕΤΩΝΤΑΣ ΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στο παρόν κεφάλαιο θα αναπτυχθούν θέματα που αφορούν τις διαδικασίες που πραγματοποιούνται στο σύστημα με σκοπό να γίνει σαφής ο τρόπος με τον οποίο έχει δομηθεί ο μηχανισμός του. Παράλληλα, θα περιγραφούν στοιχεία στα οποία θα πρέπει να βασιστούν οι διαδικασίες του συστήματος προκειμένου να γίνει αποδοτικό και εύχρηστο. Το σημείο στο οποίο θέλουμε να καταλήξουμε είναι οι βασικές αρχές που πρέπει να διέπουν συνολικά ένα τόσο εκτεταμένο σύστημα προκειμένου να ολοκληρωθούν με επιτυχία όλες οι διαδικασίες που πραγματοποιούνται.

6.1 Γενικές Αρχές και Πρότυπα

Οι παρακάτω ενότητες αναφέρονται στις βασικές αρχές πάνω στις οποίες θα στηριχθεί το σύστημα προκειμένου να είναι ευέλικτο και αποδοτικό.

6.1.1 Καλώς ορισμένη Βάση Δεδομένων

Έχει αναφερθεί σε προηγούμενες ενότητες πως το σύστημα θα αποθηκεύει πληθώρα πληροφορίας προκειμένου να είναι σε θέση να κάνει εκτενή ανάλυση των δεδομένων που διαθέτει. Προκειμένου να υπάρχει καθαρή οργάνωση της πληροφορίας και σαφής προσδιορισμός της ανά πάσα στιγμή, θα πρέπει το σύστημα να διαθέτει μια καλώς ορισμένη βάση δεδομένων. Το γεγονός αυτό, αυτομάτως σημαίνει πως πρέπει να δοθεί ιδιαίτερο βάρος κατά την επιλογή της βάσης δεδομένων που θα χρησιμοποιηθεί αλλά και κατά το σχεδιασμό των πινάκων της. Συνεπώς θα πρέπει να έχουμε εκ των προτέρων γνώση των στοιχείων που είναι απαραίτητο να αποθηκευτούν στη βάση δεδομένων, γεγονός που γεννά την ανάγκη για μία μικρή περίοδο δοκιμών και παράλληλα σχεδιασμό της βάσης με τέτοιο τρόπο ώστε να είναι εφικτές τυχόν αλλαγές που μπορεί να χρειαστεί να πραγματοποιηθούν στους πίνακές της.

6.1.2 Κώδικας βασισμένος σε διεθνή στάνταρ

Η επιτυχία ενός συστήματος εξαρτάται, συν τοις άλλοις, και στα πρότυπα τα οποία ακολουθεί και στα στάνταρ στα οποία βασίζεται. Είναι πρωταρχικής σημασίας

να μπορεί ένα σύστημα να παρέχει στους μελλοντικούς χρήστες τα εχέγγυα καλής λειτουργίας σε οποιοδήποτε λειτουργικό σύστημα και κάτω από οποιεσδήποτε συνθήκες.

Για το λόγο αυτό υπάρχουν και οργανισμοί οι οποίοι κατά καιρούς ανακοινώνουν τα πιο πρόσφατα διεθνή πρότυπα τα οποία προφανώς δεν επιβάλλονται, αλλά κρίνονται πως είναι αναγκαία για την καλή παγκόσμια λειτουργία ενός συστήματος.

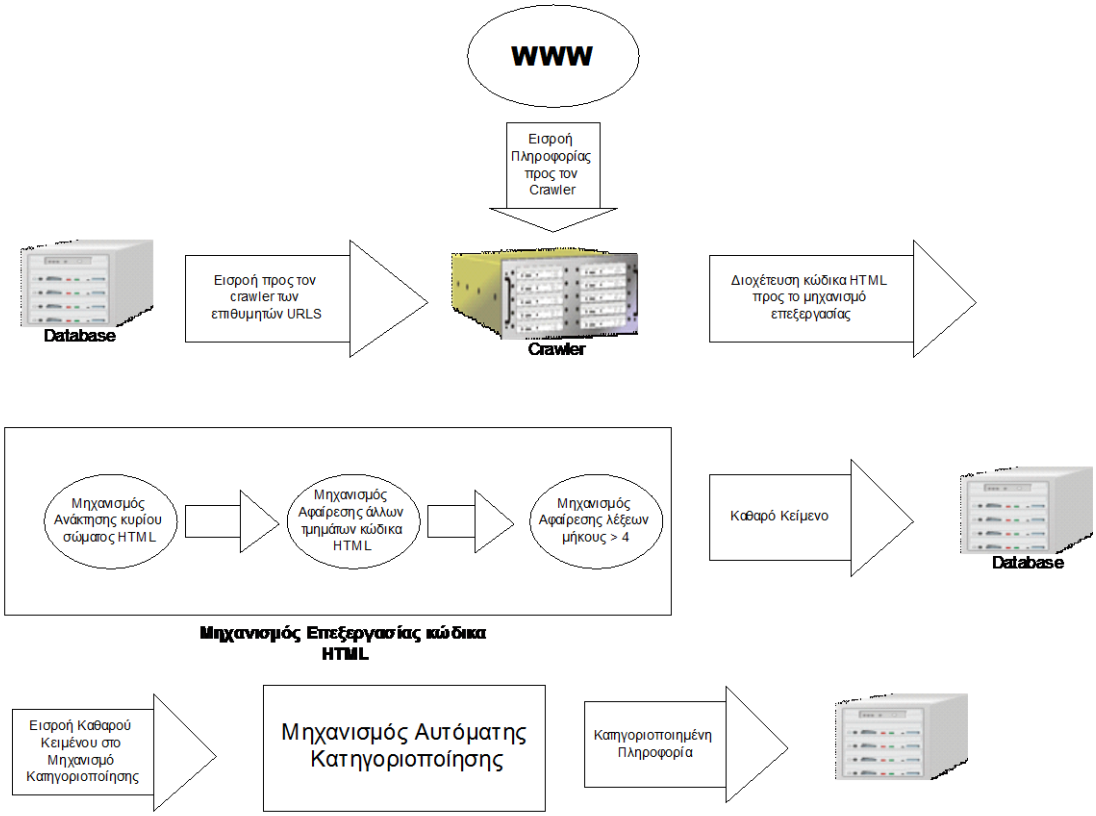
Επιπρόσθετα, να τονίσουμε πως ο κώδικας που δημιουργούμε είναι ανοιχτός, διατίθεται σε όποιον τον ζητήσει και προφανώς επιδέχεται πλειάδα βελτιώσεων. Θεωρούμε πως τα προγράμματα ανοιχτού κώδικα είναι θεωρητικά πιο κοντά στους χρήστες διότι δίνεται η δυνατότητα σε κάθε προγραμματιστή, που έρχεται σε επαφή με τον κώδικα, να κάνει τις δικές του βελτιώσεις και να προσαρμόσει το πρόγραμμα στα δικά του μέτρα.

6.2 Ροές Εργασιών

Είναι κατανοητό πως οι αρχές και τα πρότυπα που περιγράφονται παραπάνω δε μπορούν να επηρεάσουν τη ροή της εργασίας καθώς η ροή της εργασίας δεν είναι τίποτα περισσότερα από τον αλγόριθμο με τον οποίο πετυχαίνουμε τους στόχους του συστήματός μας. Ωστόσο, θα μπούμε και στη διαδικασία να αναλύσουμε πως είναι εφικτό να πραγματοποιηθούν τα διαφορετικά κομμάτια που παρουσιάζονται στο διάγραμμα ροής με τέτοιον τρόπο ώστε οι διαδικασίες που πραγματοποιούνται να ακολουθούν συγκεκριμένες αρχές και μοτίβα που θα κάνουν συνολικά το σύστημα φιλικότερο προς το χρήστη.

6.2.1 Διάγραμμα Ροής Πληροφορίας

Στο παρακάτω σχεδιάγραμμα φαίνεται η πορεία που ακολουθεί η πληροφορία από τη στιγμή που θα αναγνωστεί από το μηχανισμό ανάκτησης πληροφορίας από το Διαδίκτυο μέχρι τη στιγμή που θα καταλήξει στον τελευταίο πίνακα της βάσης δεδομένων του συστήματός μας, που περιέχει τη πληροφορία κατηγοριοποιημένη.



Εικόνα 6. Ροή πληροφορίας στο σύστημά μας

Σύμφωνα με το παραπάνω σχεδιάγραμμα, το σύστημα ξεκινά ανακτώντας από τη βάση δεδομένων τις διευθύνσεις των ιστοσελίδων προς επεξεργασία και κατηγοριοποίηση, τις οποίες εισάγει ο χρήστης χειροκίνητα στη ΒΔ. Στη συνέχεια εγκαθιδρύεται σύνδεση με το server στον οποίο βρίσκεται αποθηκευμένος ο κώδικας κάθε ιστοσελίδας και το σύστημα προχωρά στην ανάκτηση του κώδικα HTML της σελίδας, ο οποίος αποθηκεύεται δυναμικά σε μεταβλητή με τη μορφή κειμένου. Ακολουθώντας το διάγραμμα ροής το κείμενο που περιέχει το κώδικα διοχετεύεται στο μηχανισμό επεξεργασίας του συστήματος, ο οποίος εκτελεί τις παρακάτω διακριτές λειτουργίες πάνω στο κείμενο:

- Εξαγωγή του σώματος (body) του κώδικα HTML
- Αφαίρεση τμημάτων που δεν αποτελούν ωφέλιμη πληροφορία από το κώδικα
- Αφαίρεση όσων λέξεων έχουν μήκος μικρότερο ή ίσο του τέσσερα

Μετά την ολοκλήρωση των παραπάνω λειτουργιών το «καθαρό» πλέον κείμενο αποθηκεύεται στη ΒΔ του συστήματος. Στο τρίτο και τελευταίο στάδιο ο μηχανισμός

κατηγοριοποίησης δέχεται σαν είσοδο από τη ΒΔ το επεξεργασμένο κείμενο κάθε ιστοσελίδας και παρέχει στην έξοδό του τη διεύθυνση της κάθε σελίδας, τη κατηγορία που ανήκει αυτή καθώς και τη συσχέτισή της με τη συγκεκριμένη κατηγορία, πληροφορίες που τελικά καταλήγουν στη ΒΔ.

6.3 Βασικοί Μηχανισμοί Συστήματος

Στη συγκεκριμένη ενότητα θα πραγματοποιηθεί διεξοδική ανάλυση της λειτουργίας των τριών μηχανισμών που υλοποιούν το σύστημά μας.

6.3.1 Ανάλυση Μηχανισμού Ανάκτησης Πληροφορίας

Όπως έχει αναφερθεί ήδη ο μηχανισμός εξόρυξης πληροφορίας είναι ένα σύστημα που αναπτύχθηκε με στόχο την ταχύτητα. Συγκεκριμένα αφού ανακτήσει από τη ΒΔ του συστήματος τις διευθύνσεις (urls) των ιστοσελίδων που επιθυμούμε να επεξεργαστούμε, πραγματοποιεί σύνδεση ώστε να αποκτήσει πρόσβαση στο server που βρίσκεται αποθηκευμένος ο κώδικας HTML της εκάστοτε ιστοσελίδας, τον οποίο ανακτά και αποθηκεύει δυναμικά σε μία μεταβλητή τύπου StringBuffer. Για το σκοπό αυτό χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες της java: java.io.*, java.net.*. Επίσης σε αυτό το σημείο θα πρέπει να αναφέρουμε πως προτιμήσαμε να αποθηκεύουμε το κώδικα, ως κείμενο σε μεταβλητή τύπου StringBuffer αντί String. Η επιλογή μας αυτή βασίζεται στις συναρτήσεις που προσφέρει η συγκεκριμένη κλάση της Java, οι οποίες είναι κατάλληλες για την ανάκτηση πληροφορίας που εκτελούμε αλλά και για την επεξεργασία αυτής στη συνέχεια.

6.3.2 Ανάλυση Μηχανισμού Επεξεργασίας Πληροφορίας

Ο μηχανισμός επεξεργασίας δέχεται τη μεταβλητή που περιέχει το κείμενο με το κώδικα HTML και έχει σαν στόχο να το φέρει σε κατάλληλη μορφή για να δοθεί ως είσοδο στο μηχανισμό κατηγοριοποίησης. Όπως αναφέραμε και προηγουμένως ο μηχανισμός εκτελεί κάποιες λειτουργίες πάνω στο κείμενο τις οποίες θα δούμε διεξοδικότερα στη συνέχεια. Το πρώτο βήμα είναι η μετατροπή όλου του κειμένου σε πεζούς χαρακτήρες (lowercase). Επειδή πολλά περιβάλλοντα ανάπτυξης προγραμμάτων είναι case sensitive (αντιλαμβάνονται διαφορετικά τους μικρούς και τους κεφαλαίους χαρακτήρες), κρίνεται αναγκαίο να μεταφερθούν οι χαρακτήρες σε μία μορφή και επιλέγεται αυτή να είναι οι πεζοί χαρακτήρες. Στο δεύτερο βήμα απομονώνουμε το σώμα του κώδικα το οποίο βρίσκεται ανάμεσα στις ετικέτες <body> που δηλώνει την έναρξη και </body> που δηλώνει τη λήξη του σώματος ενός κώδικα HTML. Δηλαδή ανακτούμε μόνο το δεύτερο βασικό τμήμα του κώδικα καθώς γνωρίζουμε πως στο πρώτο τμήμα δε περιέχεται καθόλου ωφέλιμη πληροφορία. Στο τρίτο βήμα σκοπός μας είναι να αφαιρέσουμε τα κομμάτια κώδικα που υλοποιούν κάποιες πρόσθετες λειτουργίες μίας ιστοσελίδας και σε καμία περίπτωση δε περιέχουν ωφέλιμη πληροφορία. Ομοίως με το δεύτερο βήμα η έναρξη εκτέλεσης ενσωματωμένου κώδικα σε κώδικα HTML δηλώνεται με την ετικέτα <script> και η λήξη με την ετικέτα </script>. Επίσης σε αυτό το βήμα αφαιρούμε και κομμάτια κώδικα που υλοποιούν θέματα μορφοποίησης μίας ιστοσελίδας. Αντίστοιχα η έναρξή

τους δηλώνεται με την ετικέτα <style> και η λήξη τους με την ετικέτα </style>. Στο τέταρτο βήμα αφαιρούμε όλες τις υπόλοιπες ετικέτες. Σε αυτό το βήμα αφαιρούμε μόνο τις ετικέτες εντοπίζοντας τη θέση του συμβόλου < και > καθώς βάση της σύνταξης της γλώσσας HTML ανάμεσα σε αυτές τις ετικέτες υπάρχει διαθέσιμο ωφέλιμο κείμενο. Στο πέμπτο βήμα αφαιρούμε σημεία στίξης και οποιουσδήποτε χαρακτήρες μπορεί να αλλοιώνουν τη μορφή του κειμένου. Στο τελευταίο βήμα κάνοντας χρήση της ενσωμάτωσης στη Java των κανονικών εκφράσεων (regular expressions) αφαιρούμε όσες λέξεις έχουν μήκος μικρότερο ή ίσο του τέσσερα. Αυτό το βήμα, χρησιμοποιείται σπανίως με τις ερευνητικές απόψεις να είναι διχασμένες για το αν προσφέρει ποιοτικότερα αποτελέσματα ή όχι. Για την υλοποίηση αυτών των βημάτων χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες της Java: java.text.* και java.util.regex.*. Με την ολοκλήρωση των παραπάνω βημάτων έχουμε στην έξοδο του μηχανισμού το «καθαρό» κείμενο κάθε ιστοσελίδας. Παρακάτω δίνεται ένα παράδειγμα από την ιστοσελίδα του ειδησεογραφικού πρακτορείου CNN[25], όπου στο πίνακα 1 παρουσιάζεται τμήμα του καθαρού κειμένου όπως αποθηκεύεται στη ΒΔ από την έξοδο του μηχανισμού επεξεργασίας και δίνεται σαν είσοδο στο μηχανισμό κατηγοριοποίησης.

Εικόνα 7. Τμήμα από την ιστοσελίδα του CNN

Πίνακας 1. Κείμενο προερχόμενο από την επεξεργασία περιεχομένου του ειδησεογραφικού πρακτορείου CNN

videos world politics crime entertainment health travel living money sports video ireport impact topics annie afghanistan latino america commentary topics raquo edition international updated september director roman polanski arrested filmmaker roman polanski arrested arrest warrant stemming decades charge swiss police today academy award winning director pleaded guilty single count having unlawful sexual intercourse minor acknowledging united states before could sentenced story tumultuous polanski always spotlight include virtual element ireport secret inside hidden where producers might travel living taliban unheard father deployed squadron soldiers travel leisure sexiest affordable travel spots malaysia martinique style authenticity affordability latest brewpub makes

6.3.3 Ανάλυση Μηχανισμού Κατηγοριοποίησης Πληροφορίας

Αφού πραγματοποιηθούν όλα τα παραπάνω βήματα τα κείμενα είναι πλέον έτοιμα να κατηγοριοποιηθούν. Ο τρόπος με τον οποίο κατηγοριοποιούνται βασίζεται στη θεωρία LSI και ουσιαστικά είναι μία μικρή παραλλαγή της. Προκειμένου να γίνει κατανοητό πως κατηγοριοποιούμε ένα κείμενο θα αναλύσουμε το μοντέλο πάνω στο οποίο βασιζόμαστε.

Ας φανταστούμε λοιπόν κάθε κατηγορία να είναι ένα σημείο στο χώρο όπου οι άξονες είναι οι λέξεις-κλειδιά και το διάνυσμα που αντιπροσωπεύουν είναι η συχνότητα των λέξεων μέσα στην κατηγορία. Προκειμένου να κατηγοριοποιήσουμε ένα κείμενο θα πρέπει να βρούμε τη γωνία που σχηματίζει το διάνυσμα του κειμένου με το διάνυσμα της κατηγορίας. Όσο μικρότερη είναι η γωνία τόσο πιο κοντά είναι το κείμενο σε μία κατηγορία.

Η γωνία μεταξύ του κειμένου και της κατηγορίας θα πρέπει να βρεθεί χρησιμοποιώντας τις κοινές λέξεις που υπάρχουν για να είναι κανονικοποιημένα τα διανύσματα. Συνεπώς η διαδικασία περιλαμβάνει, εύρεση των κοινών λέξεων μεταξύ κειμένου και κατηγορίας και εν συνεχεία υπολογισμό βάση του παρακάτω τύπου.

$$a = \frac{\vec{q}\vec{v}}{|\vec{q}||\vec{v}|}$$

Ο συγκεκριμένος τύπος φράσσεται τόσο από κάτω όσο και από πάνω. Αυτό εξάγεται ως εξής. Αν σκεφτούμε ότι το q είναι το διάνυσμα του κειμένου τότε η μορφή του θα είναι ως εξής q {school, teacher, pennsylvania, …} = {2, 3, 1, …}. Το v διάνυσμα που εκπροσωπεί το διάνυσμα της κατηγορίας θα πρέπει να είναι

κανονικοποιημένο πάνω στις λέξεις με την έννοια ότι θα πρέπει να περιέχει τις ίδιες λέξεις με αυτές του q ακόμα κι αν η συχνότητα με την οποία εμφανίζονται στην κατηγορία είναι μηδενική. Συνεπώς το διάνυσμα v θα είναι της μορφής v {school, teacher, pennsylvania, …} = {22, 13, 0, …}. Εφαρμόζοντας τα συγκεκριμένα διανύσματα στον παραπάνω τύπο έχουμε σαν αποτέλεσμα.

q.v = 2.22 + 3.13 + 1.0 = 83 (1)

|q| = 3.75 (2)

|v| = 25.55 (3)

a = (1) / [(2). (3)] = 0.86 (4)

Όπως βλέπουμε από το παραπάνω αποτέλεσμα η συσχέτιση μεταξύ της κατηγορίας που αντιπροσωπεύεται από το v και του κειμένου που αντιπροσωπεύεται από το q είναι 0.86 με τη γωνία που σχηματίζεται να είναι 30°, κάτι το οποίο σημαίνει πως υπάρχει μεγάλη συσχέτιση του κειμένου με την κατηγορία. Προφανώς θα πρέπει να έχουμε τα διανύσματα όλων των κατηγοριών και να βρούμε τις γωνίες που σχηματίζει το κείμενο με κάθε κατηγορία.

Η πειραματική διαδικασία μας έδειξε πως αν κάποιο κείμενο ξεπερνά σε συσχέτιση το 0.6 τότε το κείμενο μπορεί να θεωρηθεί πρότυπο για την κατηγορία. Αν η συσχέτιση ξεπερνά το 0.5 τότε το κείμενο ανήκει στη συγκεκριμένη κατηγορία ενώ για συσχέτιση μικρότερη του 0.4 μπορούμε να είμαστε σίγουροι πως το κείμενο δεν ανήκει στην κατηγορία. Ένα πολύ γενικό κείμενο μπορεί να μας δώσει αποτελέσματα για συσχετίσεις που να βρίσκονται μεταξύ 0.2-0.5. Σε αυτή την περίπτωση πρέπει να κάνουμε ένα διαφορετικό υπολογισμό προκειμένου να εντάξουμε ένα κείμενο σε κάποια κατηγορία. Σε μία τέτοια πιθανότητα εργαζόμαστε ως εξής. Βρίσκουμε τη διαφορά μεταξύ της μεγαλύτερης συσχέτισης με τη μικρότερη και διαιρούμε τη διαφορά με τον αριθμό των διαστημάτων (για 7 κατηγορίες έχουμε 6 διαστήματα). Όσες συσχετίσεις είναι μεγαλύτερες του κατωφλιού Μεγαλύτερη_Συσχέτιση – [(Μεγαλύτερη_Συσχέτιση-Μικρότερη_Συσχέτιση) / Αριθμός_Διαστημάτων], τότε το κείμενα εντάσσεται σε αυτή την κατηγορία. Στον παρακάτω πίνακα φαίνεται ένα παράδειγμα το οποίο εξάγεται από ένα πραγματικό κείμενο που λάβαμε από μεγάλο ειδησεογραφικό portal. Το k συμβολίζει το αποτέλεσμα του παραπάνω τύπου

Πίνακας 2. Πίνακας συσχετίσεων

Κατηγορία	Συσχέτιση (Μεγαλύτερη -> Μικρότερη)	Λόγος ένταξης ή όχι (k=0.0362)
1	0.503	Εντάσσεται γιατί η συσχέτιση είναι μεγαλύτερη από 0.5
2	0.482	Εντάσσεται γιατί είναι μεγαλύτερη από Μεγ_Συς – k
3	0.442	Δεν εντάσσεται γιατί είναι μικρότερη από Μεγ_Συς – k
4	0.412	Δεν εντάσσεται γιατί είναι μικρότερη από Μεγ_Συς – k
5	0.387	Δεν εντάσσεται γιατί είναι μικρότερη από 0.4
6	0.376	Δεν εντάσσεται γιατί είναι μικρότερη από 0.4
7	0.286	Δεν εντάσσεται γιατί είναι

		μικρότερη από 0.4

Αυτό που έχουμε να παρατηρήσουμε είναι πως αρχικά τηρούνται οι κανονισμοί που θέλουν κάθε συσχέτιση μεγαλύτερη από 0.5 να εντάσσονται και κάθε συσχέτιση μικρότερη από 0.4 να μην εντάσσονται. Όπως βλέπουμε, όμως, το αποτέλεσμα στην κατηγορία 2 σημαίνει πως το κείμενο θα ενταχθεί και στην κατηγορία 2 διότι σύμφωνα με τον τύπο που περιγράφηκε στην προηγούμενη παράγραφο πληροί τις προϋποθέσεις για να μπει στη συγκεκριμένη κατηγορία. Να επισημανθεί πως αν ο τύπος μας έδινε σαν αποτέλεσμα πως θα πρέπει να ενταχθεί και ένα κείμενο με συσχέτιση μικρότερη του 0.4 τότε αυτό δε θα γινόταν διότι ο κανόνας που επιτάσσει τα κείμενα με συσχέτιση μικρότερη του 0.4 να μην εντάσσονται στην κατηγορία είναι πιο ισχυρός. Στον παρακάτω πίνακα φαίνονται οι κανόνες που ακολουθούνται σύμφωνα με τη σειρά με την οποία πρέπει να πραγματοποιηθούν. Όποιος κανόνας πραγματοποιηθεί σταματά να εκτελείται ο βρόγχος και οι επόμενοι κανόνες δεν ελέγχονται.

Πίνακας 3. Κανόνες Συσχετίσεων

#	Κανόνας	Αποτέλεσμα
1	$\alpha > 0.6$	Ένταξη στην κατηγορία, ένταξη στα πρότυπα κείμενα
2	$\alpha < 0.4$	Καμία συσχέτιση με τη συγκεκριμένη κατηγορία
3	$\alpha > 0.5$	Ένταξη στην κατηγορία
4	$\alpha > \alpha_{max}-k$	Ένταξη στην κατηγορία

Όπου με a συμβολίζεται η συσχέτιση και με k η διαφορά της μεγαλύτερης με τη μικρότερη συσχέτιση διαιρεμένη με τον αριθμό των διαστημάτων.

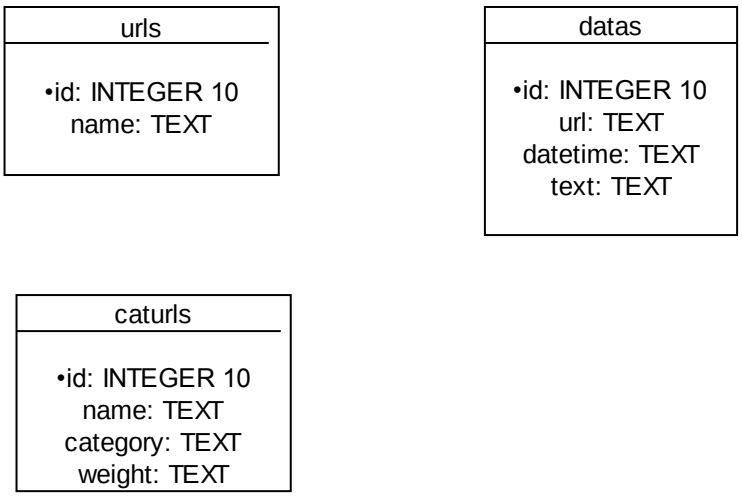
Κεφάλαιο 7

7 ΘΕΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

Στο παρόν κεφάλαιο θα περιγραφούν ζητήματα που αφορούν το σχεδιασμό της βάσης δεδομένων του συστήματος, την επικοινωνία της ΒΔ με τους μηχανισμούς του συστήματος αλλά και την υλοποίηση των βασικότερων αλγορίθμων οι οποίοι παίζουν σημαντικό ρόλο στην καλή λειτουργία του συστήματος.

7.1 Η βάση δεδομένων του συστήματος

Η βάση δεδομένων του συστήματός μας αποτελείται από 3 πίνακες, από τους οποίους ο πρώτος χρησιμοποιείται για τις διαδικασίες της ανάκτησης πληροφορίας, ο δεύτερος από το μηχανισμό επεξεργασίας πληροφορίας και κατηγοριοποίησης και ο τρίτος για την αποθήκευση των αποτελεσμάτων που προέρχονται από το μηχανισμό που κατηγοριοποιεί κείμενα. Η δομή των πινάκων φαίνεται και στο παρακάτω ER διάγραμμα.



Εικόνα 8. Σχεδιάγραμμα της βάσης δεδομένων

Όπως βλέπουμε και από το παραπάνω σχήμα, υπάρχει μία σειρά από πίνακες οι οποίοι αν και αποτελούν τα δεδομένα για διαφορετικές διαδικασίες έχουν άμεσες συσχετίσεις μεταξύ τους. Κάτι τέτοιο είναι αναμενόμενο σε ένα σύστημα στο οποίο μολονότι τα υποσυστήματα που το απαρτίζουν μπορούν να λειτουργήσουν ανεξάρτητα, υπάρχει άμεση σχέση στα δεδομένα που επεξεργάζονται. Αυτό σημαίνει πως υπάρχουν σημαντικοί δεσμοί και πληρότητα στα δεδομένα. Παράλληλα κάτι τέτοιο υποδεικνύει πως το σύστημα είναι ενιαίο και πως η αρμονική λειτουργία των υποσυστημάτων του οδηγεί στο επιθυμητό αποτέλεσμα.

7.1.1 Ανάλυση των πινάκων της βάσης δεδομένων

Στην ενότητα αυτή θα γίνει ανάλυση όλων των πινάκων που απαρτίζουν τη βάση δεδομένων και θα αναλυθούν τα πεδία που απαρτίζουν τον κάθε πίνακα καθώς και ο σκοπός ύπαρξής τους.

7.1.1.1 Πίνακας urls

Ο πίνακας αυτός περιέχει τα στοιχεία για τους ιστότοπους από τους οποίους ο μηχανισμός ανάκτησης θα αντλεί τη διεύθυνση της κάθε ιστοσελίδας με σκοπό να ανακτήσει το κώδικα HTML αυτής. Περιέχει δύο πεδία.

- id [integer (10)]: Πρόκειται για το αναγνωριστικό κάθε url, έχει δηλωθεί UNSIGNED, έχουμε επιλέξει auto_increment και είναι και το κλειδί του συγκεκριμένου πίνακα
- name [text]: Είναι η διεύθυνση (url) του κάθε ιστότοπου

7.1.1.2 Πίνακας datas

Ο πίνακας αυτός περιέχει όλα τα δεδομένα τα οποία προέρχονται από την λειτουργία του μηχανισμού επεξεργασίας του HTML κώδικα κάθε ιστοσελίδας. Συγκεκριμένα περιλαμβάνει τη διεύθυνση της κάθε ιστοσελίδας, την ακριβή ημερομηνία και ώρα που πραγματοποιήθηκε η επεξεργασία και φυσικά περιέχει το «καθαρό» κείμενο κάθε ιστοσελίδας. Από τα δεδομένα αυτού του πίνακα αντλεί πληροφορία ο μηχανισμός κατηγοριοποίησης. Περιέχει τέσσερα πεδία.

- id [integer (10)]: Πρόκειται για το αναγνωριστικό κάθε url και είναι το κλειδί του συγκεκριμένου πίνακα, όπως και στο προηγούμενο πίνακα έχει δηλωθεί UNSIGNED και έχουμε επιλέξει auto_increment.
- url [text]: Περιέχει τη διεύθυνση (url) κάθε ιστοσελίδας
- datetime [text]: Περιέχει την ημερομηνία και ώρα που ανακτήθηκε και επεξεργάστηκε ο HTML κώδικας κάθε ιστοσελίδας
- text [text]: Περιέχει το καθαρό κείμενο που προκύπτει μετά από την επεξεργασία κειμένου

7.1.1.3 Πίνακας caturls

Ο πίνακας αυτός αποτελεί την έξοδο του συστήματός μας. Τα δεδομένα του πίνακα προέρχονται από τη λειτουργία του μηχανισμού κατηγοριοποίησης. Περιλαμβάνει τη διεύθυνση κάθε ιστοσελίδας, τη κατηγορία που ανήκει και μία

ποσότητα που αναφέρεται στη συσχέτιση της πληροφορίας που είναι διαθέσιμη σε κάθε ιστοσελίδα με τη κατηγορία που εντάχθηκε από το μηχανισμό κατηγοριοποίησης. Περιέχει τέσσερα πεδία.

- id [integer (10)]: Πρόκειται για το αναγνωριστικό κάθε κατηγοριοποιημένης σελίδας και είναι το κλειδί του συγκεκριμένου πίνακα, όπως και στους προηγούμενους πίνακες έχει δηλωθεί UNSIGNED και έχουμε επιλέξει auto_increment.
- name [text]: Περιέχει τη διεύθυνση (url) κάθε ιστοσελίδας
- category [text]: Περιέχει τη κατηγορία που εντάχθηκε η ιστοσελίδα, βάση του περιεχομένου της
- weight [text]: Περιέχει τη συσχέτιση κάθε ιστοσελίδας με τη κατηγορία στην οποία τοποθετήθηκε

7.2 Διασύνδεση των μηχανισμών του συστήματος με τη βάση δεδομένων

Όπως έχουμε αναφέρει και οι τρεις μηχανισμοί του συστήματός μας θα πρέπει να είναι σε θέση να επικοινωνήσουν με τη ΒΔ, είτε για να αντλήσουν πληροφορία από αυτή, είτε για να αποθηκεύσουν πληροφορία σε αυτή. Ο μηχανισμός ανάκτησης πληροφορίας θα πρέπει να εξάγει από το πεδίο name του πίνακα urls τις διαθέσιμες διευθύνσεις ιστοσελίδων ώστε να ανακτήσει το κώδικά τους. Ο μηχανισμός επεξεργασίας θα πρέπει να αποθηκεύει στο πίνακα datas την επεξεργασμένη πληροφορία κάθε ιστοσελίδας και κάθε άλλο βοηθητικό στοιχείο. Τέλος ο μηχανισμός κατηγοριοποίησης θα πρέπει να ανακτά από το πεδίο text του πίνακα datas το «καθαρό» κείμενο ώστε να το κατηγοριοποιήσει και στη συνέχεια να αποθηκεύει τη κατηγοριοποιημένη πληροφορία στα πεδία του πίνακα caturls. Έτσι καθώς οι μηχανισμοί ανάκτησης και επεξεργασίας υλοποιήθηκαν σε γλώσσα Java και η ΒΔ με τεχνολογία Mysql για να επιτύχουμε την επικοινωνία των παραπάνω μηχανισμών με τη ΒΔ χρησιμοποιούμε τον οδηγό Mysql Connector/J 5.1 και συγκεκριμένα τη σταθερή έκδοση 5.1.8 [26].

7.3 Παρουσίαση βασικότερων αλγορίθμων

Όπως έχουμε αναφέρει ο μηχανισμός επεξεργασίας πληροφορίας αφορά το βασικότερο κομμάτι της εργασίας μας. Στις προηγούμενες ενότητες αναπτύξαμε τον τρόπο με τον οποίο μεταφέρεται η πληροφορία που επιθυμούμε από το διαδίκτυο στο σύστημά μας, καθώς επίσης και το τρόπο που μεταφέρεται ανάμεσα στους μηχανισμούς του συστήματός μας. Σε αυτή την ενότητα θα δούμε βασικούς αλγορίθμους με τους οποίους υλοποιείται η διαδικασία με την οποία γίνεται η επεξεργασία με σκοπό την εξαγωγή του καθαρού κειμένου από τον HTML κώδικα κάθε ιστοσελίδας. Οι αλγόριθμοι που παρουσιάζονται παρακάτω είναι γραμμένοι σε ψευδοκώδικα, στο παράρτημα της εργασίας είναι διαθέσιμοι υλοποιημένοι σε γλώσσα Java.

7.3.1 Αλγόριθμοι υλοποίησης του μηχανισμού επεξεργασίας

Παρακάτω παρουσιάζεται σε ψευδοκώδικα ο αλγόριθμος που ακολουθήσαμε για να απομονώσουμε από το κώδικα μίας ιστοσελίδας το σώμα (body) του.

```
Διάβασε το κείμενο
Μάρκαρε την ετικέτα έναρξης <body>
Μάρκαρε την ετικέτα λήξης </body>
Απομόνωσε το κείμενο που βρίσκεται ανάμεσα στις δύο μαρκαρισμένες ετικέτες
```

Ακολουθεί ο αλγόριθμος που αφαιρεί από το κείμενο τα κομμάτια κώδικα που υλοποιούν διάφορες λειτουργίες της ιστοσελίδας (scripts) καθώς και τα κομμάτια που υλοποιούν λειτουργίες μορφοποίησης (styles). Και τα δύο κομμάτια όπως έχουμε αναφέρει δε περιέχουν ωφέλιμη πληροφορία.

```
Μάρκαρε την ετικέτα έναρξης <script>

While (μαρκάρεις ετικέτες έναρξης)

        Μάρκαρε την ετικέτα λήξης </script>
        Αντικατέστησε το κείμενο ανάμεσα στις μαρκαρισμένες ετικέτες με το κενό χαρακτήρα

End while

Μάρκαρε την ετικέτα έναρξης <style>

While (μαρκάρεις ετικέτες έναρξης)

        Μάρκαρε την ετικέτα λήξης </style>
        Αντικατέστησε το κείμενο ανάμεσα στις μαρκαρισμένες ετικέτες με το  κενό χαρακτήρα

End while
```

Στη συνέχεια παρουσιάζεται ο τελευταίος από τους βασικούς αλγορίθμους που υλοποιούν τις λειτουργίες του μηχανισμού επεξεργασίας. Σκοπός του παρακάτω αλγορίθμου είναι η αφαίρεση από το κείμενο όσων λέξεων έχουν μήκος μικρότερο ή ίσο του τέσσερα. Για την υλοποίηση αυτής της διαδικασίας κάνουμε χρήση της ενσωμάτωσης στη γλώσσα που χρησιμοποιήσαμε των κανονικών εκφράσεων (regular expressions). Για τη χρήση μίας κανονικής έκφρασης θα πρέπει να ορίσουμε το αλφάβητο Σ. Εδώ το αλφάβητό μας αποτελείται από τους 24 πεζούς χαρακτήρες του αγγλικού αλφαβήτου [a-z], καθώς όπως έχουμε αναφέρει όλο μας το κείμενο έχει μετατραπεί σε πεζούς χαρακτήρες. Η σύνταξη της κανονικής έκφρασης που χρησιμοποιούμε είναι: [a-z][a-z][a-z][a-z][a-z][a-z]*. Η προηγούμενη κανονική έκφραση εκφράζει τις λέξεις που έχουν μήκος τουλάχιστον πέντε. Ακολουθεί ο αλγόριθμος που υλοποιεί τη διαδικασία που μόλις περιγράψαμε.

```
Όρισε το πρότυπο [a-z][a-z][a-z][a-z][a-z][a-z]*
Διάβασε όλες τις λέξεις του κειμένου

for (i=0 -> πλήθος_λέξεων)

    Συμφωνεί αυτή η λέξη με το πρότυπο
    Αν όχι, αφαίρεσε τη
    Αν ναι συνέχισε στην επόμενη λέξη

End for
```

7.4 Κατηγοριοποίηση των δεδομένων

Η κατηγοριοποίηση των δεδομένων γίνεται μέσω ενός αλγορίθμου που είναι παραλλαγή του SVM και ουσιαστικά βρίσκει τη συσχέτιση του κειμένου με τα πρότυπα κείμενα που δημιουργούν τις κατηγορίες του συστήματός μας.

Το αποτέλεσμα που επιθυμούμε να εξάγουμε είναι η συσχέτιση ενός κειμένου με κάθε κατηγορία. Αυτό που κάνουμε είναι να διαβάζουμε για ένα συγκεκριμένο κείμενο, τις λέξεις που το απαρτίζουν και το βάρος που έχει κάθε μία στο συγκεκριμένο κείμενο. Εν συνεχεία για κάθε κατηγορία που έχουμε διαβάζουμε τα βάρη πάνω στις κοινές λέξεις που υπάρχουν με το κείμενο. Στον παρακάτω κώδικα φαίνεται αυτή η διαδικασία.

Όπως μπορούμε να δούμε, αρχικά αποθηκεύουμε σε μία μεταβλητή τύπου Vector όλες τις λέξεις τις κατηγορίας και ο έλεγχος για τις κοινές λέξεις γίνεται με σύγκριση στοιχείων Vector. Ο αρχικός σχεδιασμός πραγματοποιούσε τη σύγκριση και εύρεση κοινών λέξεων με πολλαπλά ερωτήματα στη βάση. Ο αλγόριθμος που εξέφραζε αυτή τη διαδικασία ήταν:

```
Διάβασε τις λέξεις του κειμένου
    For(i=0 -> αριθμός_λέξεων)

    Υπάρχει αυτή η λέξη στη βάση για τη συγκεκριμένη κατηγορία;
    Αν ναι, αποθήκευσε στην αντίστοιχη θέση ενός άλλου Vector τα στοιχεία
    Αν όχι, αποθήκευσε στην αντίστοιχη θέση ενός άλλου Vector το μηδέν

End For
```

Παρόλο που ακούγεται μία λογική διαδικασία, συμβαίνει το εξής. Για κάθε κείμενο που θέλουμε να κατηγοριοποιήσουμε το οποίο, έστω πως έχει περίπου 20 λέξεις-κλειδιά, για τις 7 κατηγορίες που θέλουμε να βρούμε τις συσχετίσεις, έχουμε 20x7 = 140 ερωτήματα στη βάση δεδομένων. Το πρόβλημα αυτό έγινε κατανοητό, μόλις παρατηρήθηκε πως ενώ όλη η προηγούμενη διαδικασία χρειαζόταν 1-2 λεπτά για να ολοκληρωθεί, σε αυτό το σημείο η διαδικασία χρειαζόταν περίπου 7 λεπτά. Έτσι αποφασίσαμε να δημιουργούμε Vectors στην αρχή που θα περιέχουν όλες τις λέξεις της κατηγορίας και η σύγκριση να μη γίνεται με ερωτήματα στη βάση δεδομένων αλλά με σύγκριση των Vectors. Αξίζει να σημειωθεί πως ο χρόνος για την

εύρεση συσχέτισης ενός κειμένου με όλες τις κατηγορίες είναι λιγότερος από 1 δευτερόλεπτο. Οι αριθμητικές πράξεις που ακολουθούν στον κώδικα είναι γνωστές από προηγούμενο κεφάλαιο και γίνεται απλή εφαρμογή του τύπου. Κάθε αποτέλεσμα που εξάγεται, αποθηκεύεται σε πίνακα με τη μορφή, αναγνωριστικό, αναγνωριστικό κειμένου, αναγνωριστικό κατηγορίας, συσχέτιση.

Αυτός ο πίνακας περιέχει όλες τις συσχετίσεις, ανεξάρτητα αν το κείμενο ανήκει (συσχέτιση > 0.6) ή όχι (συσχέτιση < 0.4) σε μία κατηγορία, καθώς πρέπει να γίνει μεγαλύτερη επεξεργασία αυτών των αποτελεσμάτων, αφού σε κάθε κείμενο υπάρχει η περίπτωση τα αποτελέσματα να είναι σχετικά και να μην υποδηλώνουν άμεσα την κατηγορία στην οποία ανήκει το κείμενο.

Κεφάλαιο

8

8 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Η εξάπλωση του Διαδικτύου έχει λάβει χαοτικές διαστάσεις και η πληροφορία που διακινείται σε αυτό είναι υπέρογκη. Στην εποχή μας και με τα μέσα που διαθέτει ακόμα και ο απλός χρήστης, η προσθήκη περιεχομένου στο Διαδίκτυο από τον καθένα, είναι μια διαδικασία το ίδιο εύκολη με την απλή περιαγωγή στο χώρο του παγκόσμιου ιστού. Το πρόβλημα που δημιουργεί αυτή η χαοτική κατάσταση οδηγεί ακόμα και τους πιο έμπειρους χρήστες να καταναλώνουν πολύ χρόνο στην προσπάθεια εύρεσης συγκεκριμένης πληροφορίας.

Το σύστημα που δημιουργήθηκε στα πλαίσια της διπλωματικής αυτής εργασίας περιλαμβάνει μία σειρά μηχανισμών που ξεκινούν με ανάκτηση πληροφορίας και καταλήγουν στην επιστροφή της πληροφορίας στο χρήστη αφού υποστεί επεξεργασία.

Ένας τέτοιος μηχανισμός, που περιλαμβάνει πολλά υποσυστήματα χρίζει ιδιαίτερης προσοχής καθώς το παραμικρό σφάλμα σε οποιοδήποτε κομμάτι του μηχανισμού μπορεί να δημιουργήσει σοβαρό πρόβλημα συνολικά στο σύστημα.

Η διαδικασία κατασκευής ολοκληρώθηκε μετά από εργασία 6 μηνών κάτι το οποίο είναι πολύ λογικό αν σκεφτεί κανείς πως έπρεπε σε κάθε βήμα να δημιουργείται ένα υποσύστημα και να ελέγχεται εκτενώς η καλή λειτουργία του προτού συνεχίσουμε στην κατασκευή του επόμενου κομματιού.

Συμπερασματικά μπορούμε να πούμε πως το σύστημα βρίσκεται σε ένα πρωταρχικό στάδιο, ανεξάρτητα αν είναι πλήρως λειτουργικό και αποδοτικό. Οι βελτιώσεις που μπορούν να γίνουν στο μηχανισμό είναι πολλές. Ακόμα και μικρές αλλαγές που επιχειρήθηκαν κατά τη διάρκεια βελτίωσης του κώδικα μας έδειξαν ότι το σύστημα μπορεί να ανεβάσει την απόδοσή του κατακόρυφα με τη βελτίωση ορισμένων στοιχείων, τα οποία αφορούν κυρίως την επικοινωνία με τη ΒΔ αλλά και τη διαχείριση κειμένου.

Το πεδίο το οποίο καλύπτει η συγκεκριμένη εργασία είναι ευρύ και παρουσιάζει εξαιρετικό ενδιαφέρον τα τελευταία χρόνια. Πολλές και μεγάλες έρευνες έχουν γίνει στο συγκεκριμένο τομέα ενώ οι αλγόριθμοι κατηγοριοποίησης κειμένων ολοένα βελτιώνονται. Η γνώση αλλά και η ερευνητική διαδικασία στην οποία μας εισήγαγε η προσπάθεια κατασκευής και τεκμηρίωσης του συστήματος αποτελούν σημαντικά εφόδια για το μέλλον ενώ παράλληλα οι προβληματισμοί οι οποίοι δημιουργήθηκαν αποτελούν βάση για τη δημιουργία νέων τεχνικών και αλγορίθμων πάνω στο κομμάτι που αφορά την επεξεργασία και κατηγοριοποίηση πληροφορίας.

Όσο αφορά το κομμάτι της εξαγωγής χρήσιμου κειμένου από html σελίδες οι βελτιώσεις που μπορούν να γίνουν θα δώσουν ώθηση συνολικά στο σύστημα. Σε αυτό το επίπεδο πρέπει να βελτιωθεί η ποιότητα της πληροφορίας δηλαδή θα πρέπει να

περιοριστεί στο ελάχιστο το σφάλμα που γίνεται και να εξάγεται μόνο κείμενο το οποίο μας ενδιαφέρει και καμία επιπλέον πληροφορία. Επίσης θα μπορούσε να μη γίνεται απλά εξαγωγή αλλά και άμεσος έλεγχος του κειμένου ούτως ώστε να μην εισάγεται στο σύστημα το παραμικρό στοιχείο το οποίο μπορεί να αποτελέσει ανασταλτικό παράγοντα για την ομαλή λειτουργία.

Κατά τη διαδικασία κατηγοριοποίησης οι βελτιώσεις που μπορούν να γίνουν είναι η δυνατότητα χρήσης πολλών αλγορίθμων προκειμένου να βρεθούν οι συσχετίσεις των κειμένων με τις κατηγορίες. Η διαδικασία αυτή μπορεί να γίνεται για πειραματικούς λόγους (σύγκριση αλγορίθμων) είτε για επιβεβαίωση της ορθής λειτουργίας του συστήματος.

Τέλος να αναφέρουμε πως μία άμεση και πολύ χρήσιμη επέκταση της παρούσας εργασίας θα μπορούσε να αποτελέσει η δημιουργία ενός portal μέσω του οποίου οι χρήστες θα έχουν τη δυνατότητα να αλληλεπιδρούν άμεσα με το σύστημα, καθώς επίσης και η επέκταση των μηχανισμών κατηγοριοποίησης ώστε να εκπαιδεύονται από τις επιλογές του χρήστη και το σύστημα να προσφέρει μεγαλύτερη παραμετροποιησιμότητα.

Κλείνοντας τη διπλωματική θα πρέπει να τονιστεί πως η δημιουργία του συστήματος μας ανοίγει αρκετούς δρόμους για έρευνα και ανάλυση συστημάτων επεξεργασίας και κατηγοριοποίησης κειμένων.

ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα της εργασίας μας παρατίθενται τα σημαντικότερα κομμάτια κώδικα που υλοποιούν τους μηχανισμούς ανάκτησης και επεξεργασίας του συστήματός μας, τα οποία αποτελούν το βασικότερο τμήμα της παρούσας εργασίας και έχουν υλοποιθεί σε γλώσσα προγραμματισμού Java.

Αρχικά παραθέτουμε το κώδικα που υλοποιεί το μηχανισμό ανάκτησης πληροφορίας από το διαδίκτυο. Η παρακάτω συνάρτηση υλοποιεί την επικοινωνία με τον εξυπηρετητή (server), από όπου ανακτά το κώδικα HTML μίας ιστοσελίδας, τον οποίο αποθηκεύει σε μία μεταβλητή τύπου String.

```
getHtml(URL Pageurl){
    try {

        URLConnection GetConn = null;
        GetConn = null;
        PageUrl = Pageurl;
        GetConn = PageUrl.openConnection();
        GetConn.connect();


        InputStreamReader ReadIn = new InputStreamReader(GetConn.getInputStream());
        BufferedReader BufData = new BufferedReader(ReadIn);
        StringBuffer htmlSiteBuffer = new StringBuffer();
        String tmp;
        while ((tmp = BufData.readLine()) != null) {
          htmlSiteBuffer.append(tmp);
          htmlSiteBuffer.append("\n");
          }
        BufData.close();

        str = htmlSiteBuffer.toString();

        }
        catch (IOException io) {
        System.out.println(io.getMessage());
            }
  }
```

Κώδικας μηχανισμού ανάκτησης (συνάρτηση getHtml)

Στη συνέχεια παραθέτουμε κομμάτια κώδικα που υλοποιούν βασικές συναρτήσεις του μηχανισμού επεξεργασίας.

Η παρακάτω συνάρτηση αναλαμβάνει την εξόρυξη του σώματος (body) από το κώδικα HTML.

```
public StringBuffer body2body(String str2) {
      str2 = str2.toLowerCase();
      StringBuffer strtext = new StringBuffer();
      try {
         int startPosition = str2.indexOf("<body");
         int endPosition = str2.indexOf("</body>");
         str2 = str2.substring(startPosition, endPosition + 7);

         strtext.append(str2);


      } catch (StringIndexOutOfBoundsException f) {
         System.out.println(f.getMessage());
      }
      return strtext;
   }
```

Κώδικας μηχανισμού επεξεργασίας (συνάρτηση body2body)

Η συνάρτηση που ακολουθεί αναλαμβάνει τη διαδικασία αφαίρεσης κομμάτια κώδικα που δε περιέχουν ωφέλιμη πληροφορία σε ένα κώδικα HTML. Επίσης αφαιρεί και όλες τις άλλες ετικέτες που χρησιμοποιούνται από τη γλώσσα HTML για διάφορες λειτουργίες.

```
public String rm_scripts_styles(StringBuffer strtext) {
try {
         int startPosition = strtext.indexOf("<script");
         while (startPosition != -1) {
            int endPosition = strtext.indexOf("</script>");
            strtext.replace(startPosition, endPosition + 8, " ");
            startPosition = strtext.indexOf("<script");
         }


         startPosition = strtext.indexOf("<style");
         while (startPosition != -1) {
            int endPosition = strtext.indexOf("</style>");
            strtext.replace(startPosition, endPosition + 7, " ");
            startPosition = strtext.indexOf("<style");
         }

         startPosition = strtext.indexOf("<");

         while (startPosition != -1) {
            int endPosition = strtext.indexOf(">");

            while (endPosition < startPosition) {
               strtext.replace(endPosition, endPosition, "&&&");
               endPosition = strtext.indexOf(">");
            }
            strtext.replace(startPosition, endPosition + 1, " ");
```

```
            startPosition = strtext.indexOf("<");
          }
        } catch (StringIndexOutOfBoundsException f) {
          System.out.println(f.getMessage());
        }
        String str = strtext.toString();
        return str;
    }
```

Κώδικας μηχανισμού επεξεργασίας (συνάρτηση rm_scripts_styles)

Παρακάτω παραθέτουμε το κώδικα των δύο συναρτήσεων που αναλαμβάνουν να αφαιρέσουν από το κείμενο σημεία στίξης καθώς και οποιαδήποτε άλλα στοιχεία τα οποία παρατηρήσαμε πως κατέστρεφαν τη μορφή που πρέπει να έχει το επεξεργασμένο κείμενο ώστε να ενεργήσει σωστά ο μηχανισμός κατηγοριοποίησης και φυσικά τέτοιου είδους στοιχεία δεν αποτελούν ωφέλιμη πληροφορία.

```
public String addSlashes(String str) {
    final StringBuffer sb = new StringBuffer(str.length() * 2);
    final StringCharacterIterator iterator = new StringCharacterIterator(str);

    char character = iterator.current();

    while (character != StringCharacterIterator.DONE) {
      if (character == "") {
        sb.append("\\\"");
      } else if (character == '\") {
        sb.append("\\\"");
      } else if (character == '\\') {
        sb.append("\\\\");
      } else if (character == '\n') {
        sb.append("\\n");
      } else if (character == '{') {
        sb.append("\\{");
      } else if (character == '}') {
        sb.append("\\}");
      } else if (character == '&') {
        sb.append("");
      } else if (character == '?') {
        sb.append("");
      } else {
        sb.append(character);
      }

      character = iterator.next();
    }

    return sb.toString();
  }
```

```
public String rm_backsl(String sb) {
   String text = sb.replaceAll("[\\W&&[^\\\\]]+", " ");
   return text;
}
```

Κώδικας μηχανισμού επεξεργασίας (συνάρτήσεις addSlashes και rm_backsl)

Τέλος παραθέτουμε το κώδικα που υλοποιεί τη διαδικασία αφαίρεσης των λέξεων με μήκος μικρότερο ή ίσο του τέσσερα από το «καθαρό» κείμενο. Όπως έχουμε αναφέρει η διαδικασία αυτή υλοποιήθηκε με χρήση κανονικών εκφράσεων (regular expressions), μηχανισμό που υποστηρίζει η γλώσσα προγραμματισμού Java.

```
public String rm_bigwords(String sb) {
    String text = new String();
    Pattern pattern = Pattern.compile("[a-z][a-z][a-z][a-z][a-z][a-z]*");
    for (String tmp : sb.split(" ")) {
       Matcher matcher = pattern.matcher(tmp);
       if (matcher.matches()) {
            text = text + matcher.group() + " ";
       }
 }
```

Κώδικας μηχανισμού επεξεργασίας (συνάρτηση rm_bigwords)

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Mooers, C. N. 1952. Information Retrieval Viewed as Temporal Signaling. In Proceedings of the International Conference of Mathematicians, Cambridge, Massachusetts. American Mathematical Society, σελίδες 572-573.

[2] Doyle L. B. 1961. Semantic Road Maps for Literature Searchers. In Journal of the Association for Computing Machinery, 8, σελίδες 553-578.

[3] Salton, G. 1968. Automatic Information Organization and Retrieval. New York: McGraw-Hill.

[4] Shneiderman, B., Byrd, D. and Croft, B. 1998. Sorting out Searching: a User-Interface Framework for Text Searches. In Communications of the ACM, 41(4), σελίδες 95-98.

[5] Salton, G. and Buckley, C. 1988. Improving Retrieval Performance by Relevance Feedback. In Journal of the American Society for Information Science, 41, σελίδες 288-297.

[6] Cleverdon, C. W. 1972. The Cranfield Tests on Index Language Devices. In Aslib Proceedings, 19, σελίδες 173-192.

[7] Belkin, N. J., and Croft, W. B. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? In Communications of the ACM, 35(12), σελίδες 29-38.

[8] Quinlan, J. R. 1986. Induction of Decision Trees. In *Machine Learning I*.

[9] Chickering, D., Heckerman, D. AND MECK C. 1997. A Bayesian Approach for Learning Bayesian Networks with Local Structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*.

[10] Belur, V. D. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. McGraw-Hill Computer Science Series. IEEE Computer Society Press.

[11] Vapnik, V. 1995. The Nature of Statistical Learning Theory, Springer-Verlag.

[12] Cortes, C. and Vapnik, V. 1995. Support-Vector Networks. In Machine Learning, 20, σελίδες 273-297.

[13] Belur, V. D. 1991. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. McGraw-Hill Computer Science Series. IEEE Computer Society Press.

[14] Ανάκτηση Πληροφορίας: Πανεπιστημιακό Σύγγραμμα, Επίκ. Καθηγητής Χ. Μακρής.

[15] Εισαγωγή στη Θεωρία Υπολογισμού, Michael Sipser, σελίδες 73-74.

[16] M. Lennon. Pierce. D., Tarry, B.. & Willett, An evaluation of the stemming algorithms.

[17] Μηχανή αναζήτησης Google. http://www.google.com.

[18] Μηχανή αναζήτησης Altavista. http://www.altavista.com.

[19] MySQL, Βάση Δεδομένων ανοιχτού κώδικα. http://www.mysql.com.

[20] PostgreSQL, Βάση Δεδομένων ανοιχτού κώδικα. http://www.postrgresql.org.

[21] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα C. http://en.wikipedia.org/wiki/C.

[22] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα C++ (C Plus Plus). http://en.wikipedia.org/wiki/C_Plus_Plus.

[23] Το χρονικό της Java. http://ils.unc.edu/blaze/java/javahist.html.

[24] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα Perl. http://en.wikipedia.org/wiki/Perl.

[25] Ειδησεογραφικό Πρακτορείο CNN. http://edition.cnn.com.

[26] MySQL Connector/J 5.1. http://dev.mysql.com/downloads/connector/j/5.1.html

ΕΥΡΕΤΗΡΙΟ