ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Διαχωρισμός και κατηγοριοποίηση καταχωρήσεων ιστολόγιων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Αντώνιου Αναστασιάδη

Επιβλέπων: Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάϊος 2008

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Διαχωρισμός και κατηγοριοποίηση καταχωρήσεων ιστολόγιων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Αντώνιου Αναστασιάδη

Επιβλέπων: Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Δευτέρα 12 Μαΐου 2008.

.....	.....	.....
Τιμολέων Σελλής Καθηγητής ΕΜΠ	Ιωάννης Βασιλείου Καθηγητής ΕΜΠ	Νεκτάριος Κοζύρης Επίκ. Καθηγητής ΕΜΠ

Αθήνα, Μάϊος 2008

-

. . . . . .

Αναστασιάδης Αντώνιος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Ε.Μ.Π.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων της διπλωματικής μου εργασίας, ερευνητή του Ε.Κ.Ε.Φ.Ε. 'Δημόκριτος' Γιώργο Παλιούρα για την πολύτιμη βοήθειά του καθ'όλη την διάρκεια της συνεργασίας μας, καθώς και για την συλλογή δεδομένων που μου παραχώρησε. Επίσης ευχαριστώ τον Κωσταντίνο Χανδρινό για τις συμβουλές του και για τις συζητήσεις που είχαμε μαζί. Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή του Εθνικού Μετσόβιου Πολυτεχνείου Τίμο Σελλή για την εποπτεία της διπλωματικής μου εργασίας και για την συνεργασία που είχαμε.

Περίληψη

Σκοπός της εργασίας αυτής είναι η ανάπτυξη μεθόδων για τον διαχωρισμό και την εξαγωγή των καταχωρήσεων από ιστοσελίδες ιστολόγιων και την κατηγοριοποίησή τους με βάση την άποψη που εκφράζουν για κάποιο θέμα. Αυτές οι μέθοδοι εκμεταλλεύονται την συντακτική πληροφορία του κώδικα των ιστοσελίδων, τα feeds τους καθώς και τις ημερομηνίες που περιέχουν ωστε να εξάγει τις καταχωρήσεις τους. Κατόπιν, χρησιμοποιούμε έναν αλγόριθμο Διανυσμάτων Υποστήριξης ωστε να ταξινομήσουμε τις καταχωρήσεις που εξήχθησαν από ιστολόγια με κριτικές ταινιών σε δύο σύνολα θετικών και αρνητικών απόψεων αντίστοιχα.

Στα πλαίσια της εργασίας υλοποιήθηκε μια εφαρμογή στη γλώσσα Java η οποία δέχεται ένα σύνολο ιστολόγιων, τα επεξεργάζεται και εξάγει τις καταχωρήσεις τους με αυτοματοποιημένο και αποδοτικό τρόπο. Επίσης, υλοποιήθηκαν και άλλα εργαλεία τα οποία δέχονται τα δεδομένα των καταχωρήσεων, τα μετατρέπουν σε μορφή έτοιμη προς κατηγοριοποίηση και πραγματοποιούν την ταξινόμησή τους.

Η μεθοδολογία αυτή θα μπορούσε να χρησιμοποιηθεί σαν βάση για ένα σύστημα αυτόματης ανάλυσης των ιστολόγιων του διαδικτύου και ταξινόμησης της πληροφορίας τους, χρησιμοποιώντας επιπλέον μεθόδους όπως γλωσσολογική ανάλυση και αυτόματη εκμάθηση στην εξαγωγή και την ταξινόμηση των καταχωρήσεων.

Λέξεις κλειδιά: *διαδίκτυο,ιστολόγιο,κατηγοριοποίηση κειμένου,άποψη*

Abstract

The scope of this thesis was the development of methods for the automatic extraction of the posts found in blog pages on the internet, and to classify them as to the opinion they represent regarding a specific topic. Those methods take advantage of the syntactic information of the HTML code of the blog web pages, as well as their feeds and the date strings they contain. We also use an algorithm with Support Vector Machines to classify the extracted posts into two collections that represent the positive and negative opinions respectively.

Moreover, we developed a stand-alone Java application, that given a corpus of blogs, it extracts their posts in an automatic and efficient way. We also developed tools that format the extracted data in feature vector representation format that is ready for classification, as well as classify it.

This work can be used as a basis for a more complex system that finds, separates and classifies blogs using more advanced methods such as lingual analysis and machine learning to extract and classify their posts.

Key words: *internet,blog,text classification,sentiment*

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Ιστολόγια (blogs)

Τα τελευταία χρόνια το διαδίκτυο έχει γνωρίσει δραματική ανάπτυξη και έχει εισχωρήσει σε πολλές πτυχές της ζωής του σύγχρονου ανθρώπου. Ως λογικό επακόλουθο, το κόστος κατασκευής και συντήρησης μιας ιστοσελίδας είναι σχεδόν μηδενικό, και υπάρχουν πλέον έτοιμα εργαλεία στο διαδίκτυο που δίνουν οι την δυνατότητα γρήγορης δημιουργίας και εύκολης συντήρησης μιας ιστοσελίδας. Κατὲπέκταση, δίνεται η δυνατότητα σε οποιονδήποτε έχει πρόσβαση στο διαδίκτυο να γνωστοποιήσει στο ευρύ κοινό την πληροφορία που θέλει. Το γεγονός αυτό έχει οδηγήσει στην ανάπτυξη ενός νέου είδους ιστοσελίδας το οποίο ονομάζεται blog. Ο όρος blog, προέρχεται απο τις λέξεις web log και στα ελληνικά η επικρατέστερη λέξη είναι 'ιστολόγιο'. Το ιστολόγιο είναι ουσιαστικά μια ιστοσελίδα σε μορφή ημερολογίου, η οποία ενημερώνεται απο έναν ή περισσότερους ανθρώπους και στην οποία αναγράφονται παντός είδους πληροφορίες που θέλουν να μοιραστούν με τον υπόλοιπο κόσμο.

Τα ιστολόγια έχουν γνωρίσει δραματικά μεγάλη αύξηση [1] τον τελευταίο καιρό και ο αριθμός τους εκτιμάται σε αρκετά εκατομμύρια, ιδιαίτερα σε χώρες στις οποίες υπάρχει μεγάλο ποσοστό χρηστών του διαδικτύου. Η Αμερική και η Κίνα έχουν το μεγαλύτερο αριθμό χρηστών που έχουν δημιουργήσει και συντηρούν ενεργά ιστολόγια, αν και πολλές άλλες χώρες με μεγάλο ποσοστό διείσδυσης του διαδικτύου στεγάζουν επίσης τεράστιους αριθμούς ιστολόγιων.

Η διάκριση μιας ιστοσελίδας που εμπίπτει στην κατηγορία των ιστολόγιων από τις άλλες δεν μπορεί να γίνει με βάση κάποια σταθερά κριτήρια, καθώς ο ορισμός του ιστολόγιου είναι εν γένει αρκετά χαλαρός. Τα γενικά χαρακτηριστικά που μπορεί να έχει μια ιστοσελίδα για να χαρακτηριστεί ως ιστολόγιο είναι πρωταρχικά τα εξής:

- Διαχωρισμός της πληροφορίας σε ξεχωριστές καταχωρήσεις (posts).
- Κατηγοριοποίησή τους με βάση την ημερομηνία, τον συγγραφέα ή το θεματικό υπόβαθρο (η με όποιο πιθανό συνδυασμός τους), συνήθως σε μορφή που θυμίζει ημερολόγιο.
- Δυνατότητα προσθήκης σχολίων από τον αναγνώστη.
- Δυνατότητα παρακολούθησης με βάση rss feeds.

Επίσης πολλά ιστολόγια προσφέρουν επιπλέον δυνατότητες όπως ευρετήριο (index) των καταχωρήσεων με βάση την ημερομηνία, τον συγγραφέα ή το θέμα, εγγραφή των χρηστών ως συνδρομητές ωστε να συμμετάσχουν σε μία επώνυμη συζήτηση, καθώς και άλλες μεμονωμένες δυνατότητες οι οποίες συναντώνται περιστασιακά.

Η διαδικασία δημιουργίας μιας εμφανίσιμης ιστοσελίδας είναι συνήθως ιδιαίτερα χρονοβόρα διαδικασία. Ο δημιουργός της σελίδας θα πρέπει να ασχοληθεί με τον σχεδιασμό και τον προγραμματισμό της, καθώς και με την προσθαφαίρεση κώδικα σε κάθε προσθήκη κάποιας νέας καταχώρησης. Για αυτόν τον λόγο τα περισσότερα ιστολόγια πλέον δημιουργούνται με έτοιμα συστήματα που υπάρχουν για αυτό τον σκοπό, πολλά απο τα οποία διατίθενται ελεύθερα στο διαδίκτυο[1]. Το τελευταίο γεγονός διευκολύνει ακόμα περισσότερο τον απλό χρήστη που ενδιαφέρεται για την γρήγορη δημοσίευση κάποιων πληροφοριών δίχως να ασχοληθεί με τις τεχνικές λεπτομέρειες του εγχειρήματος, ανέξοδα και με ελάχιστο κόπο. Ως αποτέλεσμα, έχουν δημιουργηθεί εκατομμύρια ιστοσελίδες οι οποίες περιέχουν δραματικά μεγάλο όγκο πληροφοριών.

Ανέκαθεν γίνονταν προσπάθειες ωστε αφενός να μετρηθούν, και αφετέρου και σημαντικότερο να διαχωριστούν τα δεδομένα των καταχωρήσεων από την υπόλοιπη μή ενδιαφέρουσα πληροφορία. Εάν γίνει αυτό, τότε θα έχουμε στα χέρια μας ένα τεράστιο μέγεθος δεδομένων, από το οποίο με την κατάλληλη ανάλυση [2] θα μπορέσουμε να εξάγουμε χρησιμότατα συμπεράσματα. Όσο μεγαλύτερος είναι ο αριθμός των ιστολόγιων που αναλύουμε, τόσο μεγαλύτερο πλήθος ανθρώπων καλύπτουμε και τόσο αντιπροσωπευτικότερα είναι τα συμπεράσματά μας.

Η πληροφορίες αυτές χρησιμεύουν λόγου χάριν για να βοηθήσουν μια έρευνα όσον αφορά τον αντίκτυπο ενός γεγονότος σε ένα μέρος του πληθυσμού. Ή επίσης μια εταιρεία η οποία θα ήθελε να έχει κάποια στοιχεία σχετικά με τις γνώμες των καταναλωτών ενός προϊόντος της. Γενικότερα, τα δεδομένα

[1] Όπως πχ τα Blogger, Livejournal και άλλα τα οποία σχολιάζουμε στην ενότητα 2.1

[2] Με την χρήση κάποιου αλγορίθμου όπως για παράδειγμα ενός ταξινομητή

που θα έχουμε στα χέρια μας θα περιλαμβάνουν παντός είδους πληροφορίες που αφορούν τον σύγχρονο άνθρωπο και μπορούν να χρησιμοποιηθούν για ερευνητικούς σκοπούς, marketing, διαφήμιση, στατιστικές αναλύσεις[2] και γενικότερα σε ο,τιδήποτε χρειάζονται πληροφορίες που μπορεί να προκύψουν απο μια μεγάλη μάζα ανθρώπων [3].

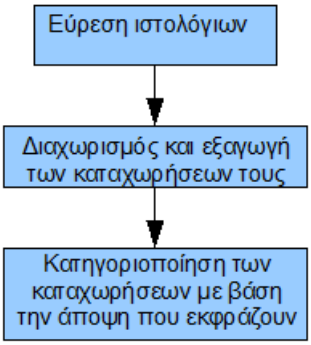
1.2 Σκοπός της εργασίας

Σκοπός της εργασίας αυτής είναι να αναπτυχθεί μια εφαρμογή η οποία θα αναλύει συντακτικώς μια ιστοσελίδα ιστολόγιου και θα εξάγει με αποδοτικό τρόπο τις καταχωρήσεις του συγγραφέα (ή των συγγραφέων στην περίπτωση που δέν είναι μόνο ένας). Σε δεύτερο σταδιο, τα δεδομένα αυτά θα τροφοδοτήσουν κάποιον αλγόριθμο αναγνώρισης και κατηγοριοποίησης της άποψης που εκφράζουν. Οι μέθοδοι εξαγωγής των καταχωρήσεων θα επικεντρώνονται στο να εντοπίσουν τα δεδομένα που έχουν εισάγει οι συγγραφείς της ιστοσελίδας και να τα διαχωρίσουν από τα υπόλοιπα άχρηστα δεδομένα που δέν ανήκουν στις καταχωρήσεις. Η εφαρμογή δέχεται ένα σύνολο από διευθύνσεις ιστοσελίδων (URLs) [3], τις επεξεργάζεται, και αποθηκεύει τα εξαγόμενα δεδομένα στο δίσκο. Τα δεδομένα αυτά αποθηκεύονται σε μορφή html αλλά και σε απλό κείμενο (ascii format) ωστε να είναι ευκολότερη η μετέπειτα κατηγοριοποίησή της άποψής τους δίχως να χρειάζεται επιπλέον συντακτική ανάλυση.

Στο σημείο αυτό σημειώνουμε οτι η εφαρμογή υποθέτει οτι όλες οι εισαγόμενες διευθύνσεις ιστοσελίδων αντιστοιχούν σε ιστολόγια. Δέν έχει υλοποιηθεί δηλαδή κάποιος ευριστικός μηχανισμός αναγνώρισης μιας ιστοσελίδας ιστολόγιου από κάποια άλλη που δέν είναι. Η υλοποίηση ενός τέτοιου μηχανισμού δέν κρίθηκε απαραίτητη, καθώς η εύρεση μεγάλου αριθμού ιστολόγιων είναι πλέον εύκολη και στο διαδίκτυο υπάρχουν συνεχώς ανανεώσιμοι κατάλογοι με εκατομμύρια ιστολόγια. Υπάρχουν επίσης εργασίες που διαπραγματεύονται το πρόβλημα της αναγνώρισης των ιστολόγιων [4], καθώς και έτοιμα εργαλεία για αυτό τον σκοπό [5]. Γενικότερα ο χαρακτηρισμός ή όχι μιας ιστοσελίδας ως ιστολόγιο χαρακτηρίζεται απο έλλειψη σταθερών κριτηρίων, συνεπώς το έργο της αναγνώρισης τους επαφίεται στον χρήστη της εφαρμογής και δέν συμπεριλαμβάνεται στα πλαίσια της παρούσας εργασίας.

Το γεγονός αυτό δέν αποτέλεσε πρόβλημα, καθώς για τους σκοπούς της διπλωματικής εργασίας τροφοδοτήσαμε την εφαρμογή με αρκετές χιλιάδες γνωστών ιστολόγιων. Οι διευθύνσεις των ιστολόγιων εξήθχησαν από δεδομένα που προσέφερε η εταιρεία Intelliseek για ερευνητικούς σκοπούς [6], στα

[3] Unified Resource Locator



Σχήμα 1.1: Διάγραμμα των επιμέρους βημάτων της διαδικασίας που θα ακολουθήσουμε.

οποία υπήρχαν πολύ περισσότερα ιστολόγια από όσα χρειαστήκαμε.

Είναι ήδη γνωστό οτι η συντακτική ανάλυση αποτελεί μονάχα ένα βήμα προς την εξαγωγή του νοήματος απο δεδομένα, και ενδεχομένως να μπορούσαμε να επιτύχουμε υψηλότερη ακρίβεια εξαγωγής εάν χρησιμοποιούσαμε κάποιες μεθόδους γλωσσολογικής ανάλυσης. Όμως, μια όσο το δυνατόν αξιόπιστη συντακτική ανάλυση θα προσφέρει μια γερή βάση για την υλοποίηση μιας πλατφόρμας που θα μειώσει αρκετά το σηματοθορυβικό λόγο των αποτελεσμάτων και θα χρησιμοποιηθεί για περαιτέρω έρευνα.

1.3 Προσέγγιση του προβλήματος

Για τον σκοπό που αναφέραμε προηγουμένως, πρέπει να γίνουν:

- Η συλλογή των ιστολόγιων προς ανάλυση.
- Η εξαγωγή των καταχωρήσεων τους.
- Η κατηγοριοποίηση των καταχωρήσεων.

Με αυτή τη σειρά τα εφαρμόζουμε και αξιολογούμε τα πειράματά μας, όπως φαίνεται και στο διάγραμμα 1.1.

1.4 Διάρθρωση της εργασίας

Σε αυτό το κεφάλαιο έγινε μια εισαγωγή όσον αφορά τα ιστολόγια και τη σημασία τους σήμερα.

Το επόμενο κεφάλαιο παρουσιάζει μια εισαγωγή σε προηγούμενες εργασίας που διαπραγματεύονται το θέμα μας καθώς και στα γνωστά συστήματα δημιουργίας ιστολόγιων στο διαδίκτυο.

Το 3ο κεφάλαιο περιγράφει το τεχνολογικό υπόβαθρο της κατηγοριοποίησης κειμένου πανω στο οποίο βασίζεται η παρούσα εργασία.

Το 4ο κεφάλαιο ασχολείται με τα τεχνικά θέματα της εφαρμογής διαχωρισμού των ιστολόγιων, με το πώς σχεδιάστηκαν οι μέθοδοι που χρησιμοποιεί η εφαρμογή και πώς αυτές εφαρμόστηκαν, και με την διαδικασία της ταξινόμησής των καταχωρήσεων.

Το κεφάλαιο 5 αναφέρει αναλυτικές οδηγίες για την εγκατάσταση και την χρήση της εφαρμογής.

Στο κεφάλαιο 6, στην πρώτη ενότητα αξιολογούμε την απόδοση της εφαρμογής διαχωρίζοντας πραγματικά δεδομένα από ιστολόγια τα οποία συλλέξαμε. Στην δεύτερη ενότητα εκτελούμε πειράματα ωστε να αξιολογήσουμε την κατηγοριοποίηση των καταχωρήσεων που προέκυψαν. Και στις δύο ενότητες σχολιάζουμε τα αποτελέσματα, καθώς και παραθέτουμε κάποιες ιδέες που ενδεχομένως μελλοντικά να βελτιώσουν την απόδοση της εφαρμογής μας.

Κεφάλαιο 2

Σχετικές εργασίες

Στο κεφάλαιο αυτό παρουσιάζουμε τα γνωστά εργαλεία δημιουργίας ιστολόγιων, τις προηγούμενες εργασίες που έχουν πραγματοποιηθεί στο θέμα του διαχωρισμού των ιστολόγιων, καθώς και το σχετικό υπόβαθρο μηχανικής μάθησης που αφορά τον διαχωρισμό και την κατηγοριοποίηση των καταχωρήσεων τους.

2.1 Γνωστά συστήματα δημιουργίας ιστολόγιων

Ένας απο τους λόγους που τα ιστολόγια γνωρίζουν δραματική αύξηση είναι οτι η δημιουργία και η συντήρησή τους διευκολύνεται σημαντικά με την χρήση έτοιμων συστημάτων. Τα συστήματα αυτά, καθώς δημιουργούν τις ιστοσελίδες των ιστολόγιων τοποθετούν κάποιες ετικέτες (tags) μέσα στον κώδικα ωστε να είναι εύκολη η διαχείριση και η ανανέωσή τους. Τα ποιό γνωστά συστήματα κατασκευής ιστολόγιων τα οποία αναλύουμε στην αντίστοιχη μέθοδο όπως θα δούμε παρακάτω, είναι τα εξής:

- Blogger. Το blogger[1] είναι ίσως το πιο ευρέως χρησιμοποιούμενο εργαλείο δημιουργίας ιστολόγιων, και ανήκει στην εταιρεία Google. Στην εικόνα 2.1 βλέπουμε ένα ιστολόγιο που έχει δημιουργηθεί μέσω του Blogger και επίσης στεγάζεται στον ιστότοπό του. Ενδεικτικά, ο κώδικας HTML που περιέχει την δεύτερη καταχώρηση είναι ο εξής:

```
<div class='post hentry uncustomized-post-template'>
<a name='1827303795865676508'></a>
<h3 class='post-title entry-title'>
```

[1] http://www.blogger.com

Figure 2.1: Ιστολόγιο που έχει δημιουργηθεί με το σύστημα Blogger. Οι καταχωρήσεις φαίνονται μέσα στα μαύρα πλαίσια.

```
<a href='...'>Sat. sit stay</a>
</h3>
<div class='post-body entry-content'>
<p>sitting in my living room...
</div>
</div>
```

Παρατηρούμε οτι το κείμενο της καταχώρησης εσωκλείεται στο χαρακτηριστικό div με ετικέτα "post-body entry-content".

- Live Journal. Το Live Journal [2] είναι επίσης ένα εργαλείο δημιουργίας ιστολόγιων που χρησιμοποιείται απο εκατομμύρια χρήστες. Στην εικόνα 2.2 βλέπουμε ένα ιστολόγιο που δημιουργήθηκε αυτό το σύστημα.

[2] http://www.livejournal.com

Σε αυτην την περίπτωση, οι καταχωρήσεις του ιστολόγιου (περικλείονται με μαύρα πλαίσια) δεν είχαν κάποιες ετικέτες στον HTML κώδικά τους, παρα μόνο όσες εντολές ήταν απαραίτητες για την μορφοποίηση του κειμένου.

- Typepad[3], άλλο ένα γνωστό εργαλείο για την εύκολη δημιουργία και συντήρηση ιστολόγιων. Όντας ένα εμπορικό προϊόν, το Typepad προσφέρει αρκετά πιο εξελιγμένες δυνατότητες απο τα προηγούμενα δύο εργαλεία που αναφέραμε. Ένα παράδειγμα ιστολόγιου που παρήγαγε φαίνεται στην εικόνα 2.3. Ο κώδικας HTML της δεύτερης καταχώρησης είναι ο εξής:

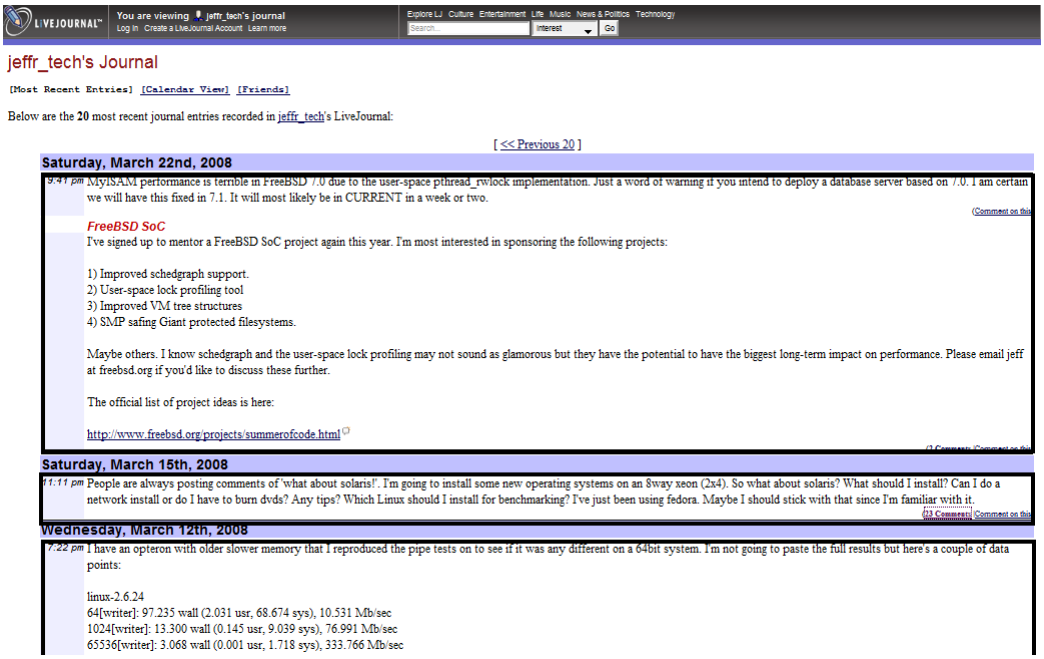
```
<div class="entry" id="entry-47351366">
<h3 class="entry-header"><a
 href="http://www.welovegifts.tv/2008/03/pd.html">
 Pink Digital Photo Frame with Calendar & Clock</a>
</h3>
<div class="entry-content">
<div class="entry-body">
<p><img title=..."</p>
<p>Also available in black...</p>
<p>Currently available...</p>
</div>
</div>
</div>
```

Στο ιστολόγιο αυτό τα περιεχόμενα των καταχωρήσεων περιέχονται ανάμεσα σε χαρακτηριστικά με ετικέτα “entry-body”.

- Wordpress[4], ένα σύστημα δημιουργίας ιστολόγιων το οποίο διατίθεται δωρεάν και είναι ελεύθερο λογισμικό.

- Serendipity[5], επίσης ένα λογισμικό δημιουργίας ιστολόγιων παρόμοιο με το Wordpress το οποίο όμως χρησιμοποιείται πιο σπάνια.

[3]http://www.typepad.com
[4]http://www.wordpress.com
[5]http://www.s9y.com

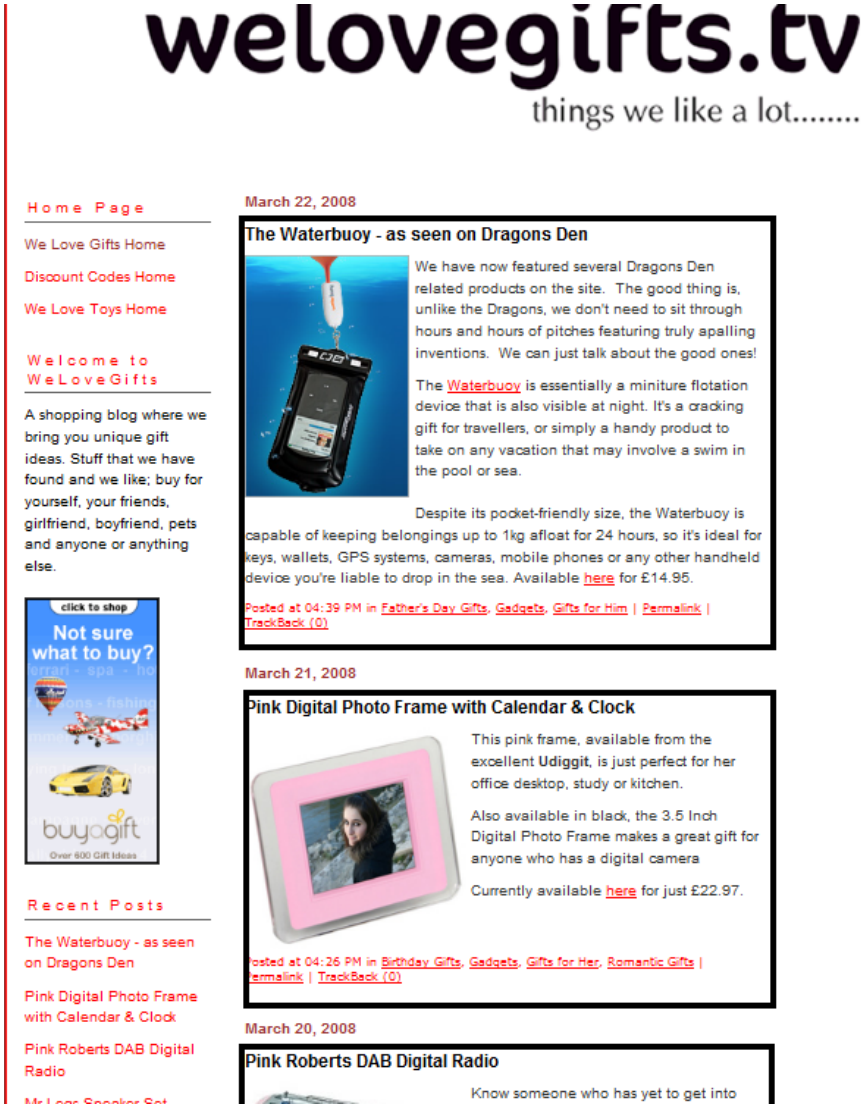


Σχήμα 2.2: Ιστολόγιο που έχει δημιουργηθεί με το σύστημα Livejournal. Οι καταχωρήσεις φαίνονται μέσα στα μαύρα πλαίσια.

2.2 Εργασίες αυτόματης εξαγωγής πληροφορίας απο ιστοσελίδες και ιστολόγια

Στο διαδίκτυο υπάρχει μεγάλο μέγεθος δομημένης πληροφορίας σε HTML μορφή. Παραδείγματα τέτοιας πληροφορίας είναι ειδήσεις, οικονομικά δεδομένα, ομιλίες, άρθρα, καθώς και τα περιεχόμενα των ιστολόγιων. Συνεπώς, έχουν προταθεί πολλές προσεγγίσεις στο πρόβλημα της αυτόματης εξαγωγής των δεδομένων αυτών [7]. Μία είναι η προσπάθεια να πεισθούν οι παροχείς και οι δημιουργοί των πληροφοριών να ενσωματώνουν μεταπληροφορίες οι οποίες θα περιγράφουν τα δεδομένα που υπάρχουν στις ιστοσελίδες τους και θα διευκολύνουν τον εντοπισμό τους και την αναζήτηση μεσα σε αυτά [8].

Μια εναλλακτική προσπάθεια ανάκτησης των δομημένων πληροφοριών είναι η δημιουργία αλγορίθμων που εκμεταλλεύονται την μορφοποίηση των ιστοσελίδων και προσπαθούν να διαχωρίσουν τα δεδομένα με βάση τις δομές στις οποίες υπάρχουν. Πρωταρχικής σημασίας είναι να αναπαρασταθούν τα δεδομένα της ιστοσελίδας σε μια εύκολα προσπελάσιμη μορφή η οποία θα διατηρεί την δομική πληροφορία της. Η επικρατούσα αναπαράσταση



Σχήμα 2.3: Ιστολόγιο που έχει δημιουργηθεί με το σύστημα Typepad. Οι καταχωρήσεις φαίνονται μέσα στα μαύρα πλαίσια.

στην επεξεργασία ιστοσελίδας πλέον είναι η δομή Document Object Model (DOM). Το DOM [9] είναι μια δομή δέντρου της οποίας κάθε φύλλο αντιστοιχεί ένα βασικό στοιχείο του html κώδικα της ιστοσελίδας. Με τη χρήση του η επεξεργασία της σελίδας διευκολύνεται σημαντικά, καθώς η πληροφορία των σχέσεων, της τοποθέτησης και των περιεχομένων των στοιχείων της είναι εύκολα προσβάσιμη και ο προγραμματιστής μπορεί να ασχοληθεί απευθείας με την επεξεργασία τους. Επίσης, υπάρχουν έτοιμες βιβλιοθήκες οι οποίες αναλαμβάνουν την δημιουργία του δέντρου DOM, και στο επόμενο κεφάλαιο θα δούμε πώς χρησιμοποιήθηκαν στην παρούσα εργασία.

Οι Debnath και άλλοι [10] περιγράφουν τον αλγόριθμο ContentExtractor ο οποίος εξάγει τις χρήσιμες καταχωρήσεις, τις οποίες καλούν και 'καταχωρήσεις πρωτεύοντος περιεχομένου' (Primary Content Blocks). Ο αλγόριθμος αυτός βασίζεται στο γεγονός οτι σε διαφορετικές ιστοσελίδες απο τον ίδιο ιστότοπο, τα δεδομένα τα οποία διαφέρουν απο τη μια σελίδα στην άλλη αλλά αλλάζουν με παρόμοιο τρόπο θα είναι αυτά που μας ενδιαφέρουν. Για παράδειγμα, εάν σε μια σελίδα χρηματιστηρίου συγκρίνουμε σελίδες που απεικονίζουν διαφορετικές μετοχές, το μεγαλύτερο ποσοστό της σελίδας θα παραμένει ίδιο, ενώ το τμήμα που απεικονίζει την εκάστοτε μετοχή θα είναι πάντοτε διαφορετικό για καθε σελίδα, και μάλιστα με παρόμοιο τρόπο. Έτσι, απομονώνοντας τα τμήματα αυτά έχουμε απομονώσει την πληροφορία που ζητάμε.

Πιο αναλυτικά, ο αλγόριθμος ContentExtractor λειτουργεί ως εξής: Αρχικά διαχωρίζεται η ιστοσελίδα σε τμήματα με βάση τα στοιχεία που ορίζουν την διαμόρφωση της, για παράδειγμα τους πίνακες. Έπειτα, εφαρμόζεται το κριτήριο απομόνωσης των κόμβων. Το κριτήριο αυτό βασίζεται σε κάποια κρίσιμη τιμή κατωφλίου (Threshold parameter). Κατόπιν υπολογίζουμε κατα πόσον διαφέρουν τα τμήματα τα οποία παραμένουν στην ίδια τοποθεσία απο σελίδα σε σελίδα. Εάν ο αλγόριθμος επιστρέψει αποτέλεσμα μεγαλύτερο της κρίσιμης τιμής, κρίνεται οτι τα τμήματα αυτά διαφέρουν αρκετά ωστε να απομονωθούν και να καταχωρηθούν ως χρήσιμη πληροφορία. Αντίθετα, τα δεδομένα τα οποία δεν διαφέρουν αρκετά κρίνεται οτι αποτελούν μη ενδιαφέρουσα πληροφορία.

Παρόμοιους αλγόριθμους παρουσιάζουν οι F. Douglis και T. Ball [11], καθώς και ο Breuel [12], ο οποίος συγκρίνει την ιεραρχία των κόμβων του DOM tree απο διαφορετικές ιστοσελίδες εφαρμόζοντας κατα πλάτος αναζήτηση, και εξάγει τους κόμβους που διαφέρουν.

Οι αλγόριθμοι αυτοί έχουν ένα κοινό χαρακτηριστικό: Βασίζονται στην

ομοιότητα (similarity) των τμημάτων αυτών σε διαφορετικές σελίδες απο την ίδια όμως κλάση σελίδων, συνήθως απο τον ίδιο ιστότοπο (domain). Κατ' επέκταση, προσφέρουν ικανοποιητικά ποσοστά εξαγωγής της χρήσιμης πληροφορίας απο σελίδες προερχόμενες απο μια πηγή οι οποίες μοιάζουν, αυτό όμως δεν ισχύει στα ιστολόγια. Το πεδίο των ιστολόγιων χαρακτηρίζεται απο τεράστια ανομοιογένεια, εκτεταμένες ιδιομορφίες καθώς και αρκετές προσωπικές πινελιές του εκάστοτε συγγραφέα, ακόμα και απο ιστολόγια τα οποία έχουν δημιουργηθεί με το ίδιο εργαλείο, ή στεγάζονται κατω απο τον ίδιο ιστότοπο.

Μία ενδιαφέρουσα μέθοδο παρουσίασαν οι Nanno και άλλοι [13] [14]. Η μέθοδος αυτή βασίζεται στο γεγονός οτι τα ενδιαφέροντα δεδομένα κάποιου ιστολόγιου θα είναι συνήθως διαχωρισμένα με βάση τις ημερομηνίες τους. Πράγματι, ως ιστοσελίδες που σκοπός τους είναι η κατηγοριοποίηση των δεδομένων που ορίζει ο εκάστοτε δημιουργός τους, τα ιστολόγια ικανοποιούν την προαπαιτούμενη αυτή ιδιότητα στην πλειοψηφία τους, χωρίς να λείπουν βέβαια και οι εξαιρέσεις από τον κανόνα όπως θα δούμε και στα αποτελέσματα της εφαρμογής που υλοποιήσαμε. Η εργασία τους χρησιμοποιεί μια μέθοδο εξαγωγής των καταχωρήσεων με βάση τις ημερομηνίες που πιθανώς θα υπάρχουν ανάμεσά τους. Η μέθοδος πραγματοποιεί τα εξής βήματα:

1. Αρχικά εντοπίζονται και απομονώνονται οι ημερομηνίες με χρήση κανονικών εκφράσεων (regular expressions).

2. Απομονώνονται οι κόμβοι με τις ημερομηνίες καθώς και οι ακριβείς τοποθεσίες τους στο δέντρο DOM της ιστοσελίδας.

3. Εξάγονται τα ενδιάμεσα δεδομένα, τα οποία και αποτελούν τις ζητούμενες καταχωρήσεις.

Στην παρούσα εργασία υλοποιήσαμε μια μέθοδο βασισμένη σε αυτή την ιδέα. Εκτός αυτής της μεθόδου, οι αλγόριθμοι που χρησιμοποιήσαμε, είναι περισσότερο ειδικευμένοι στα ιστολόγια και εκμεταλλεύονται κάποια χαρακτηριστικά που παρατηρούνται συχνά σε αυτά, όπως θα δούμε στην αναλυτική περιγραφή τους στο κεφάλαιο 4.

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

Η Μηχανή Διανυσμάτων Υποστήριξης (Support Vector Machine) [15] αποτελεί μια δημοφιλή μέθοδο μηχανικής εκμάθησης που έχει βρεί ευρεία εφαρμογή σε πολλά πρακτικά θέματα. Η Μηχανική Μάθηση (Machine Learning) είναι αναπόσπαστο κομμάτι της Τεχνητής Νοημοσύνης, καθώς η ικανότητα μάθησης αποτελεί το βασικότερο χαρακτηριστικό μιας οντότητας που θεωρείται «νοήμων» με την ευρύτερη έννοια του όρου. Αντλεί γνώση από διάφορα επιστημονικά πεδία και κυρίως από τη στατιστική, τη θεωρία πληροφορίας και τη γνωστική επιστήμη (cognitive science). Ένα σύστημα Μηχανικής Μάθησης βελτιώνεται αυτόματα με την απόκτηση εμπειρίας, προσαρμόζοντας κατάλληλα τη λειτουργία του σε αλληλεπίδραση με το περιβάλλον στο οποίο δραστηριοποιείται. Η κατασκευή μιας μηχανής με γενικευμένη δυνατότητα μάθησης αποτελεί για την ώρα ουτοπία, αλλά έχουν αναπτυχθεί πολλοί αλγόριθμοι που ακολουθούν διάφορες στρατηγικές μάθησης με αξιοσημείωτες επιδόσεις σε επιμέρους προβλήματα. Οι αλγόριθμοι Μηχανικής Μάθησης χρησιμοποιούνται σε πληθώρα επιστημονικών και εμπορικών εφαρμογών. Καθοριστική είναι η συμβολή τους στους κλάδους της Εξόρυξης Γνώσης από Δεδομένα (Data Mining), της Αναγνώρισης Ομιλίας (Speech Recognition) και της Αυτόματης Κατηγοριοποίησης Κειμένου, με την οποία θα ασχοληθούμε.

3.1 Ο αλγόριθμος SVM

Η Μηχανή Διανυσμάτων Υποστήριξης (Support Vector Machine) [15] αντιμετωπίζει το πρόβλημα της κατηγοριοποίησης επιλέγοντας έναν μικρό αριθμό στιγμιοτύπων εκπαίδευσης από κάθε κλάση, τα διανύσματα υποστήριξης

(support vectors), που συνορεύουν στο χώρο του προβλήματος με στιγμιότυπα άλλων κλάσεων. Με βάση τα επιλεγμένα αυτά διανύσματα κατασκευάζει ένα υπερεπίπεδο Ν διαστάσεων, οπου Ν ο αριθμός των χαρακτηριστικών του σώματος εκπαίδευσης, έτσι ώστε να επιτυγχάνεται ο βέλτιστος διαχωρισμός των στιγμιοτύπων εκπαίδευσης. Στην απλή περίπτωση των 2 διαστάσεων ο αλγόριθμος θα προσπαθήσει να βρει το βέλτιστο υπερεπίπεδο μίας διάστασης, δηλαδή μία γραμμή.

Δεδομένου ενός συνόλου εκπαίδευσης με ζευγάρια στιγμιοτύπων-ιδιοτήτων $(x_i, y_i), i = 1, ..., l$, οπου $x_i \in R^n$, η Μηχανή Διανυσμάτων Υποστήριξης απαιτεί την λύση του εξής προβλήματος:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \cdot \sum_{i=1}^{i=l} \xi_i$$

όταν

$$y_i(w^T \phi(x_i) + b) \geq 1 - j_i, j_i \geq 0$$

οπου $\frac{1}{2} w^T w$ το μέτρο της απόστασης μεταξύ επιπέδων που ορίζουν τα επίπεδα της κάθε κλάσης. Τα διανύσματα εκμάθησης x_i απεικονίζονται σε ένα χώρο πολλών (άπειρων, ενδεχομένως) διαστάσεων μέσω της συνάρτησης φ. Κατόπιν, ο αλγόριθμος εντοπίζει ένα γραμμικό υπερεπίπεδο το οποίο διαχωρίζει βέλτιστα τον χώρο αυτόν. Η παράμετρος C>0 είναι η παράμετρος σφάλματος, και η συνάρτηση $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ είναι η συνάρτηση πυρήνα (kernel function).

Οι βασικές συναρτήσεις πυρήνα είναι οι εξής:

- Γραμμική (linear): $K(x_i, x_j) = (x_i)^T x_j$.
- Πολυωνυμική (polynomial): $K(x_i, x_j) = (\gamma(x_i)^T x_j + r)^d, \gamma > 0$
- Ακτινωτής Βάσης (RBF): $K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0.$

Τα συστήματα ταξινόμησης που βασίζονται στον αλγόριθμο αυτό αποτελούν σήμερα μια από τις δημοφιλέστερες προσεγγίσεις στο χώρο της κατηγοριοποίησης κειμένου λόγω της αποτελεσματικότητας και της ταχύτητας που επιδεικνύουν καθώς και της δυνατότητας χειρισμού χώρων μεγάλης διάστασης. Προσφέρονται ιδιαίτερα για την επίλυση προβλημάτων μάθησης που δεν μπορούν να αντιμετωπιστούν από γραμμικά μοντέλα, καθώς μπορούν να παράγουν μη γραμμικές επιφάνειες απόφασης. Ένα ακόμα πλεονέκτημα

των SVM είναι η ικανότητά τους να χειρίζονται πολύ μεγάλους χώρους χαρακτηριστικών. Επίσης, αξιοσημείωτη είναι η ανεκτικότητα που παρουσιάζουν όσον αφορά στο πλήθος των στιγμιοτύπων εκπαίδευσης, ιδιαίτερα όταν αυτό διαφέρει μεταξύ των δύο κλάσεων, καθώς τα SVM (και γενικότερα οι ταξινομητές) δεν επιδιώκουν απλώς να ελαχιστοποιήσουν το σφάλμα στα δεδομένα εκπαίδευσης, αλλά να τα διαχωρίσουν αποτελεσματικά σε ένα χώρο μεγάλης διάστασης.

Κατά τη διαδικασία αξιολόγησης, τα δεδομένα χωρίζονται συνήθως σε δύο ξένα μεταξύ τους σύνολα: Το σύνολο εκπαίδευσης, που χρησιμοποιείται για την εκμάθηση του ταξινομητή και το σύνολο επικύρωσης (validation set) που χρησιμοποιείται για την εκτίμηση της αποτελεσματικότητας σε μελλοντικά παραδείγματα. Η τεχνική αυτή είναι γνωστή ως προσέγγιση εκπαίδευσης και επικύρωσης (train and validation). Μία εναλλακτική μέθοδος είναι η δοκιμή κ-πλής σταυρωτής επικύρωσης (k-fold cross-validation). Σύμφωνα με την προσέγγιση αυτή, το σύνολο των δεδομένων διαχωρίζεται σε κ ξεχωριστές ομάδες. Ο αλγόριθμος εκμάθησης τρέχει επίσης κ φορές, μια για κάθε ομάδα. Σε κάθε φάση τα δεδομένα απο την ομάδα αυτή θεωρούνται δεδομένα επικύρωσης ενώ τα υπόλοιπα κ-1 σύνολα χρησιμοποιούνται ως δεδομένα εκμάθησης. Τελικά, κάθε στιγμιότυπο των δεδομένων έχει ελεγχθεί ακριβώς μια φορά και το τελικό αποτέλεσμα είναι η ακρίβεια εκμάθησης όλου του συνόλου. Στα πειράματα της εργασίας μας στην ενότητα 6.4.4 αξιολογήσαμε και τις δύο αυτές μεθόδους.

Ο αλγόριθμος Διανυσμάτων Υποστήριξης απαιτεί κάθε στιγμιότυπο προς εκμάθηση ή ταξινόμηση να υφίσταται με μορφή διανύσματος πραγματικών αριθμών. Στην περίπτωσή μας, θα πρέπει οι τιμές των λέξεων και η κλάση στην οποια ανήκουν να αναπαρασταθούν σαν αριθμοί.

Ένα άλλο χαρακτηριστικό που χρήζει προσοχής είναι οτι συνήθως η απόδοση του αλγορίθμου SVM αυξάνεται εάν προηγηθεί η κανονικοποίηση (Normalisation) των δεδομένων. Κανονικοποίηση είναι η διαδικασία μεσω της οποίας οι τιμές των στοιχείων πολλαπλασιάζονται με κάποιον αριθμό με σκοπό το αποτέλεσμα να περιέχει τιμές μόνο εντός κάποιου διαστήματος, συνήθως [-1,1] ή [0,1]. Το κυριότερο πλεονέκτημα της διαδικασίας αυτής είναι οτι αποφεύγουμε την περίπτωση οπου στοιχεία με υψηλές τιμές κυριαρχούν έναντι αυτών με χαμηλότερες τιμές. Ένα άλλο πλεονέκτημα είναι οτι αποφεύγονται αριθμητικές δυσκολίες στην επεξεργασία των στοιχείων απο τις μαθηματικές ρουτίνες του αλγορίθμου, εάν κάποιες τιμές είναι ιδιαίτερα μεγάλες. Δεδομένου οτι δεν χάνεται ούτε αλλιώνεται καποιο μέρος της αρχικής

πληροφορίας των δεδομένων, είναι φρόνιμο σε κάθε περίπτωση να υφίσταται η κανονικοποίησή τους και μετά η περαιτέρω επεξεργασία τους, το οποίο και κάνουμε στα πειράματα της εργασίας αυτής.

Παράμετροι του αλγορίθμου SVM - libsvm

Για την κατηγοριοποίηση των καταχωρήσεων, χρησιμοποιήσαμε την βιβλιοθήκη libsvm [16] , η οποία προσφέρει μια υλοποίηση του αλγορίθμου Support Vector Machines καθώς και κάποια βοηθητικά προγράμματα για την ευκολία του χρήστη. Το κυρίως πρόγραμμα που χρησιμοποιήσαμε δέχεται ως είσοδο τα δεδομένα εκμάθησης ή ταξινόμησης και εκτελεί τις εξής λειτουργίες:

- Εκμάθηση του ταξινομητή με βαση τα εισαγόμενα δεδομένα
- Κατηγοριοποίηση των δεδομένων με τη χρήση του ταξινομητή
- Κ-πλή σταυρωτή επικύρωση στα εισαγόμενα δεδομένα.

Η libsvm προσφέρει και παραμετροποίηση των βασικών παραμέτρων του αλγορίθμου: Εκτός απο τις προεπιλεγμένες τιμές για τις παραμέτρους c, γ (οι οποίες είναι οι 1, 1/K αντίστοιχα) του πυρήνα του αλγόριθμου εκμάθησης μπορεί να δεχθεί και τιμές ορισμένες απο τον χρήστη. Επίσης μας δίνει την δυνατότητα να χρησιμοποιήσουμε κάποια εναλλακτική συνάρτηση πυρήνα (γραμμική, πολυωνυμική και RBF). Οι Hsu, Chang απέδειξαν [1] οτι ο πυρήνας RBF γενικά αποτελεί μια καλή λύση λόγω του οτι δεν αντιστοιχίζει γραμμικά τα στιγμιότυπα στον N-διάστατο χώρο, και σε αντίθεση με τον γραμμικό πυρήνα μπορεί να χειρίζεται περιπτώσεις οπου η σχέσεις κλάσεων και στιγμιοτύπων δεν είναι γραμμικές. Η εύρεση των πιο αποδοτικών αυτών τιμών συνήθως προκύπτει απο διαδοχικούς κύκλους εκπαίδευσης-αποτίμησης εως ότου καταλήξουμε στις βέλτιστες τιμές. Στην εργασία μας, όπως θα δούμε και στην ενότητα 6.4.4, πραγματοποιήσαμε πολλαπλά πειράματα με ποικίλες τιμές παραμέτρων και διαφορετικούς πυρήνες. Ευτυχώς, η κάπως επίπονη αυτή διαδικασία απλοποιείται σημαντικά με την χρήση ενός ακόμα χρήσιμου προγράμματος της βιβλιοθήκης libsvm, το οποίο αυτοματοποιεί τους κύκλους εκτέλεσης-δοκιμής και τελικά επιστρέφει τις παραμέτρους με την μέγιστη απόδοση, ενω επιπλέον δημιουργεί και καποια διαγράμματα για την οπτική απεικόνιση της απόδοσης του κατα τη διάρκεια των δοκιμών.

[1] A practical guide to SVM - http://www.csie.ntu.edu.tw/ cjlin

-	C_1	...	C_n	...	C_N
T_1	d_{11}	...	d_{1n}	...	d_{NN}
...	...	...	...	...	...
T_m	d_{m1}	...	d_{mn}	...	d_{mN}
...	...	...	...	...	...
T_M	d_{M1}	...	d_{Mn}	...	d_{MN}

Σχήμα 3.1: Πίνακας αντιστοίχησης κειμένων με θεματικές κατηγορίες

3.2 Αυτόματη κατηγοριοποίηση κειμένου

Η αυτόματη κατηγοριοποίηση ενός κειμένου σε κάποιες θεματικές κατηγορίες είναι μια διαδικασία μεσω της οποίας κρίνεται, με χρήση κάποιας μεθόδου μηχανικής μάθησης, σε ποιές απο τις διαθέσιμες κατηγορίες θα ενταχθεί το κείμενο.

Δεδομένου οτι έχουμε M κείμενα και N θεματικές κατηγορίες, το αποτέλεσμα της διαδικασίας μπορούμε να περιγραφεί με έναν M επί N πίνακα. Στον πίνακα αυτόν η κάθε γραμμή θα αντιστοιχεί σε ένα κείμενο, και η κάθε στήλη σε μια θεματική κατηγορία όπως φαίνεται στον πίνακα 3.1. Η κάθε τιμή d_{mn} του πίνακα θα είναι αληθής εάν το κείμενο m ανήκει στην κατηγορία n ή αλλιώς ψευδής.

Η αντιστοίχηση αυτή πραγματοποιείται μέσω μιας ιδανικής άγνωστης συνάρτησης $\phi : T \times C \rightarrow \{0, 1\}$, την οποία προσπαθούμε να προσεγγίσουμε μέσω της συνάρτησης $\phi' : T \times C \rightarrow \{0, 1\}$ η οποία καλείται ταξινομητής και αντιστοιχίζει τα έγγραφα στις κατηγορίες στις οποίες ανήκουν. Πεδίο ορισμού αυτής της συνάρτησης είναι ένα σύνολο οντοτήτων σε κάποια δεδομένη αναπαράσταση, η οποία αποτελεί το χώρο στιγμιοτύπων του προβλήματος. Η πλέον συνηθισμένη αναπαράσταση είναι αυτή που παρέχει το μοντέλο του διανυσματικού χώρου (vector space model), [17]. Σύμφωνα με αυτό το μοντέλο, τα στιγμιότυπα αναπαρίστανται ως διανύσματα, τα στοιχεία των οποίων αναπαριστούν τα χαρακτηριστικά του στιγμιότυπου που έχουν επιλεγεί ως σχετικά για το συγκεκριμένο πρόβλημα. Τα χαρακτηριστικά μπορούν να παίρνουν συμβολικές ή αριθμητικές τιμές.

Ο ταξινομητής θα πρέπει πάντα να βασίζεται στην ενδογενή γνώση του κειμένου και όχι σε πληροφορίες που προέρχονται απο εξωτερικές πηγές. Αυτό περιλαμβάνει και τυχόν μετα-πληροφορίες για το κείμενο, όπως π.χ. ο συγγραφέας ή ο τίτλος. Επίσης, ο ταξινομητής θα πρέπει να αντιμετωπίζει

τις κατηγορίες σαν απλούς συμβολισμούς και σε καμία περίπτωση δεν θα πρέπει να εκμεταλλεύεται καποια επιπλέον πληροφορία που να τις συσχετίζει με την σημασία του κειμένου, όπως π.χ. εαν η λέξη κάποιας κατηγορίας συμπεριλαμβάνεται σε κάποια κείμενα ή υποδηλώνει κάποια έκφραση.

Η διαδικασία ταξινόμησης είναι εν γένει υποκειμενική, καθώς σε ενα κείμενο δεν υπάρχει πληροφορία που να δίνει έναν αυστηρά μονοσήμαντο συσχετισμό μεταξύ της σημασιολογίας του και μιας θεματικής κατηγορίας. Το φαινόμενο αυτο παρατηρείται ακόμα και σε ανθρώπινο επίπεδο, οταν δύο άτομα εκφράζουν διαφορετικές ερμηνείες και απόψεις για το ίδιο ακριβώς κείμενο. Για παράδειγμα ένα κείμενο που σχολιάζει κάποια απεργιακά κινήματα μπορεί να καταταχθεί στα πολιτικά,οικονομικά,ασφαλιστικά και πιθανώς σε άλλα θέματα.

3.2.1 Κατηγοριοποίηση κειμένου με βάση την άποψη

Στόχος της εργασίας μας είναι να κατηγοριοποιηθούν κάποιες καταχωρήσεις που έχουν εξαχθεί απο ιστολόγια. Οι καταχωρήσεις αυτές περιέχουν όλες κριτικές, και στόχος είναι τις διαχωρίσουμε με βάση την άποψη την οποία εκφράζουν.

Στο σημείο αυτό σημειώνουμε οτι γενικότερα η άποψη κάποιου κειμένου είναι δύσκολο να οριστεί μεθοδολογικά. Μπορεί να είναι άμεση (implicit) ή έμμεση (explicit), για παράδειγμα η πρόταση ¨Θα μπορούσες να το αποφύγεις αυτό που έγινε¨ δηλώνει την έμμεση άποψη του συγγραφέα για το γεγονός, ενώ η πρόταση ¨Αυτό που έκανες είναι κάκιστο¨ δηλώνει άμεση άποψη για το θέμα. Ακόμα, η άποψη δεν είναι πάντα δύτιμη έννοια, και μπορεί να διαχωριστεί σε πολύ περισσότερες στάθμες απο δύο, για παράδειγμα [εξαιρετική / θετική / μέτρια / αδιάφορη / αρνητική / κατακεραυνωτική].

Η Ειρήνη Καλδέλη, στην διπλωματική της εργασία με τίτλο Έκπαίδευση ταξινομητών κειμένου για το χαρακτηρισμό άποψης' [18], πραγματοποιεί αρκετά πειράματα κατηγοριοποίησης κειμένων με βάση την άποψη χρησιμοποιώντας μεθόδους όπως γλωσσολογική ανάλυση και χρήση λεξικών υποκειμενικότητας. Οι σημαντικότερες μεταξύ αυτών των μεθόδων είναι οι εξής:

1. Ο συνυπολογισμός των όρων άρνησης: Η μέθοδος αυτή βασίζεται στη χρήση του αρνητικού νοήματος που προκύπτει απο συνήθεις λέξεις που υποδηλώνουν άρνηση για κάτι, όπως "no","not","didn't" κ.τ.λ. Η εμβέλεια των όρων άρνησης είναι δύσκολο να προσδιοριστεί επακριβώς, καθώς άλλοτε περιλαμβάνει μόνο την αμέσως επόμενη λέξη και άλλοτε

εκτείνεται σε περισσότερες, ακόμα και μέχρι το τέλος της πρότασης. Στην πράξη αποδείχθηκε οτι ο συνυπολογισμός των όρων άρνησης δεν προσφέρει κάποια βελτίωση στην ακρίβεια της κατηγοριοποίησης, μάλιστα η ορθότητα των αποτελεσμάτων μειώθηκε κατα μικρό ποσοστό. Δεδομένης της πολυπλοκότητας της μεθόδου δεν κρίθηκε σκόπιμο να την συμπεριλαμβάνουμε στα πειράματά μας.

2. Διαφορετική αναπαράσταση λεκτικών μονάδων, ο τρόπος δηλαδή με τον οποίο επιλέγονται οι λέξεις - χαρακτηριστικά των κειμένων ωστε να τροφοδοτήσουν τον αλγόριθμο μηχανικής μάθησης. Η αναπαράσταση των λέξεων ενδέχεται να είναι δυαδική, με τιμές συχνοτήτων καθώς και με τιμές TF/IDF. Η δυαδική αναπαράσταση παίρνει δύο τιμές, ανάλογα με τον αν υπάρχει ο εκάστοτε όρος στο κείμενο ή όχι. Η αναπαράσταση με τιμές συχνοτήτων αποθηκεύει την συχνότητα εμφάνισης του κάθε όρου στο κείμενο, και η αναπαράσταση με τιμές TF/IDF παρέχει ένα μέτρο της σημαντικότητας της κάθε λέξης σε σχέση με το κείμενο στο οποίο υπάρχει. Στα πειράματα τα οποία εκτελέσαμε συμπεριλήφθησαν και οι τρείς μέθοδοι.

3. Δοκιμή διαφόρων συναρτήσεων πυρήνα του αλγορίθμου SVM. Ο αλγόριθμος SVM μπορεί να υλοποιηθεί με διαφορετικές συναρτήσεις πυρήνα, όπως αναλύουμε στην επόμενη ενότητα. Στην εργασία μας πειραματιστήκαμε με τρείς συναρτήσεις πυρήνα.

Στα πλαίσια της εργασίας μας, και για λόγους απλότητας, θα θεωρήσουμε την άποψη σαν μια μεταβλητή δύο τιμών: Εάν η άποψη ενός κειμένου είναι θετική τότε αυτό μεταφράζεται στην τιμή '1', ενώ αν είναι αρνητική στην τιμή '0'.

Κεφάλαιο 4

Ανάλυση και σχεδίαση

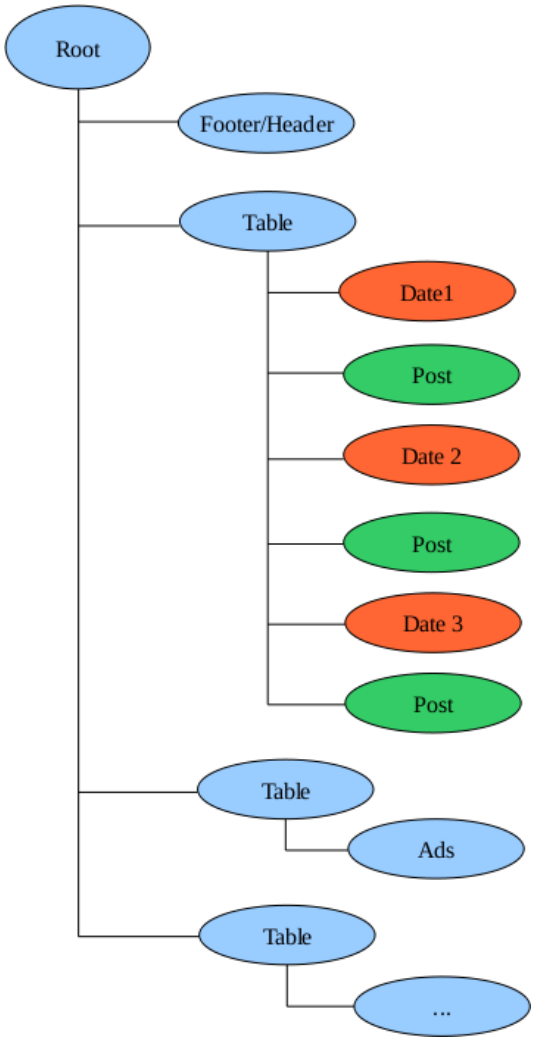
Σε αυτό το κεφάλαιο αρχικά γίνεται μια εισαγωγή στα γενικότερα χαρακτηριστικά των ιστολόγιων, και έπειτα επεξηγούνται αναλυτικά οι μέθοδοι που χρησιμοποιήθηκαν για την εξαγωγή των δεδομένων των καταχωρήσεών τους.

4.1 Ένα απλό ιστολόγιο

Ένα κοινό ιστολόγιο δέν είναι παρα μια ιστοσελίδα η οποία περιέχει, με κάποιο τρόπο, διαχωρισμένες τις πληροφορίες που έχει εισάγει ο χρήστης. Συνήθως το περιεχόμενο αυτό είναι απλό κείμενο, αρκετά όμως είναι και τα ιστολόγια τα οποία εμπεριέχουν φωτογραφίες και άλλου τύπου δεδομένα. Ένα παράδειγμα δέντρου DOM μιας τέτοιας υποθετικής ιστοσελίδας απεικονίζεται στην εικόνα 4.1.

Στην περίπτωση αυτή η εξαγωγή των καταχωρήσεων θα ήταν απλούστατη. Με μια απλή ανάλυση θα απομονώναμε τα δεδομένα μεταξύ των <post> και </post> tags, και τελικά θα είχαμε τις καταχωρήσεις που μας ενδιαφέρουν. Βέβαια, αυτό προϋποθέτει να γνωρίζουμε οτι οι καταχωρήσεις περιέχονται ανάμεσα στα <post> tags. Επειδή συνήθως τα ιστολόγια δημιουργούνται με βάση κάποια γνωστά εργαλεία, σαν αυτά που παρουσιάσαμε στο προηγούμενο κεφάλαιο, είμαστε σε θέση να γνωρίζουμε σημαντικό ποσοστό από τα χρησιμοποιούμενα tags.

Για τους σκοπούς της εργασίας, το ιστολόγιο το θεωρούμε ως μια ιστοσελίδα η οποία περιέχει κάποια δεδομένα διαχωρισμένα με βάση κάποια κριτήρια. Το μοναδικό κριτήριο είναι συνήθως η ημερομηνία, αλλά υπάρχουν και



Σχήμα 4.1: Απλουστευμένη απεικόνιση ενός υποθετικού δέντρου DOM ενός blog.

άλλες περιπτώσεις. Για παράδειγμα, σε κάποια ιστολόγια στα οποία συμμετέχουν πολλοί χρήστες στα κριτήρια διαχωρισμού θα συμπεριλαμβάνεται και το όνομα του χρήστη που προσέθεσε την εκάστοτε καταχώρηση. Μια πιθανή παραλλαγή του προηγούμενου παραδείγματος απεικονίζεται στην εικόνα 4.2.

4.2 Περίληψη μεθόδων κατάτμησης των ιστολόγιων

Υλοποιήθηκαν οι παρακάτω μέθοδοι που αναφέρονται σύμφωνα με την σειρά εφαρμογής τους στο κάθε ιστολόγιο:

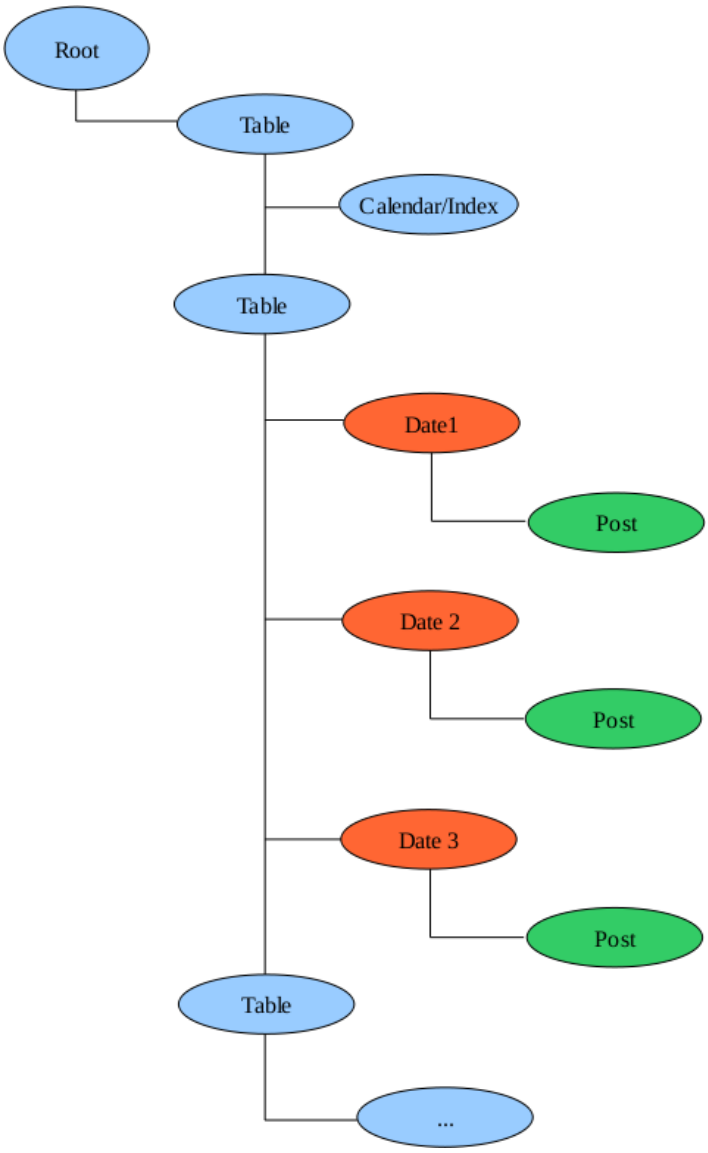
1. Ανάλυση των feeds με την μέθοδο FeedParser, η οποία διαχωρίζει τις καταχωρήσεις χρησιμοποιώντας τις πληροφορίες που μας παρέχουν.
2. Αποθήκευση (Caching) των ιστοσελίδων στο σκληρό δίσκο την πρώτη φορά που τις επισκεπτόμαστε, ωστε οι μετέπειτα προσπελάσεις τους να γίνουν απο τον σκληρό δίσκο ο οποίος είναι τάξεις μεγέθους ταχύτερος απο το διαδίκτυο.
3. Ανάλυση των αναγνωριστικών (tags) με την μέθοδο GeneratorScan. Η μέθοδος αυτη διαχωρίζει τις καταχωρήσεις με βάση τα αναγνωριστικά που προστίθενται στον HTML κώδικα απο το εκάστοτε εργαλείο δημιουργίας τους.
4. Διαχωρισμός με βάση τις ημερομηνίες με τον αλγόριθμο DateExtractor. Η μέθοδος αυτή διαχωρίζει τις καταχωρήσεις χρησιμοποιώντας σαν ενδιάμεσα όρια τις ημερομηνίες που παρεμβάλλονται μεταξύ τους.

Επίσης, ένα σχηματικό διάγραμμα της διαδικασίας που ακολουθείται στις ενότητες του παρόντος κεφαλαίου παρουσιάζεται στο σχήμα 4.3.

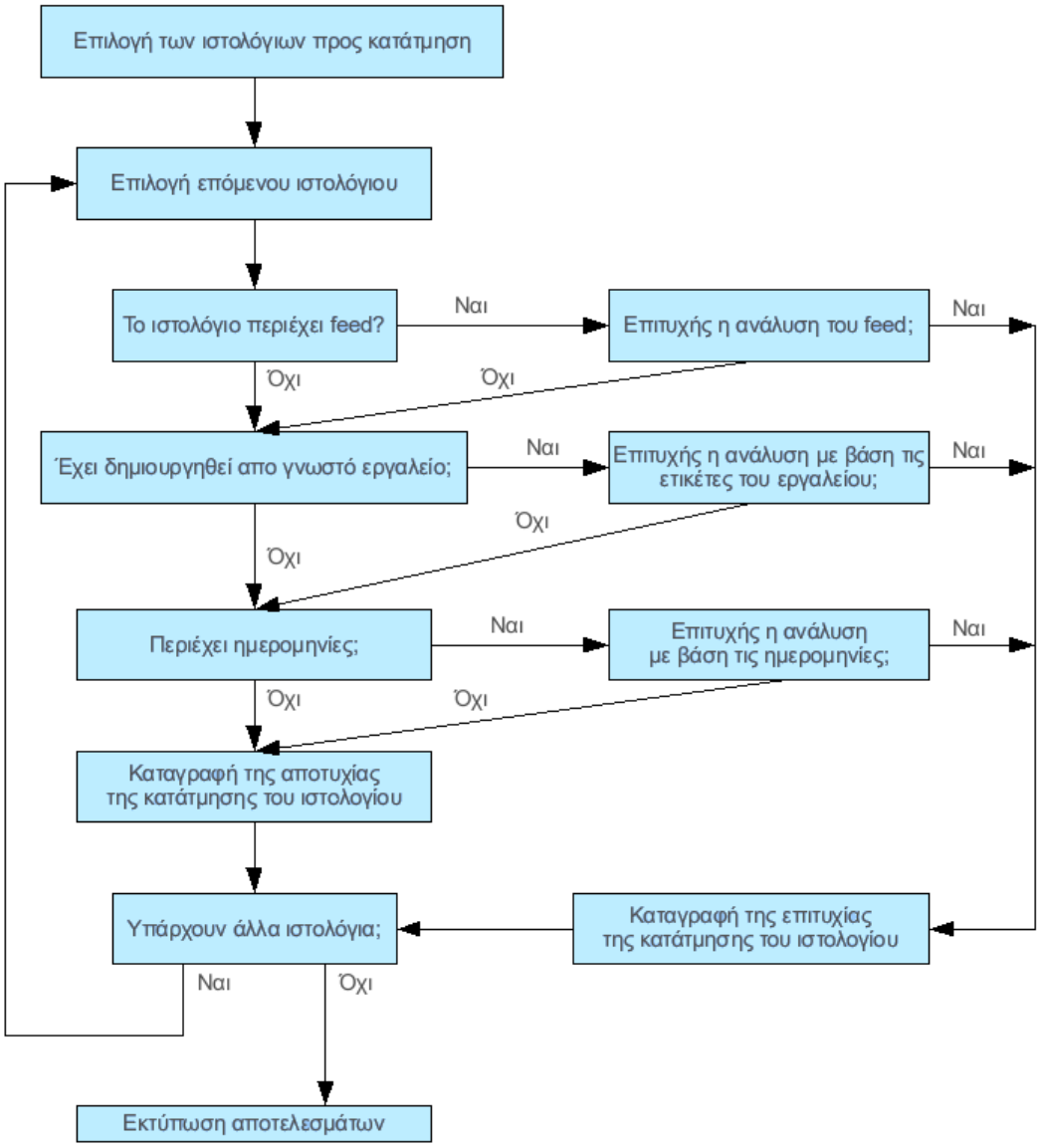
4.3 FeedParser - Αναζήτηση και ανάλυση των rss feeds

Η πρώτη μέθοδος αναζητά εάν η ιστοσελίδα έχει κάποια feeds που αναφέρονται στις καταχωρήσεις που μας ενδιαφέρουν. Τα feeds [1] είναι ένας αρκε-

[1] Για περισσότερες πληροφορίες δείτε το http:\\en.wikipedia.org\wiki\RSS_(file_format)



Σχήμα 4.2: Εναλλακτικό παράδειγμα του δέντρου DOM κάποιου υποθετικού blog.



Σχήμα 4.3: Σχηματική αναπαράσταση της διαδικασίας του διαχωρισμού των καταχωρήσεων

τά χρησιμοποιούμενος τρόπος αποθήκευσης δεδομένων σε XML μορφή, και χρησιμοποιείται συνήθως σαν ευρετήριο σε σελίδες με συχνές ενημερώσεις των δεδομένων τους. Η χρησιμότητά τους έγκειται στο οτι μπορεί ο χρήστης μέσω κάποιου προγράμματος ανάγνωσης (rss feed aggregator) να διαβάσει τα χρήσιμα δεδομένα της σελίδας χωρίς να την επισκευθεί μέσω του browser του, κερδίζοντας έτσι χρόνο καθώς έχει την δυνατότητα να ενημερώνεται αυτόματα για πιθανές αλλαγές και ανανεώσεις των περιεχομένων της. Στην περίπτωσή μας, αρκετά ιστολόγια έχουν τέτοια feeds που περιέχουν διαχωρισμένες τις καταχωρήσεις και έτσι η εξαγωγή τους επιτυγχάνεται εύκολα με μια απλή ανάλυση των feeds.

Την περίοδο που πρωτοχρησιμοποιήθηκαν τα feeds, είχαν δημιουργηθεί αρκετά ασύμβατα είδη καθώς και αρκετοί διαφορετικοί τρόποι προσπέλασής τους. Το γεγονός αυτό καθιστούσε δυσχερή την αυτόματη ανάγνωσή τους απο ένα πρόγραμμα καθώς έπρεπε να υλοποιηθούν ξεχωριστές μέθοδοι για κάθε πιθανή διαμόρφωση. Με την πάροδο του χρόνου επικρατήσανε οι εξής διαφορετικές μορφοποιήσεις:

- rss 1.0 που είναι και το παλαιότερο πρωτόκολλο
- rss 2.0 (μια αναβάθμιση του προηγουμένου)
- atom (ασύμβατο με τα παραπάνω)

Είναι αναγκαίο να μπορούμε να προσπελάσουμε και τις τρείς αυτές βασικές υλοποιήσεις με διαφανή ως προς την εφαρμογή τρόπο, καθώς είναι άγνωστο εκ των προτέρων ποιά χρησιμοποιείται. Για την ανάλυση και τον διαμερισμό των feeds χρησιμοποιήθηκε η βιβλιοθήκη Rome[2], η οποία έχει την ικανότητα να αναγνωρίζει όλες τις μορφές των feeds, και να εξασφαλίζει σωστό αποτέλεσμα για κάθε διαθέσιμη έκδοση. Ο τρόπος που χρησιμοποιείται είναι ανεξάρτητος του είδους του feed, και απο την οπτική γωνία του προγραμματιστή είναι μια κομψή και καθαρή λύση.

Όσον αφορά την υλοποίηση της ανάλυσης, τα βήματα είναι τα εξής:

- Έλεγχος για ύπαρξη ενός ή και παραπάνω feed και εύρεση του σωστού, καθώς ενδέχεται να υπάρχουν και άλλα τα οποία περιέχουν άσχετη πληροφορία (όπως ευρετήρια άλλων ιστολογίων ή κατάλογοι με συνδέσμους προς άλλα ιστολόγια). Καθώς ο κώδικας της ιστοσελίδας περιέχει

[2]http:\\rome.dev.java.net

τα URLs των feeds, θα πρέπει να επιλέξουμε το σωστό URL του feed που περιέχει τις καταχωρήσεις.

- Προσπέλαση και εξαγωγή των περιεχομένων του feed.
- Αντιμετώπιση της περίπτωσης που το feed συμπεριλαμβάνει μέρος των καταχωρήσεων και όχι ολόκληρα τα δεδομένα. Σε πολλές περιπτώσεις τα feeds δεν περιέχουν την κάθε καταχώριση ολόκληρη, παρα μόνο τις 2-3 πρώτες γραμμές τους. Καθώς εμείς θέλουμε ολόκληρη την καταχώριση, υλοποιήσαμε μια μέθοδο που εξάγει την πλήρη καταχώριση χρησιμοποιώντας τα δεδομένα του feed.

Στις επόμενες υποενότητες παρουσιάζουμε την προσέγγιση που αναπτύξαμε για καθένα απο τα παραπάνω βήματα.

4.3.1 Έλεγχος για ύπαρξη και εύρεση του feed

Για να διαπιστώσουμε την ύπαρξη ή όχι του feed, εκμεταλλευόμαστε το οτι τα feeds συνήθως συμπεριλαμβάνονται στις σελίδες με τη μορφή ενός συνδέσμου (link) στον κώδικα της σελίδας. Αρχικά επιθυμούμε να διαχωρίσουμε τους συνδέσμους που οδηγούν σε feeds απο τους υπόλοιπους. Για το σκοπό αυτό δημιουργούμε μια λίστα με σκοπό να αποθηκεύσουμε εκεί τις διευθύνσεις των feeds, και πραγματοποιούμε μια σειρά από ελέγχους:

1. Αν η σελίδα περιέχει σύνδεσμο σε κάποια άλλη ιστοσελίδα (target link), και η σελίδα-προορισμός περιέχει στην αρχή την συμβολοακολουθία “rss”, τότε πιθανότατα αυτός θα είναι σύνδεσμος σε rss feed και τον προσθέτουμε στην αντίστοιχη λίστα.
2. Αν η σελίδα περιέχει σύνδεσμο με όνομα που εμπεριέχει την συμβολοακολουθία “rss”, τότε προσθέτουμε και αυτόν τον σύνδεσμο.
3. Εάν η σελίδα περιέχει πλήθος συνδέσμων που συμπεριλαμβάνονται στις παραπάνω περιπτώσεις, τότε απλά επιλέγουμε έναν τυχαίο με προτεραιότητα όσους στεγάζονται στον ίδιο ιστότοπο (“domain”) με την ιστοσελίδα. Συνήθως περιέχονται πολλά feeds που περιέχουν τις καταχωρήσεις, απλά είναι διαφορετικού τύπου για συμβατότητα με περισσότερα προγράμματα ανάγνωσης. Καθώς εμείς έχουμε την δυνατότητα να αναλύσουμε παντός τύπου feed, δεν μας ενδιαφέρει κάποιο συγκεκριμένο, συνεπώς επιλέγουμε τυχαία.

4.3.2 Εξαγωγή των περιεχομένων του feed

Σε δεύτερη φάση εξάγουμε τα περιεχόμενα του feed που έχουμε εντοπίσει απο το προηγούμενο βήμα. Η εξαγωγή ανάγεται στο να ξεχωρίσουμε ποιά απο τα πεδία δεδομένων περιέχουν το κείμενο των καταχωρήσεων. Κατά σύμβαση, το όνομα του πεδίου των περιεχομένων που μας επιστρέφουν οι ρουτίνες της βιβλιοθήκης είναι κοινό, και έτσι η διαδικασία διεκπεραιώνεται χωρίς κόπο μέσω των έτοιμων μεθόδων της βιβλιοθήκης που χρησιμοποιούμε.

4.3.3 Εύρεση του πλήρους κειμένου των καταχωρήσεων

Το τρίτο βήμα προκύπτει από το γεγονός οτι αρκετά feeds δέν περιέχουν ολόκληρες τις καταχωρήσεις, και συνήθως αρκούνται στο να παρουσιάσουν μια συνοπτική μορφή από κάθε καταχώρηση, συνήθως την πρώτη σειρά ή την πρώτη παράγραφο. Προφανώς αυτό γίνεται για οικονομία χώρου και με το σκεπτικό οτι αν ο χρήστης ενδιαφερθεί περαιτέρω για το περιεχόμενο, θα επισκευτεί την ιστοσελίδα για να διαβάσει το υπόλοιπο. Απο την μεριά μας αυτό είναι ελλιπές, και έτσι υλοποιήθηκε η δυνατότητα να εντοπίζονται με βάση τα ήδη υπάρχοντα και ελλιπή δεδομένα τα αντίστοιχα πλήρη. Ο αλγόριθμος που αναλαμβάνει αυτήν την διεργασία είναι ο εξής:

1. Έλεγχος εάν τα δεδομένα του feed είναι ελλιπή. Κατά σύμβαση, τα ελλιπή feeds περιέχουν καταχωρήσεις που τελειώνουν με την συμβολοσειρά '...', και έτσι μπορούμε να γνωρίζουμε πότε είναι ελλιπής η καταχώρηση και πότε όχι.

2. Για κάθε καταχώρηση που τελειώνει με '...':

   - Αναζητούμε ποιό μέρος της ιστοσελίδας αρχίζει με την ίδια συμβολοσειρά όπως και η καταχώρηση αυτή, δηλαδή ποιά είναι η αντίστοιχη πλήρης καταχώρηση στην ιστοσελίδα απο την οποία προήλθε και η ελλιπής καταχώρηση του feed.

   - Εξάγουμε τον κόμβο της ιστοσελίδας που περιέχει την πλήρη αυτή καταχώρηση και τον αποθηκεύουμε αντί της αρχικής ελλιπής καταχώρησης.

   Εάν δέν είναι ελλιπής η καταχώρηση τότε εξάγουμε αυτούσιο το περιεχόμενο του feed.

Στο σημείο αυτό έχουμε πραγματοποιήσει την ανάλυση που αφορά τα feeds. Εάν έχει βρεθεί και αναλυθεί το σωστό feed, τότε ο διαχωρισμός των καταχωρήσεων έχει ολοκληρωθεί. Στην περίπτωση που δέν βρέθηκε κάποιο, η εφαρμογή προχωρά στην επόμενη μέθοδο.

4.4 Caching της ιστοσελίδας

Τα rss feeds δέν περιέχονται μέσα στον κώδικα της ιστοσελίδας, αλλά σε ξεχωριστά αρχεία στον εξυπηρετητή (server) στον οποίο στεγάζεται ιστολόγιο. Για τις μεθόδους που παρουσιάζονται στις επόμενες υποενότητες, δέν χρειαζόμαστε τίποτε άλλο παρά τον ίδιο τον κώδικα της σελίδας που περιέχει και τις καταχωρήσεις. Ο κώδικας αυτός θα προσπελασθεί πολλές φορές καθώς απαιτείται η ανάλυσή του σε διάφορα στάδια. Η επαναλαμβανόμενη ανάκτηση της σελίδας από την διαδικτυακή της τοποθεσία οδηγεί σε μεγάλη σπατάλη χρόνου και διαθέσιμου εύρους ζώνης, τόσο από την μεριά μας όσο και από την μεριά του εξυπηρετητή που φιλοξενεί την ιστοσελίδα. Συνεπώς, μας συμφέρει να κατεβάσουμε την ιστοσελίδα και να την αποθηκεύσουμε τοπικά, καθώς έτσι:

- Κάνουμε μεγάλη εξοικονόμηση χρόνου σε διαδοχικές προσπελάσεις, διότη η τοπική προσπέλαση είναι τάξεις μεγέθους γρηγορότερη.
- Μας δίνεται η δυνατότητα να δημιουργήσουμε ένα μεγάλο σύνολο ιστολόγιων τα οποία μπορούν να χρησιμεύσουν ως σύνολο δοκιμών για περαιτέρω αναλύσεις.

Για αυτούς τους λόγους υλοποιήθηκε το caching των ιστοσελίδων που προέρχονται από το διαδίκτυο. Ο όρος Caching χρησιμοποιείται γενικότερα σαν μια διαδικασία η οποία αποθηκεύει σε κάποιο γρήγορο μέσο συχνά χρησιμοποιούμενες πληροφορίες. Με αυτόν τον τρόπο, η προσπέλασή τους πραγματοποιείται πολύ γρηγορότερα χωρίς να είναι αναγκαία η επαναλαμβανόμενη ανάκτησή τους με τον πρώτο (συνήθως αργό) τρόπο.

Η εφαρμογή αποθηκεύει την ιστοσελίδα του ιστολόγιου σε κάποιον τοπικό κατάλογο στον σκληρό δίσκο. Το όνομα αποθήκευσης είναι το ίδιο με τη διαδικτυακή διεύθυνση του ιστολόγιου, με την προσθήκη της συμβολοσειράς '.parsed' στο τέλος της κατάληξης του αρχείου. Για παράδειγμα, το http://www.blog.com θα αποθηκευτεί ως www.blog.com.parsed. Εάν κάποιο ιστολόγιο υπάρχει τοπικά με το ίδιο όνομα, τότε η εφαρμογή προσπερνά

την αποθήκευσή του εξοικονομώντας ακόμα περισσότερο χρόνο. Αυτό είναι βολικό καθώς μας δίνεται η δυνατότητα να έχουμε μία συνεχώς εμπλουτιζόμενη λίστα από αποθηκευμένα ιστολόγια, και κάθε φορά να περιορίζονται οι προσπελάσεις στο διαδίκτυο μόνο σε σελίδες τις οποίες δεν έχουμε επισκευθεί έως τώρα.

Καθώς μια ιστοσελίδα ενδέχεται να έχει ανανεωθεί απο την τελευταία φορά που την ανακτήσαμε, η χρήση του caching θα πρέπει να περιορίζεται σε διαδικασίες με βραχυπρόθεσμα ενδιάμεσα διαστήματα. Εάν έχει μεσολαβήσει μεγάλο διάστημα απο την τελευταία ανάκτηση το caching θα πρέπει να απενεργοποιηθεί ωστε η εφαρμογή να ανακτήσει τις νεότερες εκδόσεις των ιστοσελίδων.

4.5 Έλεγχος και ανάλυση των αναγνωριστικών - Μέθοδος GeneratorScan

Πριν αρχίσουμε να περιγράφουμε την μέθοδο, σκόπιμο είναι να περιγράψουμε λίγο την δομή κάποιας ιστοσελίδας και τις δομές δεδομένων με τις οποίες εργαζομαστε. Η ανάλυση πραγματοποιείται με την βοήθεια της βιβλιοθήκης Htmlparser [3], η οποία είναι η πιο πλήρης βιβλιοθήκη όσον αφορά συντακτική ανάλυση ιστοσελίδων στην Java. Η συγκεκριμένη βιβλιοθήκη παρέχει έτοιμες μεθόδους ανάλυσης του HTML κώδικα της σελίδας, καθώς και δομές δεδομένων που απλουστεύουν σημαντικά την πρόσβαση σε αυτές. Η βασική δομή που μας προσφέρει η βιβλιοθήκη Htmlparser για την επεξεργασία του HTML κώδικα είναι ο κόμβος (node). Κόμβος μπορεί να είναι ένα οποιοδήποτε βασικό πεδίο της σελίδας, όπως μία συμβολοσειρά, ένας πίνακας, μια εικόνα και γενικότερα οτιδήποτε μπορεί να περιγραφεί μέσω αναγνωριστικών (tags) της γλώσσας HTML. Η όλη ιστοσελίδα αποθηκεύεται σε μία δενδρική δομή από κόμβους, το δέντρο DOM, με αρχικό κόμβο (ρίζα) το πρωταρχικό "<html>" tag που προηγείται πάντα του κώδικα μιας ιστοσελίδας. Κατόπιν αποθηκεύονται οι κόμβοι-παιδιά του, οι οποίοι συνήθως είναι τα ”title” και ”body”, μετά τα παιδιά αυτών και ούτω καθεξής. Η προσπέλαση με αυτό τον τρόπο καθίσταται εύκολη, καθώς αρκεί να διατρέξουμε το δέντρο των κόμβων.

Η δεύτερη μέθοδος, η οποία είναι και η πιο απλή από τις τρεις, αναζητά μέσα στον html κώδικα του ιστολόγιου αναγνωριστικά τα οποία γνωρίζουμε ο-

[3] http://htmlparser.sourceforge.net/

τι διαχωρίζουν τις καταχωρήσεις μεταξύ τους, δηλαδή μεσολαβούν μεταξύ των δεδομένων των καταχωρήσεων και του κειμένου ή του κώδικα της υπόλοιπης ιστοσελίδας. Εάν βρεθούν τέτοια αναγνωριστικά, απλά εξάγει το περιεχόμενο που περικλείουν και έτσι έχουμε στην διάθεσή μας τα δεδομένα της κάθε καταχώρησης. Τα αναγνωριστικά αυτά τα έχουμε προσδιορίσει εξετάζοντας τον κώδικα που παράγεται απο τα γνωστά εργαλεία δημιουργίας ιστολόγιων. Η λίστα μπορεί να μήν είναι εξαντλητική, αλλά περιλαμβάνει τα πιο γνωστά συστήματα τα οποία χρησιμοποιούνται στην πλειοψηφία των ιστολόγιων που συναντήσαμε. Τα αναγνωριστικά περιέχονται σε ένα αρχείο το οποίο διαβάζει το πρόγραμμα της εφαρμογής κατα την εκκίνηση. Με αυτόν τον τρόπο καθίσταται εύκολη η προσθήκη ενός νέου εργαλείου και των χαρακτηριστικών αναγνωριστικών του χωρίς να παρέμβουμε καθόλου στον κώδικα του προγράμματος, γεγονός που αυξάνει την επεκτασιμότητα του προγράμματος. Το αρχείο αυτό έχει ονομασθεί "generators" και βρίσκεται στον ίδιο κατάλογο που τρέχει η εφαρμογή, από όπου και γίνεται η ανάγνωσή του κάθε φορά που αυτή αρχίζει. Η διαμόρφωση του αρχείου είναι η εξής:

```
*
<blogger>
<post>
<postcore>
...
*
<typepad>
<post>
<mainpost>
...
```

Η πρώτη συμβολοσειρά μετα απο κάθε αστερίσκο αντιστοιχεί σε ένα εργαλείο δημιουργίας ιστολόγιων το οποίο χρησιμοποιεί τα αναγνωριστικά που ακολουθούν κατόπιν, μέχρι και τον επόμενο αστερίσκο. Όλα δηλαδή τα tags που ξέρουμε εκ των προτέρων οτι χρησιμοποιεί το εργαλείο αυτό για να διαχωρίσει τις καταχωρήσεις. Κατά σύμβαση, τα εργαλεία που χρησιμοποιούνται στον σχεδιασμό ιστοσελίδων αποθηκεύουν το όνομά τους σαν συμβολοσειρά στον HTML κώδικα μέσα σε ένα αναγνωριστικό με το όνομα “generator”. Διαβάζοντας αυτήν την συμβολοσειρά μπορούμε να καταλάβουμε με ποιό εργαλείο έχει δημιουργηθεί η ιστοσελίδα του ιστολόγιου [4]. Εάν κάποιο εργαλείο

[4]Να σημειωθεί οτι μπορεί κάλλιστα μια σελίδα να έχει δημιουργηθεί με κάποιο εργαλείο

αντιστοιχεί στη συμβολοσειρά με το “generator” tag που υπάρχει στην ιστοσελίδα, αυτό σημαίνει οτι η σελίδα αυτή δημιουργήθηκε με το συγκεριμένο εργαλείο.

Τα βήματα που ακολουθεί η μέθοδος αυτή είναι τα εξής:

1. Ανάγνωση της ιστοσελίδας που αποθηκεύτηκε προηγουμένως μέσω της λειτουργίας caching.

2. Έλεγχος εάν δημιουργήθηκε με κάποιο γνωστό σύστημα.

3. Εάν δημιουργήθηκε με κάποιο γνωστό σύστημα, έλεγχος ύπαρξης κάποιου γνωστού αναγνωριστικού διαχωρισμού του συστήματος αυτού.

4. Εάν κάποιο αναγνωριστικό υπάρχει, τότε προχωράμε στην εξαγωγή των δεδομένων που περικλείονται ανάμεσα στους κόμβους που περιέχουν αναγνωριστικά τέτοιου τύπου.

Ο διαχωρισμός γίνεται με έναν αλγόριθμο που αναζητά όλους τους κόμβους του dom tree που περιέχουν το tag που μας ενδιαφέρει εκτελώντας κατα πλάτος αναζήτηση, και αποθηκεύει το περιεχόμενό τους σαν ξεχωριστές καταχωρήσεις. Εάν δεν βρεθεί κάποιο αναγνωριστικό, το πρόγραμμα προχωρά στην επόμενη μέθοδο.

Ας σημειώσουμε οτι στην περίπτωση που δέν υπάρχει το αναγνωριστικό “generator” στην σελίδα, ή υπάρχει μεν αλλά δέν έχει ευρεθεί κανένα αναγνωριστικό διαχωρισμού από το αντίστοιχο γνωστό σύνολο, το πρόγραμμα θα προχωρήσει στην επόμενη μέθοδο. Οι περιπτώσεις αυτές είναι πιο συχνές από ό,τι αναμένεται, και συνήθως προκύπτουν λόγω της μεγάλης παραμετροποίησης των τελικών ιστοσελίδων από τον χρήστη, μετά την αρχική τους σχεδίαση.

4.6 Ανάγνωση Ημερομηνιών

Σε αυτό το σημείο αναλύεται η τρίτη και τελευταία μέθοδος που χρησιμοποιεί η εφαρμογή ωστε να διαχωρίσει τις καταχωρήσεις.

Το οτι έχουμε φτάσει έως εδώ σημαίνει οτι οι προηγούμενες μέθοδοι έχουν αποτύχει. Συνεπώς, ισχύουν τα εξής όσον αφορά την ιστοσελίδα προς ανάλυση:

και να μήν περιέχει παρόλαυτά το συγκεκριμένο tag. Συνήθως αυτό επαφίεται στην επιλογή του χρήστη, αν και τα περισσότερα εργαλεία το τοποθετούν ερήμην.

- Δεν περιέχει κάποια ένδειξη οτι δημιουργήθηκε από γνωστό εργαλείο.
- Δεν περιγράφει την πληροφορία εσωκλείοντας την σε γνωστά tags.
- Δεν προσφέρει κάποιο feed για την εύκολη εύρεση των καταχωρήσεων.

Η μέθοδος αυτή βασίζεται στο γεγονός οτι τα ενδιαφέροντα δεδομένα ενός ιστολογίου μπορούν να διαχωριστούν με βάση τις ημερομηνίες καταχώρησής τους. Εξ ορισμού, ως ιστοσελίδες που σκοπός τους είναι η κατηγοριοποίηση των δεδομένων που ορίζει ο εκάστοτε δημιουργός τους, τα ιστολόγια ικανοποιούν την προαπαιτούμενη αυτή ιδιότητα στην πλειοψηφία τους. Δεν λείπουν βέβαια και οι εξαιρέσεις από τον κανόνα.

Η μέθοδος αυτή εξάγει τα δεδομένα των κόμβων που εμπεριέχονται ανάμεσα σε ημερομηνίες, ακολουθώντας τα παρακάτω βήματα:

1. Αναγνώριση των ημερομηνιών μέσα στο κείμενο με χρήση κανονικών εκφράσεων (regular expressions).
2. Εξαγωγή των κόμβων που περιέχουν αυτές τις ημερομηνίες, καθώς και του κοντινότερου κοινού πατρικού κόμβου.
3. Απο το σύνολο των πατρικών κόμβων υπολογισμός του μεγέθους του περιεχομένου που βρίσκεται ανάμεσα στις ημερομηνίες.
4. Επιλογή του πατρικού κόμβου με το μεγαλύτερο μέγεθος περιεχομένων (κόμβων-παιδιών).
5. Εξαγωγή των δεδομένων που βρίσκονται μεταξύ των ημερομηνιών, τα οποία περιέχονται σε κόμβους-παιδιά του προηγουμένως επιλεγμένου κόμβου.

Όταν τελειώσει η μέθοδος τελικά θα έχουμε απομονώσει τον πατρικό κόμβο που περιέχει την ομάδα ημερομηνιών με το μεγαλύτερο μέγεθος περιεχομένων. Οι επιμέρους μέθοδοι εξηγούνται αναλυτικά στις επόμενες υποενότητες.

4.6.1 Αναγνώριση ημερομηνιών με χρήση κανονικών εκφράσεων

Το πρώτο βήμα του αλγορίθμου είναι η επιτυχής αναγνώριση των ημερομηνιών που υπάρχουν στην ιστοσελίδα. Για τον σκοπό αυτό χρησιμοποιήθηκαν κανονικές εκφράσεις, οι οποίες αναγνωρίζουν με υψηλό ποσοστό

επιτυχίας τις ημερομηνίες σε διάφορες μορφές. Οι κανονικές εκφράσεις που χρησιμοποιούμε αναγνωρίζουν ημερομηνίες οι οποίες ενδέχεται να είναι αριθμητικές με τις εξής μορφές:

- mm/dd/yyyy
- dd/mm/yyyy
- mm/dd/yy
- dd/mm/yy
- m/d/y
- d/m/y

ή αλφαριθμητικές όπως η εξής μορφή

- dd <month> yyyy (π.χ. 23 Dec 2006)

Ένα πρόβλημα που προκύπτει σε αυτό το σημείο είναι η αναγνώριση των ημερομηνιών που είναι γραμμένες σε γλώσσα διαφορετική από την αγγλική. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με την εισαγωγή επιπλέον κανονικών εκφράσεων που να συμπεριλαμβάνουν τις αντίστοιχες λέξεις για τους μήνες.

4.6.2 Εξαγωγή των κόμβων που περιέχουν τις ημερομηνίες

Σε αυτό το βήμα αναλύουμε την ιστοσελίδα ώστε να ξεχωρίσουμε τους κόμβους που περιέχουν τις ημερομηνίες, καθώς και τον πατρικό τους κόμβο. Η διαδικασία αυτή χρησιμεύει ώστε να εντοπίσουμε το σύνολο δεδομένων που μας ενδιαφέρει και να το απομονώσουμε από την υπόλοιπη πληροφορία που εμπεριέχεται στην ιστοσελίδα. Απο το προηγούμενο βήμα έχουμε μία λίστα με όλους τους κόμβους που περιέχουν μια ημερομηνία. Εάν η αρχική υπόθεσή μας είναι σωστή, μέσα από όλο αυτό το σύνολο θα υπάρχει ένα υποσύνολο κόμβων που θα διαχωρίζουν τις δημοσιεύσεις. Οι κόμβοι αυτοί θα έχουν έναν κοινό κόμβο-πατέρα ο οποίος θα περιέχει και τις καταχωρήσεις. Ακολουθεί ένα ενδεικτικό παράδειγμα που εξηγεί την έννοια του κόμβου-πατέρα. Ας υποθέσουμε οτι έχουμε τον εξής κώδικα μιας ιστοσελίδας (υπεραπλοποιημένος για οικονομία χώρου):

```
<head>
...
<body>
...
<table>
      <date>...</date>
      <post>...</post>
      <date>...</date>
      <post>...</post>
      ...
</table>
...
</body>
```

Σε αυτό το απλοϊκό παράδειγμα, ο κόμβος -πατέρας είναι ο 'table' ο οποίος υποδηλώνει οτι οι καταχωρήσεις είναι αποθηκευμένες στα κελιά μίας δομής τύπου 'table', πιθανότατα για μια ομοιόμορφη απεικόνιση στον browser του χρήστη. Βεβαίως, είναι πιθανό να υπάρχουν πολύ περισσότεροι κόμβοι όπως εντολές μορφοποίησης ή επιπλέον tags μέσα στους κόμβους με τις ημερομηνίες και τις καταχωρήσεις. Καθώς μας ενδιαφέρουν μόνο οι ημερομηνίες, οι υπόλοιποι κόμβοι αγνοούνται. Στα αποτελέσματα του διαχωρισμού, όσα δεδομένα υπήρχαν ανάμεσα στις καταχωρήσεις θα συμπεριληφθούν στα HTML δεδομένα των διαχωρισμένων καταχωρήσεων, ενώ τα υπόλοιπα απλά θα απορριφθούν.

Ο αλγόριθμος εύρεσης κοινού πατέρα λειτουργεί ώς εξής:

1. Εύρεση όλων των κόμβων που περιέχουν συμβολοακολουθία που συμφωνεί με τις κανονικές εκφράσεις των ημερομηνιών. Αυτοί οι κόμβοι θα είναι ολες οι ημερομηνίες που περιέχει η ιστοσελίδα. Μια απλοποιημένη εκδοχή της μεθόδου περιγράφεται σε ψευδοκώδικα ως εξής:

```
nodes=get all HTML nodes from HtmlParser;
dates=empty List of size N;
for node i in nodes do
    for regex in dateRegexes do
        if regex matches node i then
            insert node in dates;
        end
    end
end
```

2. Διαχώρισε τους κόμβους σε σύνολα με βάση τον πλησιέστερο κοινό πατρικό κόμβο που έχουν,και απέρριψε κόμβους που περιέχουν γνωστά υποσύνολα με κόμβους - πατέρες. Ο αντίστοιχος ψευδοκώδικας είναι ο εξής:

```
parentNodes = dates;
groupParentNodes = parentNodes;
for groupParentNode in groupParentNodes do
    repeat
        tempParentNode = the parent node of groupParentNode;
        insert tempParentNode in groupParentNodes;
    until tempParentNode is <body> ;
end
remove parentNodes from groupParentNodes;
for groupParentNode a in groupParentNodes do
    for groupParentNode b in groupParentNodes do
        if a is parent of b then
            remove a from groupParentNodes;
        end
    end
end
remove duplicates from groupParentNodes;
```

Να σημειώσουμε οτι μας ενδιαφέρει ο *πλησιέστερος* κοινός πατέρας: Πάντα θα υπάρχει ένας κοινός πατέρας για όλους τους κόμβους, και στην ακραία περίπτωση αυτός θα είναι ο κόμβος "body", ο οποίος εξ' ορισμού ορίζει το περιεχόμενο μιας ιστοσελίδας. Εάν όμως επιλέξουμε κάποιον άλλο πατρικό κόμβο εκτός απο τον πλησιέστερο, τόσο μεγαλύτερη πιθανότητα υπάρχει να

συμπεριλάβουμε άχρηστες πληροφορίες μαζί με τις πληροφορίες των καταχωρήσεων, καθώς πληθαίνουν οι κόμβοι-παιδιά που δεν αφορούν τις καταχωρήσεις. Ακριβώς γιαυτό το λόγο εφαρμόζεται το τρίτο βήμα στον παραπάνω αλγόριθμο. Για παράδειγμα, έστω οτι υπάρχουν δύο υποσύνολα κόμβων ημερομηνιών Α,Β με 5 και 10 ημερομηνίες αντίστοιχα. Έστω επίσης οτι ο πατρικός κόμβος που περιέχει το σύνολο Α είναι ο ΠΑ και το σύνολο Β ο ΠΒ. Όσο ανεβαίνουμε επίπεδο στο DOM tree, κάποια στιγμή θα βρούμε έναν κόμβο ΠΓ ο οποίος θα είναι γονέας του ΠΑ και του ΠΒ. Θα ήταν όμως λάθος να δημιουργήσουμε ένα νέο σύνολο 15 ημερομηνιών με πατρικό κόμβο τον ΠΓ, καθώς οι ημερομηνίες που περιέχονται έχουν ήδη συμπεριληφθεί στα συνολα Α,Β. Επομένως, όποιος κόμβος περιέχει μόνο γνωστά υποσύνολα ημερομηνιών απορρίπτεται.

4.6.3 Επιλογή του πατρικού κόμβου με το μεγαλύτερο μέγεθος περιεχομένων

Πολλές φορές τα ιστολόγια περιέχουν επιπλέον ημερομηνίες εκτός απο αυτές που ανήκουν στις καταχωρήσεις, όπως ένα ημερολόγιο ή ένα ευρετήριο (index) ωστε ο χρήστης να επιλέγει τις καταχωρήσεις με βάση την ημερομηνία τους. Από το προηγούμενο βήμα έχουμε ήδη όλες τις ημερομηνίες και τους πατρικούς τους κόμβους. Ένας εξ αυτών θα είναι ο σωστός και θα περιέχει τις καταχωρήσεις ως κόμβους-παιδιά, και οι υπόλοιποι θα είναι λανθασμένοι και θα περιέχουν αδιάφορα δεδομένα. Η επιλογή του σωστού κόμβου γίνεται σε αυτό το βήμα. Το κριτήριο που μας οδηγεί είναι οτι εφόσον ο ζητούμενος κόμβος-πατέρας περιέχει τις καταχωρήσεις, το μέγεθος του κειμένου των παιδιών του θα είναι κατα κανόνα μεγαλύτερο από το μέγεθος των υπολοίπων. Ακριβέστερα, ως 'μέγεθος' ορίζουμε τον όγκο του κειμένου που περιέχεται σε όλους τους κόμβους που ανήκουν στο δέντρο με ρίζα τον κόμβο-πατέρα, και το υπολογίζουμε αναδρομικά. Σε αυτό το σημείο τίθεται το ερώτημα κατά πόσον ισχύει η υπόθεση οτι ο κόμβος με το μεγαλύτερο περιεχόμενο μέγεθος θα είναι και αυτός που περιέχει τις καταχωρήσεις. Με βάση τα όσα ιστολόγια έχουμε δεί μέχρι τώρα η υπόθεση αυτή ευσταθεί, καθώς οι συνηθέστερες λειτουργίες που μπορούν να περιέχουν ημερομηνίες και ταυτόχρονα ανάμεσα τους κείμενο θα είναι είτε ημερολόγια είτε ευρετήρια. Και στις δύο περιπτώσεις το μέγεθος του κειμένου είναι σημαντικά μικρότερο απο το συνολικό κείμενο των καταχωρήσεων.

Συνοψίζοντας, ο αλγόριθμος σε αυτό το βήμα είναι ο εξής:

1. Για κάθε κόμβο-πατέρα υπολόγισε αναδρομικά το συνολικό μέγεθος του κειμένου που περιέχουν οι κόμβοι-παιδιά του.

2. Επέλεξε τον κόμβο-πατέρα με το μεγαλύτερο περιεχόμενο μέγεθος.

4.6.4 Εξαγωγή των δεδομένων που βρίσκονται μεταξύ των ημερομηνιών

Σε αυτό το σημείο έχουμε τον σωστό κόμβο-πατέρα που περιέχει τις καταχωρήσεις, και το μόνο που απομένει είναι να εξάγουμε τα δεδομένα που παρεμβάλλονται ανάμεσα στις ημερομηνίες. Η ανάλυση γίνεται διατρέχοντας κατά βάθος το δέντρο των κόμβων. Απο την στιγμή που θα βρεθεί μια ημερομηνία η εφαρμογή θεωρεί ως καταχώρηση το κείμενο που συναντά έως την επόμενη ημερομηνία. Στο σημείο αυτό θεωρεί οτι το επόμενο κείμενο θα είναι της δεύτερης καταχώρησης, και ούτω καθεξής.

Στο πέρας αυτής της μεθόδου έχουμε αναλύσει τις ημερομηνίες της ιστοσελίδας και έχουμε εξάγει τις καταχωρήσεις που εμπεριέχονται ανάμεσά τους. Η επιτυχία της μεθόδου βασίζεται σε δύο βασικές υποθέσεις:

1. Οι καταχωρήσεις διαχωρίζονται με ημερομηνίες. Αυτό είναι η βασική μας υπόθεση, εφόσον εάν αυτή δέν ισχύει η εξαγωγή των καταχωρήσεων δέν θα είναι δυνατή. Στην πλειοψηφία των ιστολόγιων διαπιστώσαμε οτι υπάρχουν ημερομηνίες.

2. Οι ημερομηνίες είναι είτε αριθμητικές είτε αλφαριθμητικές, και σε μορφή που αναγνωρίζονται απο τις κανονικές εκφράσεις που αναφέραμε προηγουμένως.

Η αναγνώριση των ημερομηνιών δεν περιορίζεται μόνο στην αγγλική γλώσσα, καθώς η προσθήκη μιας γλώσσας δεν χρειάζεται κάτι παραπάνω απο το να προσθέσουμε τις κατάλληλες κανονικές εκφράσεις στο αντίστοιχο αρχείο. Στην εργασία μας προσθέσαμε κανονικές εκφράσεις για την ελληνική καθώς και την ισπανική γλώσσα. Επίσης, το ποσοστό εντοπισμού των ημερομηνιών έχει την δυνατότητα να βελτιωθεί αρκετά με την προσθήκη περισσότερο εξειδικευμένων regular expressions, καθώς και με κάποιο πρόγραμμα κανονικοποίησης ημερομηνιών.

4.7 Περιγραφή της διαδικασίας ταξινόμησης των καταχωρήσεων

Στο κεφάλαιο αυτό περιγράφουμε τις εντολές και τα προγράμματα που χρησιμοποιήσαμε για την ταξινόμηση των καταχωρήσεων που εξάγονται απο τα ιστολόγια.

Οι καταχωρήσεις και τα κείμενα που θα επεξεργαστούμε θα πρέπει να είναι αποθηκευμένα σε έναν κατάλογο ως αρχεία κειμένου. Επειδή για τις ανάγκες της εκπαίδευσης θα πρέπει να γνωρίζουμε την κλάση που ανήκει το κάθε κείμενο, κατα σύμβαση το όνομά τους θα πρέπει να αρχίζει με "neg" εάν η κλάση είναι αρνητική ή "pos" εάν είναι θετική. Έτσι, το πρόγραμμα θα αποθηκεύσει και την κατάλληλη κλάση του κάθε κειμένου στο τελικό αρχείο με τα διανύσματα χαρακτηριστικών. Καθώς πραγματοποιήσαμε δύο είδη πειραμάτων, ένα με τιμές συχνότητας των λέξεων των κειμένων και ένα με τιμές TF/IDF, δημιουργήθηκαν δύο scripts τα οποία δέχονται ως όρισμα τον κατάλογο με τα αρχεία προς επεξεργασία και εξάγουν ως αποτέλεσμα τα δεδομένα με τα οποία θα τροφοδοτηθεί ο αλγόριθμος ταξινόμησης.

Τα προγράμματα αυτά αναλαμβάνουν να εκτελέσουν τα εξής βήματα:

- Ανάγνωση του δοσμένου καταλόγου και εύρεση όλων των αρχείων που περιέχονται.
- Εξαγωγή των λέξεων (tokens) απο κάθε αρχείο.
- Δημιουργία των τιμών προς αποθήκευση (συχνότητες λέξεων ή τιμές tf/idf). Σε αυτο το σημείο μπορούμε να χρησιμοποιήσουμε ένα ήδη υπάρχον λεξικό.
- Εξαγωγή των δεδομένων στην έξοδο του προγράμματος.
- Αποθήκευση του λεξικού σε αρχείο, εάν αυτό δίνεται (προεραιτική δυνατότητα).

Η αποθήκευση του λεξικού των χαρακτηριστικών σε αρχείο είναι αναγκαία εαν ο ταξινομητής εκπαιδευτεί σε διαφορετικό σώμα απο το σώμα κατηγοριοποίησης, όταν δηλαδή δεν χρησιμοποιήσουμε σταυρωτή επικύρωση. Καθώς τα δύο σώματα θα πρέπει να έχουν παρόμοια αναπαράσταση των διανυσμάτων υποστήριξής τους, με την τελευταία λειτουργία ο ταξινομητής αποθηκεύει τα διανύσματα χαρακτηριστικών έτσι ώστε οι τιμές τους να συμπίπτουν.

Στο επόμενο βήμα θα πρέπει να κανονικοποιήσουμε τα δεδομένα που παρήχθησαν προηγουμένως. Χρησιμοποιούμε το κατάλληλο εργαλείο για αυτην την διαδικασία το οποίο είναι μέρος της libsvm και ονομάζεται "svm-scale". Ως είσοδο δέχεται τις παραμέτρους του διαστήματος κανονικοποίησης και το αρχείο δεδομένων προς κανονικοποίηση, και στην έξοδό του παράγει τα κανονικοποιημένα δεδομένα.

Κατόπιν, τροφοδοτούμε το πρόγραμμα της libsvm που εκτελεί την ταξινόμηση, 'svm-train', με τα κανονικοποιημένα δεδομένα. Αυτό γίνεται τρέχοντας το εκτελέσιμο με όρισμα το αρχείο δεδομένων απο το προηγούμενο βήμα. Η έξοδος θα είναι τα αποτελέσματα της κατηγοριοποίησης των αρχικών κειμένων.

Ακολουθεί ένα παράδειγμα ωστε να γίνει η συνολική διαδικασία κατανοητή. Τα αρχεία κειμένου προς ταξινόμηση τα έχουμε στον κατάλογο "corpusdir", και εκτελούμε δοκιμές 5-πλής σταυρωτής επικύρωσης:

1. Δημιουργία σώματος εκπαίδευσης απο τον κατάλογο με της καταχωρήσεις:

   ```
   $ ./gen-training corpusdir >train.1
   $ ./gen-training-tfidf corpusdir >train.2
   ```

   Το αρχείο 'train.1 περιέχει τιμές με αναπαράσταση συχνοτήτων, ενώ το 'train.2' περιέχει TF/IDF τιμές.

2. Κανονικοποίηση των αρχείων με τις αναπαραστάσεις των διανυσμάτων υποστήριξης:

   ```
   $ svm-scale -l 0 -u 1 train.1 >train.1.scaled
   $ svm-scale -l 0 -u 1 train.2 >train.2.scaled
   ```

   Τα αρχεία 'train.1, 'train.1 περιέχουν τις κανονικοποιημένες τιμές απο το βήμα 1.

3. Εφαρμογή 5-πλής σταυρωτής επικύρωσης στα κανονικοποιημένα δεδομένα:

   ```
   $ svm-train -v 5 train.1
   ...
   Accuracy: 55%
   ```

```
$ svm-train -v 5 train.2
...
Accuracy: 85%
```

Μετά απο κάθε εντολή βλέπουμε την ακρίβεια των πειραμάτων μας.

Κεφάλαιο 5

Υλοποίηση

5.1 Η εφαρμογή ανάλυσης των ιστολόγιων

Εφαρμόζοντας στην πράξη τις μεθόδους διαχωρισμού των ιστολόγιων που περιγράφονται στο προηγούμενο κεφάλαιο, υλοποιήσαμε μια εφαρμογή με γραφικό περιβάλλον εργασίας η οποία διευκολύνει την μαζική ανάλυση των ιστολόγιων, και φροντίζει για τον υπολογισμό των στατιστικών στοιχείων της αξιολόγησης. Η εφαρμογή είναι γραμμένη σε Java. Χρησιμοποιήθηκε η έκδοση 5 αυτής, η οποία είναι και η προτεινόμενη έκδοση, αλλά έχει δοκιμαστεί και με την νεότερη έκδοση 6 δίχως προβλήματα.

5.1.1 Εγκατάσταση και προαπαιτούμενα στοιχεία

Για να τρέξει η εφαρμογή, αποσυμπιέζουμε σε κάποιον κατάλογο της αρεσκείας μας το εσωκλειώμενο αρχείο "thesis.zip", πηγαίνουμε σε αυτόν και:

- Σε Microsoft Windows περιβάλλον εκτελούμε το "thesis.bat"
- Σε UNIX περιβάλλον, εκτελούμε το αρχείο "thesis.sh"

  Η εφαρμογή χρησιμοποιεί τις βιβλιοθήκες Rome και HTMLparser, οι οποίες για λόγους ευκολίας ενσωματώθηκαν στο αρχείο "thesis.zip" της εφαρμογής. Αυτό ήταν δυνατό καθώς καλύπτονται απο άδειες ελεύθερου λογισμικού που επιτρέπουν την ελεύθερη διακίνηση τους. Έτσι, ο χρήστης δεν χρειάζεται να κάνει κατι άλλο για να τρέξει την εφαρμογή εκτός απο το να αποσυμπιέσει το αρχείο αυτό.

5.1.2 Διαμόρφωση της εφαρμογής και συνοδευτικά αρχεία ρυθμίσεων

Η εφαρμογή έχει τα εξής προαπαιτούμενα αρχεία, τα οποία θα πρέπει να βρίσκονται στον κατάλογο dist/ απο τον οποίο τρέχουμε και την εφαρμογή:

- 'parsed/', ο κατάλογος στον οποίο αποθηκεύονται τα αποτελέσματα. Για κάθε ιστοσελίδα που αναλύεται, δημιουργούνται τρία αρχεία σε αυτόν τον κατάλογο (όπου '<blog>' το URL της εκάστοτε ιστοσελίδας) :

  1. '<blog>.parsed.html', το αρχείο στο οποίο αποθηκεύεται η HTML πληροφορία των καταχωρήσεων.
  2. '<blog>.parsed.txt', το αρχείο στο οποίο αποθηκεύεται μόνο το κείμενο των καταχωρήσεων, ωστε να γίνει εύκολη η κατηγοριοποίησή του μετά.
  3. '<blog>.txt', το αρχείο που περιέχει την διαδικασία που ακολουθήθηκε απο την εφαρμογή και ποιές μέθοδοι εφαρμόστηκαν κατα την διάρκεια της ανάλυσής του ιστολόγιου.

- 'cached/', ο κατάλογος στον οποίο αποθηκεύονται όλα τα .html αρχεία που η εφαρμογή προσπελάζει στο διαδίκτυο.

- 'regexes.txt', το αρχείο που περιέχει τις κανονικές εκφράσεις για τις ημερομηνίες, μία ανα γραμμή.

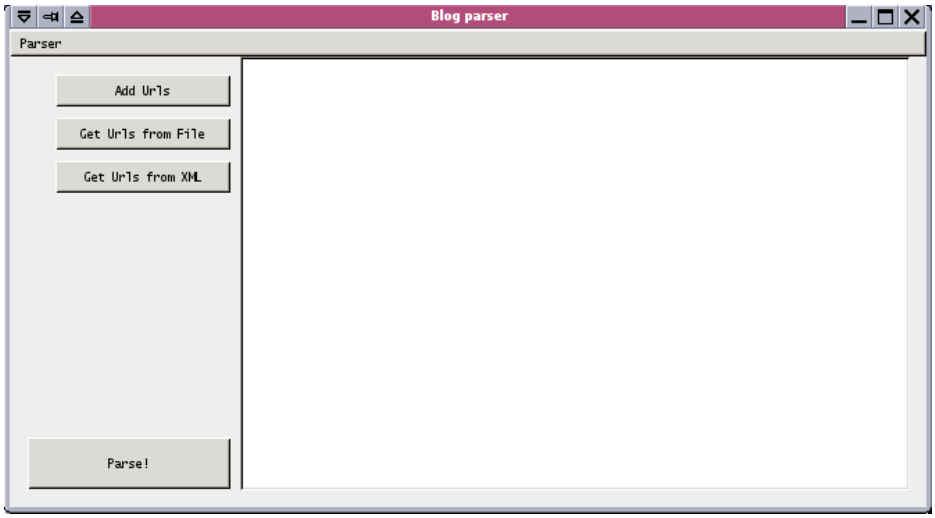
- 'config.txt', το αρχείο που περιέχει τις αρχικές ρυθμίσεις του προγράμματος.

- 'generators.txt', το αρχείο που περιέχει τις ετικέτες και τους generators στους οποίους αντιστοιχούν.

Οι κατάλογοι αυτοί θα πρέπει να δημιουργηθούν και να είναι άδειοι πριν εκτελεσθεί η εφαρμογή για πρώτη φορά.

Επίσης, η εφαρμογή είναι σε θέση να αναλύει τα αρχεία που είχαμε στην διάθεσή μας απο την εταιρεία intelliseek. Τα αρχεία αυτά είναι σε XML μορφή με την εξής μορφοποίηση:

```
<blog>
<url>http://...</url>
```



Σχήμα 5.1: Το αρχικό παράθυρο της εφαρμογής

```
<post>...</post>
<post>...</post>
...
</blog>
```

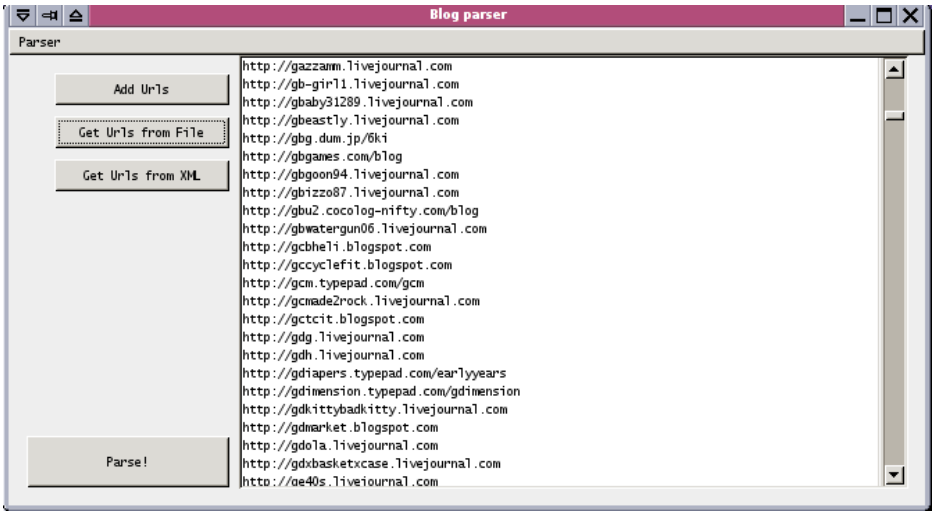
και τα URLs που εξάγονται αποθηκεύονται σε ένα αρχείο κειμένου, ένα σε κάθε γραμμή.

Στην εικόνα 5.1 απεικονίζεται ένα στιγμιότυπο του κεντρικού παραθύρου της εφαρμογής, το οποίο εμφανίζεται μόλις φορτώσει το πρόγραμμα.

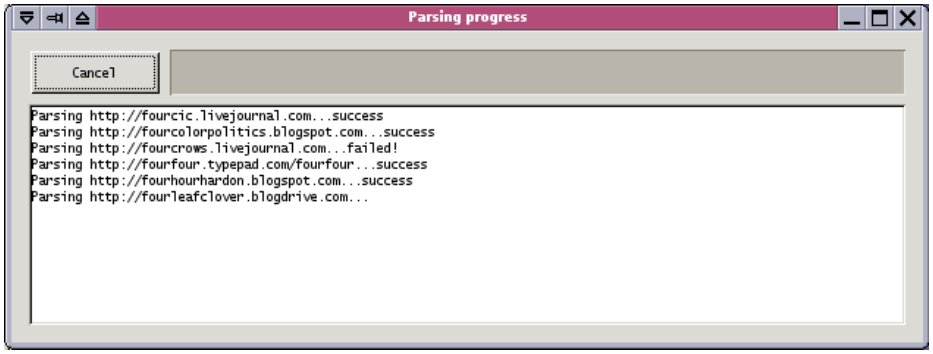
5.1.3 Εισαγωγή των URLs των ιστολόγιων

Επιλέγοντας 'Add urls', ο χρήστης καλείται να εισάγει την διεύθυνση μιας ιστοσελίδας που θέλει να αναλύσει. Η διεύθυνση θα προστεθεί στην λίστα δεξιά, η οποία περιέχει όλα τα URLs προς ανάλυση. Επειδή συνήθως θέλουμε να αναλύσουμε μεγάλο αριθμό ιστοσελίδων, είναι καλύτερα να αποθηκεύουμε τα URLs σε ένα αρχείο. Επιλέγοντας 'Get Urls from File', μπορούμε να επιλέξουμε ένα αρχείο το οποίο θα περιέχει μια λίστα με τα URLs των ιστολόγιων που θέλουμε. Η λίστα αυτή θα πρέπει να περιέχει ένα URL ανα γραμμή ωστε να μπορέσει το πρόγραμμα να τις αναγνωρίσει. Μόλις επιλεχθούν, θα εισαχθούν στη λίστα δεξιά, όπως φαίνεται στην εικόνα 5.2

Η επιλογή 'Get Urls from XML' δέν προσθέτει κάποια ιστολόγια για ανάλυση. Η λειτουργία που προσφέρει είναι να εξάγει τα URLs των ιστολόγιων



Σχήμα 5.2: Μετά την προσθήκη URLS προς ανάλυση



Σχήμα 5.3: Η πρόοδος κατα τη διάρκεια της ανάλυσης

από τα έτοιμα αρχεία XML τα οποία έχουμε στη διάθεσή μας απο την εταιρεία Intelliseek, μορφοποιημένα όπως αναγράφεται στην ενότητα 5.1.2.

5.1.4 Εκκίνηση ανάλυσης και εμφάνιση αποτελεσμάτων

Όταν ο χρήστης έχει επιλέξει όσα ιστολόγια θέλει να αναλύσει, θα πρέπει να πατήσει στο τελευταίο κουμπί που αναγράφει “Parse”. Το πρόγραμμα τότε θα δημιουργήσει ένα νέο παράθυρο, όπως απεικονίζεται στην εικόνα 5.3, το οποίο δίνει πληροφορίες όσο διαρκεί η εργασία της ανάλυσης:

Επιπλέον εμφανίζεται μια μπάρα προόδου, η οποία δείχνει το ποσοστό της συνολικής διαδικασίας που έχει διεκπεραιωθεί. Επίσης, αναγράφονται τα ιστολόγια τα οποία έχουν αναλυθεί, εάν η ανάλυση έγινε επιτυχώς ή όχι,

και στην τελευταία περίπτωση εάν η αποτυχία οφείλεται σε άκυρη διεύθυνση URL ή σε αδυναμία του προγράμματος να διαχωρίσει τις καταχωρήσεις. Εάν ο χρήστης θελήσει να ακυρώσει την διαδικασία, τότε θα πρέπει να επιλέξει ακύρωση ("Cancel").

Σε κάθε περίπτωση, είτε η διαδικασία τελειώσει κανονικά είτε ο χρήστης την διακόψει, όποια αποτελέσματα υπήρξαν μέχρι το σημείο αυτό αποθηκεύονται στο αρχείο με όνομα "results.txt", στον ίδιο κατάλογο απο όπου τρέξαμε το πρόγραμμα.

Κεφάλαιο 6

Έλεγχος

Σε αυτό το κεφάλαιο αξιολογούμε την εφαρμογή σε πραγματικά δεδομένα. Αρχικά εξηγούμε πώς συλλέξαμε τα ιστολόγια που χρησιμοποιήθηκαν και εξάγουμε τις καταχωρήσεις τους . Κατόπιν, κατηγοριοποιούμε τις καταχωρήσεις αυτές με βάση την άποψή τους και σχολιάζουμε τα αποτελέσματα.

6.1 Εύρεση μεγάλου αριθμού ιστολόγιων για τις δοκιμές μας

Σαν σύνολο δοκιμών χρησιμοποιήθηκε ένας μεγάλος αριθμός από ιστολόγια: 7857 ιστολόγια αναλύθηκαν από το πρόγραμμα στη διάρκεια της εκπόνησης της διπλωματικής εργασίας. Καθώς ήταν αδύνατο να συλλέξουμε χειροκίνητα τοσο μεγάλο αριθμό απο ιστολόγια, επεξεργαστήκαμε τα δεδομένα που έχει παραχωρήσει η εταιρεία Intelliseek στο ίδρυμα 'Δημόκριτος' για ερευνητικούς σκοπούς. Τα αρχεία αυτά περιέχουν εναν τεράστιο αριθμό απο ιστοσελίδες ιστολογίων, καθώς και την τοποθεσία τους στο διαδίκτυο. Έτσι, υλοποιήθηκε η λειτουργία "Extract URLs from XML" της εφαρμογής η οποία δέχεται ως είσοδο τα XML αρχεία της Intelliseek, εξάγει τα URLs τους και τα αποθηκεύει σε κάποιο αρχείο. Καθώς τα δεδομένα της Intelliseek χρονολογούνται πριν από ένα με δύο χρόνια [6], αρκετά URLs δείχνουν σε διαφορετικές ιστοσελίδες απο τις αρχικές, και πολλές δεν είναι πλέον ιστολόγια ή δεν υπάρχουν πια. Η εφαρμογή έχει τη δυνατότητα να απορρίπτει τις ιστοσελίδες που δεν υπάρχουν, αλλα δεν μπορεί να αναγνωρίσει εαν μια ιστοσελίδα είναι ιστολόγιο ή όχι. Για να αποφύγουμε αυτες τις περιπτώσεις ελέγξαμε με ένα script το μέγεθος των ιστοσελίδων όταν είχαν αποθηκευτεί

στο δίσκο μέσω της λειτουργίας caching της εφαρμογής: Εάν το μέγεθος αυτό ήταν μικρότερο απο 2 kilobytes τότε πιθανότατα η σελίδα ήταν κάποιο μύνημα λάθους, όπως 404 error (το οποίο σημαίνει οτι η αρχική σελίδα δεν βρέθηκε), και απορρίφθηκε.

Καθώς τα εργαλεία δημιουργίας ιστολόγιων συνήθως προσφέρουν και την εύκολη δημοσίευση τους στον δικτυακό τους τόπο, παρατηρώντας το domain των URLs διαπιστώσαμε οτι από τα 7857 τα 5600 είχαν δημιουργηθεί με εργαλεία όπως Wordpress (wordpress.com) , Blogger (blogspot.com) και Livejournal (livejournal.com). Καθώς το domain είναι απλά ενδεικτικό, η τιμή αυτή χρησιμεύει μόνο σαν κάτω όριο, και πιθανότατα τα ιστολόγια που δημιουργήθηκαν απο έτοιμα συστήματα είναι αρκετά περισσότερα. Επίσης, τα ιστολόγια που συλλέχθησαν είναι στην συντριπτική τους πλειοψηφία στην αγγλική γλώσσα, δίχως όμως να απουσιάζουν η κινεζική και η ισπανική γλώσσα.

6.2 Αποτελέσματα ανάλυσης και σχολιασμός

Η αξιολόγηση του συστήματος έγινε με βάση τα παρακάτω στοιχεία σχετικά με την ανάλυση του κάθε ιστολόγιου:

- Αρχικά εξετάζουμε εάν κάποιο ιστολόγιο κατατμήθηκε επιτυχώς η όχι. Η επιτυχία εδω αναφέρεται μόνο στο κατα πόσο μπόρεσε το πρόγραμμα να αναγνώσει την ιστοσελίδα και να την αναλύσει με κάποια από τις μεθόδους που αναπτύξαμε. Συνεπώς, αποτυχία θα έχουμε σε περίπτωση που δέν μπόρεσε να βρεθεί η σελίδα, ή αν καμία μέθοδος δεν επέστρεψε αποτελέσματα κάποιου είδους. Όμως, δεν μπορούμε να είμαστε απολύτως σίγουροι για την ορθότητα των αποτελεσμάτων, παρα μόνο εάν ελεχθούν από ανθρώπινο μάτι. Παρόλαυτά, οι μέθοδοι με τα rss feeds καθώς και με την αναγνώριση των tags είναι σχεδόν πάντα ακριβείς, δηλαδή όταν επιστρέφουν επιτυχή διαχωρισμό αυτός έχει γίνει σχεδόν πάντα σωστά.

- Εάν το ιστολόγιο αναλύθηκε επιτυχώς, κρατάμε στοιχεία για τη μέθοδο ανάλυσης που χρησιμοποιήθηκε.

- Υπολογίζουμε τα τελικά ποσοστά επιτυχίας και χρήσης των μεθόδων στο σύνολο των ιστολόγιων που αναλύθηκαν.

Απο τα 7857 ιστολόγια που αναλύθηκαν, τα 7069 διαχωρίστηκαν επιτυχώς, δηλαδή σε ποσοστό 90%. Απο αυτά:

- 4614 (ποσοστό 65,27%) ιστολόγια αναλύθηκαν με βάση τα rss feeds τους. Τα αποτελέσματα αναμένεται να είναι απολύτως σωστά και επαληθεύσιμα, καθώς τα rss feeds χρησιμοποιούνται για αυτόν καθεαυτόν τον σκοπό του διαχωρισμού των καταχωρήσεων.

- 1123 (ποσοστό 15,89%) ιστολόγια αναλύθηκαν με βάση τα αναγνωριστικά (tags) που περιέχονταν στις ιστοσελίδες. Ο αριθμός αυτός συμπεριλαμβάνει τα ιστολόγια στα οποία εντοπίστηκε το εργαλείο δημιουργίας τους (generator) και αυτά στα οποία δέν εντοπίστηκε και συνεπώς έγινε αναζήτηση αναγνωριστικών. Όπως και στην προηγούμενη μέθοδο, η συντριπτική πλειοψηφία των αποτελεσμάτων σε αυτήν την κατηγορία αναμένεται να είναι σωστά, λόγω της φύσεως της μεθόδου: τα tags χρησιμοποιούνται ακριβώς για τον σκοπό της αναγνώρισης.

- 1332 (ποσοστό 18,84%) ιστολόγια διαχωρίστηκαν με βάση τις ημερομηνίες που περιείχαν.

6.3 Διαχωρισμός με ημερομηνίες - Αξιολόγηση και επαλήθευση

Σε δεύτερο επίπεδο εξετάζουμε πιο λεπτομερώς την περίπτωση στην οποία αναμένουμε να έχουν γίνει λάθη, δηλαδή την περίπτωση διαχωρισμού με βάση τις ημερομηνίες.

Καθώς προαναφέρθηκε, η μέθοδος διαχωρισμού των ημερομηνιών ενδέχεται να οδηγήσει σε λανθασμένα αποτελέσματα, καθώς δεν υπάρχει σίγουρος τρόπος να εξασφαλίσουμε μηχανικά την ορθότητα του συνόλου των ημερομηνιών που επέλεξε. Έτσι, εκτός απο τα πειράματα τα οποία αφορούν την συνολική απόδοση της εφαρμογής, θεωρήσαμε σκόπιμο να επαληθεύσουμε με το χέρι τα αποτελέσματα αυτής της μεθόδου. Για τον σκοπό αυτόν, ξεχωρίσαμε ενα σύνολο απο ιστολόγια τα οποία το πρόγραμμα ανέλυσε τελικά με την μέθοδο που μας ενδιαφέρει: Τα ιστολόγια αυτά περιέχουν όλα ημερομηνίες και στην αρχική φάση ανάλυσής τους η εφαρμογή χρησιμοποίησε την μέθοδο των ημερομηνιών και μόνο για να διαχωρίσει τις καταχωρήσεις τους, καθώς οι άλλες μέθοδοι απέτυχαν. Κατόπιν, κρίναμε δια χειρός την αξιοπιστία των αποτελεσμάτων.

Οι περιπτώσεις στις οποίες η μέθοδος ενδέχεται να κάνει λάθος, δεδομένου του οτι το ιστολόγιο περιέχει ημερομηνίες, είναι οι εξής:

- Να μήν εντοπίσει ημερομηνίες, το οποίο πιθανότατα οφείλεται σε ανεπαρκείς κανονικές εκφράσεις, αλλά ενδέχεται να παρουσιαστεί και όταν ο html κώδικας της σελίδας είναι κακογραμμένος και ελλιπής.

- Να βρεί ημερομηνίες αλλά να μην επιλέξει αυτές που περικλείουν τις καταχωρήσεις. Στην περίπτωση αυτή υπάρχουν πολλαπλά σύνολα ημερομηνιών στο ιστολόγιο, και συνήθως ο λόγος που συμβαίνει η λάθος επιλογή είναι ότι κάποιο σύνολο περικλείει δεδομένα μεγαλύτερου μεγέθους απο ό,τι οι ζητούμενες καταχωρήσεις.

- Να επιλέξει το σωστό σύνολο ημερομηνιών αλλά να προκύψει σφάλμα κατα τη διάρκεια της εξαγωγής των ενδιάμεσων καταχωρήσεων, το οποίο πάλι συνήθως οφείλεται σε κακογραμμένο κώδικα της ιστοσελίδας.

Στην πρώτη περίπτωση, η μέθοδος θα επιστρέψει αδυναμία εξαγωγής. Θα γνωρίζουμε συνεπώς οτι η ιστοσελίδα δεν διαχωρίστηκε σωστά, και το αποτέλεσμα θα καταχωρηθεί σωστά ως εσφαλμένο. Ιστοσελίδες που ανήκουν σε αυτή την περίπτωση δεν συμπεριλαμβάνονται στο παρόν πείραμα, καθώς ζητάμε να επαληθεύσουμε την πραγματική ορθότητα της μεθόδου σε σωστά κατάυτήν αποτελέσματα.

Στην δεύτερη περίπτωση, η μέθοδος δεν μπορεί να αναγνωρίσει οτι το σύνολο των ημερομηνιών είναι εσφαλμένο και θα επιστρέψει θετικό αποτέλεσμα εξαγωγής με λανθασμένα όμως δεδομένα. Η περίπτωση αυτή είναι και η πιο επιζήμια καθώς αλλοιώνει και το συνολικό ποσοστό των τελικών αποτελεσμάτων και στατιστικών επιτυχίας.

Τέλος, εάν ευρεθεί το σωστό σύνολο αλλα αποτύχει η εξαγωγή των καταχωρήσεων, αυτό σημαίνει οτι η ιστοσελίδα περιέχει λανθασμένο κώδικα. Στην περίπτωση αυτή δεν είναι δυνατή η εξαγωγή με την προτεινόμενη μέθοδο.

Στην προσπάθειά μας να εξετάσουμε το ποσοστό των ψευδών αποτελεσμάτων, ελέγξαμε συνολικά 90 ιστοσελίδες, τις οποίες επιλέξαμε ως εξής: Αρχικά τροφοδοτήσαμε την εφαρμογή με ενα μεγάλο σύνολο τυχαίων ιστολόγιων απο τα δεδομένα της Intelliseek, και μετα το πέρας της ανάλυσής τους, επιλέξαμε τις ιστοσελίδες που διαχωρίστηκαν με βάση την μέθοδο των ημερομηνιών.

Κατόπιν χειροκίνητου ελέγχου, τα πραγματικά αποτελέσματα είναι τα εξής:

Λόγος αποτυχίας	Πλήθος ιστοσελίδων
Εύρεση εσφαλμένου συνόλου ημ/ών	3
Εύρεση σωστού συνόλου ημ/ών αλλα εσφαλμένος διαχωρισμός	1

Διαπιστώνουμε έτσι οτι ο αλγόριθμος δίνει ενα αρκετά καλό ποσοστό διαχωρισμού με ελάχιστα σφάλματα.

6.4 Αξιολόγηση της κατηγοριοποίησης των καταχωρήσεων

6.4.1 Συλλογή δεδομένων προς εκμάθηση και ταξινόμηση

Για την αξιολόγηση της κατηγοριοποίησης είναι απαραίτητο να εξετάσουμε τις καταχωρήσεις που αφορούν μια συγκεκριμένη θεματική περιοχή. Δεδομένου της ταχείας διάδοσής των ιστολογίων στο διαδίκτυο, το φάσμα των πιθανών θεμάτων που μπορεί να διαπραγματεύεται ένα ιστολόγιο είναι τεράστιο, συνεπώς είναι ανέφικτη μια προσπάθεια κατηγοριοποίησης κάποιου τυχαίου συνόλου.

Το θέμα που επιλέξαμε προς κατηγοριοποίηση είναι οι κριτικές ταινιών. Επειδή οι κριτικές ταινιών συνοδεύονται συνήθως και από μια ένδειξη βαθμολόγησης της ταινίας με αστέρια ή κάποια αριθμητική τιμή, η σωστή κατηγορία μπορεί να καθοριστεί με αυτοματοποιημένο τρόπο, γεγονός που καθιστά το πεδίο των κινηματογραφικών ταινιών ιδιαίτερα πρόσφορο για τη διεξαγωγή πειραμάτων.

Αρχικά αναζητήσαμε ιστολόγια τα οποία ασχολούνται με την κριτική ταινιών. Τα ιστολόγια αυτά αρχικά βρέθηκαν μέσα απο μηχανές αναζήτησης, και κατόπιν βρήκαμε και άλλα μέσω σχετικών συνδέσμων στις ιστοσελίδες τους, συλλέγοντας τελικά 70 ιστολόγια. Κατόπιν, διαχωρίσαμε τις καταχωρήσεις τους μέσω της εφαρμογής μας, απο οπου προέκυψαν 235 καταχωρήσεις με κριτικές ταινιών. Οι διαχωρισμένες αυτές καταχωρήσεις αποτελούν το σώμα κειμένων το οποίο χρησιμοποιείται για την αξιολόγηση του ταξινομητή μας.

Επίσης, το σώμα με τις κριτικές ταινιών των Pang et al. [PLV02] [1] προσφέρει ένα έτοιμο σύνολο με κριτικές ταινιών, τις οποίες και συμπεριλάβαμε

[1]Τα δεδομένα είναι διαθέσιμα στο http://www.cs.cornell.edu/people/pabo/movie-review-data/

στο σύνολό μας, μαζί με τις καταχωρήσεις που αναφέρονται προηγουμένως. Η βάση των Pang et al. αποτελείται από 1000 θετικές και 1000 αρνητικές κριτικές κινηματογραφικών ταινιών, που συλλέχθηκαν από τη βάση δεδομένων του newsgroup rec.arts.movies.reviews. Σε κάθε κατηγορία, οι κριτικές ταινιών με κοινό συγγραφέα είναι το πολύ 20 και από το κείμενο των εγγράφων έχουν αφαιρεθεί η βαθμολόγηση και τα html tags. Οι κριτικές που περιλαμβάνονται στη βάση είναι μεγάλες σε έκταση (αποτελούνται κατά μέσο όρο από 33 προτάσεις) και καλά δομημένες.

Επίσης απομονώθηκαν περίπου 200 κριτικές ταινιών απο την ιστοσελίδα http://www.rottentomatoes.com, η οποία είναι απο τις πιο γνωστές σελίδες που ασχολούνται με ταινίες και κινηματογραφικό υλικό. Στην περίπτωση αυτή οι κριτικές είναι μικρότερου μεγέθους και λιγότερο περιγραφικές, με αποτέλεσμα να περιέχουν περισσότερες λέξεις χαρακτηρισμού της εκάστοτε ταινίας ανα μέγεθος κειμένου σε σχέση με το προηγούμενο δείγμα.

Όλα τα παραπάνω δεδομένα συγκεντρώθηκαν σε ένα σύνολο το οποίο είμαστε σίγουροι οτι διαπραγματεύεται μόνο κριτικές ταινιών και περιέχει εναν μεγάλο αριθμό κριτικών με ποικιλομορφία στο μέγεθος και στο ύφος.

6.4.2 Δημιουργία των αρχείων κατηγοριοποίησης

Για την επεξεργασία απο τον ταξινομητή libsvm τα δεδομένα χρειάζεται να μορφοποιηθούν κατάλληλα, υπο μορφή διανυσμάτων χαρακτηριστικών. Το κάθε κείμενο κριτικής αποτελεί ένα στιγμιότυπο (instance) για τον ταξινομητή και αναπαρίσταται ως ενα διάνυσμα χαρακτηριστικών. Το πρώτο χαρακτηριστικό αντιπροσωπεύει την τιμή της κλάσης του εκάστοτε στιγμιότυπου, στην περίπτωσή μας κατα σύμβαση ¨1¨ αν το κείμενο αποτελεί θετική κριτική κάποιας ταινίας ή ¨0¨ εάν ειναι αρνητική. Καθένα απο τα χαρακτηριστικά αντιπροσωπεύει την εμφάνιση μιας λέξης απο το λεξικό το οποίο έχει δημιουργηθεί απο το σύνολο των κειμένων. Η τιμή του χαρακτηριστικού αυτού είναι η συχνότητα εμφάνισης της λέξης στο αντίστοιχο στιγμιότυπο (το οποίο είναι ενα κείμενο κριτικής).

Να σημειώσουμε εδώ οτι λέξεις μεγέθους ενός χαρακτήρα αγνοήθηκαν. Επίσης αφαιρέθηκαν τα html tags καθώς και ειδικοί χαρακτήρες που θα δυσχέραιναν την ταξινόμηση, όπως unicode σύμβολα.

Εναλλακτική μέθοδος αναπαράστασης χαρακτηριστικών - TF/IDF

Μία εναλλακτική αναπαράσταση των δεδομένων προς επεξεργασία που χρησιμοποιείται γενικότερα στην μηχανική εκμάθηση και την ανάκτηση πληροφοριών είναι οι τιμές TF/IDF [2]. Πολλές φορές η απεικόνιση με βάση τις τιμές αυτές προτιμάται έναντι των απλών συχνοτήτων εμφάνισης των λέξεων, καθώς οι τιμές TF/IDF παρέχουν μια εκτίμηση της σημαντικότητας ενός όρου (term) σε ένα κείμενο. Η τιμή tf απεικονίζει την σχετική συχνότητα μιας λέξης σε κάποιο στιγμιότυπο, και υπολογίζεται ως εξής:

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{6.1}$$

οπου ο αριθμητής είναι η συχνότητα εμφάνισης της λέξεως t_i στο κείμενο d_i και ο παρονομαστής το άθροισμα των συχνοτήτων εμφάνισης όλων των λέξεων στο κείμενο (ή αλλιώς, το μέγεθος του κειμένου σε λέξεις). Η τιμή idf ορίζεται ως ανάστροφη συχνότητα στιγμιότυπου και υπολογίζεται απο τον ακόλουθο τύπο:

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{6.2}$$

οπου ο αριθμητής είναι ο συνολικός αριθμός στιγμιοτύπων στο σύνολο μας και ο παρονομαστής ο αριθμός των στιγμιοτύπων που περιέχουν τη λέξη t_i. Η τελική τιμή που θα χαρακτηρίζει την κάθε λέξη t_i σε κάθε κείμενο d_i είναι η:

$$tfidf_{ij} = tf_{ij} * idf_i \tag{6.3}$$

και η οποία αντικαθιστά την απλή συχνότητα εμφάνισης απο το προηγούμενο βήμα.

Για να συγκρίνουμε τις δύο μεθόδους εκτελέσαμε ακριβώς τα ίδια πειράματα με τα ίδια σύνολα κειμένων.

6.4.3 Κανονικοποίηση των δεδομένων

Συνήθως σε προβλήματα μηχανικής μάθησης πριν την εκπαίδευση του ταξινομητή φροντίζουμε ώστε τα δεδομένα να έχουν κανονικοποιηθεί σε ένα συγκεκριμένο διάστημα. Το προτέρημα της κανονικοποίησης είναι οτι αποφεύγουμε την περίπτωση οπου στιγμιότυπα με μεγάλο εύρος τιμών επικρατούν δυσανάλογα αυτών με μικρότερο εύρος τιμών. Η κανονικοποίηση

[2]Term Frequency/Inverse Document Frequency

επίσης διευκολύνει την αποφυγή προβλημάτων σε αριθμητικές πράξεις με υπερβολικά μεγάλες τιμές. Ο Sarle στο Part 2 of Neural Networks FAQ [19] εξηγεί αναλυτικότερα τους λόγους που θα πρέπει να κανονικοποιούνται οι τιμές των χαρακτηριστικών κατα την χρήση των νευρωνικών δικτύων, οι περισσότεροι απο τους οποίους ισχύουν και στην περίπτωση των ταξινομητών svm. Κατά την διαδικασία της κανονικοποίησης οι τιμές των χαρακτηριστικών περιορίζονται σε ένα επιθυμητό διάστημα, συνήθως [-1,1] ή [0,1] όπως και προτιμήθηκε στην περίπτωσή μας. Η μέθοδος που χρησιμοποιήσαμε είναι η γραμμική κανονικοποίηση, στην οποία η κανονικοποιημένη τιμή ενός χαρακτηριστικού προκύπτει απο τον τύπο

$$x' = (x - m_i)/(M_i - m_i) \tag{6.4}$$

στην οποία τα m_i και M_i είναι αντίστοιχα η μέγιστη και ελάχιστη τιμή του χαρακτηριστικού i, το x είναι η αρχική τιμή του στο στιγμιότυπο και x' η τελική κανονικοποιημένη τιμή.

Εάν δεν επισημαίνεται το αντίθετο, όλα τα δεδομένα που αναφέρουμε στα πειράματά μας έχουν υποστεί κανονικοποίηση.

6.4.4 Διεξαγωγή πειραμάτων

Τα πειράματα διεξήχθησαν σε δύο σκέλη. Και στα δύο μετράμε την ακρίβεια ταξινόμησης του αλγορίθμου SVM, η οποία ορίζεται ως εξής:

$$accuracy = \frac{blogs\ \ classified\ \ correctly}{total\ \ blogs\ \ classified}$$

και η οποία είναι το μόνο μέτρο της ποσοτικής αξιολόγησης των αποτελεσμάτων που μας προσφέρει η libsvm.

Στο πρώτο μέρος των πειραμάτων εφαρμόσαμε σε ένα βήμα την εκμάθηση και την κατηγοριοποίηση χρησιμοποιώντας 10-πλή σταυρωτή επικύρωση στο σύνολο των δεδομένων που αναφέραμε στην παράγραφο 6.4.1. Στο δεύτερο μέρος, διαχωρίσαμε το σύνολο των δεδομένων σε δύο υποσύνολα, απο τα οποία το πρώτο χρησιμοποιήθηκε σαν σώμα εκπαίδευσης και το δεύτερο σαν σώμα προς κατηγοριοποίηση. Το σώμα που χρησιμοποιήθηκε για την εκπαίδευση του ταξινομητή αποτελείται απο τις κριτικές των Pang et all και απο τις κριτικές του rotten tomatoes, ενώ το σώμα προς κατηγοριοποίηση περιέχει μόνο τις καταχωρήσεις των ιστολόγιων. Η πρακτική χρησιμότητα αυτής της μεθόδου βασίζεται στο οτι με έναν ήδη εκπαιδευμένο ταξινομητή μπορούμε

να κατηγοριοποιούμε συνεχώς νέες καταχωρήσεις που διαχωρίζουμε από νέα ή ανανεωμένα ιστολόγια με την εξής διαδικασία:

- Αναζητούμε ιστολόγια με σχετικό θεματικό περιεχόμενο.
- Χρησιμοποιούμε την εφαρμογή μας ωστε να διαχωρίσουμε τις καταχωρήσεις τους.
- Τοποθετούμε τις καταχωρήσεις σε ένα σώμα προς κατηγοριοποίηση.
- Κατηγοριοποιούμε τις καταχωρήσεις με τον ήδη εκπαιδευμένο ταξινομητή.

Και στις δύο περιπτώσεις τα πειράματα έγιναν με απλή απεικόνιση των συχνοτήτων των στιγμιοτύπων, καθώς και με τιμές TF/IDF. Καθώς οι παράμετροι C, γ του αλγορίθμου διανυσμάτων υποστήριξης εχουν μεγάλη επιρροή στο τελικό αποτέλεσμα, διεξάγαμε πειράματα με τις προεπιλεγμένες τιμές τους καθώς και με τιμές που προέκυψαν απο αναζήτηση των βέλτιστων τιμών, χρησιμοποιώντας το εργαλείο της libsvm που εκτελεί επαναλαμβανόμενους κύκλους κατηγοριοποίησης και αξιολόγησης των αποτελεσμάτων.

Κατηγοριοποίηση όλων των κειμένων με σταυρωτή επικύρωση

Για την διεξαγωγή των πειραμάτων χρησιμοποιούμε συνολικά 2.435 κείμενα τα οποία περιέχουν κριτικές ταινιών των Pang et all, της ιστοσελίδας rotten tomatoes καθώς και τις καταχωρήσεις των ιστολόγιων που διαχωρίσαμε. Στο πρώτο πείραμα τρέξαμε τον αλγόριθμο με τις ερήμην παραμέτρους $C = 1, \gamma = 1/2435$ της βιβλιοθήκης libsvm και τον προεπιλεγμένο πυρήνα (πυρήνας ακτινωτής βάσης). Όσον αφορά το μέγεθος του σώματος εκπαίδευσης (παράμετρος K) στην σταυρωτή επικύρωση, πραγματοποιήθηκαν μερικά πειράματα στο διάστημα [3,20] και διαπιστώσαμε οτι η αλλαγή του δεν επιφέρει μετρήσιμες αλλαγές στα αποτελέσματα. Έτσι, επιλέξαμε K=10 δηλαδή δεκαπλή σταυρωτή επικύρωση. Η ακρίβεια του αλγορίθμου σε αυτήν την περίπτωση ήταν 50,5819%. Τα αποτελέσματα αυτα θα χρησιμοποιηθούν σαν το κάτω όριο (Baseline Results) ωστε να έχουμε κάποιο ενδεικτικό μέτρο σύγκρισης για τα επόμενα πειράματα. Κατόπιν, έγινε η αναζήτηση των βέλτιστων τιμών των παραμέτρων C και γ. Για κάθε συνάρτηση πυρήνα εκτελέσαμε διαδοχικές φάσεις του αλγορίθμου εκμάθησης με εκθετικά αυξανόμενες τιμές $\boxed{2^{16}, 2^{15}...2^{3}}$ των παραμέτρων , και τελικά καταλήξαμε στις τιμές με την βέλτιστη απόδοση, όπως φαίνεται στον πίνακα 6.1.

Πυρήνας	Ακρίβεια κατηγοριοποίησης	C	γ
Γραμμικός	73,5%	0,5	0,078125
Πολυωνυμικός	60,68%	0,5	0,5
Ακτινωτής βάσης	73,5%	2048	0,000122
Σιγμοειδής	73,5%	2048	0,000122

Σχήμα 6.1: Αποτελέσματα με τις Βέλτιστες τιμές παραμέτρων για κάθε συνάρτηση πυρήνα.

Πυρήνας	Ακρίβεια κατηγοριοποίησης	C	γ
Γραμμικός	83,48%	0,5	0,078125
Πολυωνυμικός	53,04%	0,5	0,0078125
Ακτινωτής βάσης	84,84%	32	0,000122
Σιγμοειδής	83,75%	2048	0,000122

Σχήμα 6.2: Αποτελέσματα με τις βέλτιστες τιμές παραμέτρων για κάθε συνάρτηση πυρήνα με τιμές TF/IDF.

Όπως παρατηρούμε, η μέγιστη ακρίβεια της κατηγοριοποίησης είναι 73,5%. Ο πολυωνυμικός πυρήνας δεν κατάφερε να αποδώσει όσο οι υπόλοιποι, οι οποίοι είχαν ακριβώς το ίδιο αποτέλεσμα. Σε μεγαλύτερο μέγεθος σώματος ενδέχεται να είχαμε μεγαλύτερες διαφοροποιήσεις.

Σε αυτο το σημείο εκτελέσαμε τα παραπάνω πειράματα με δεδομένα που προέκυψαν μέσω της απεικόνισης με TF/IDF τιμές. Τα αποτελέσματα με τις βέλτιστες παράμετρους για διαφορετικούς πυρήνες SVM φαίνονται στον πίνακα 6.2.

Τελικά παρατηρούμε οτι σε όλα τα πειράματα η αναπαράσταση των λέξεων με τιμές TF/IDF υπερτερεί σημαντικά έναντι της αναπαράστασης συχνοτήτων. Η μέγιστη διαφορά που επιτύχαμε είναι 11,34% (πυρήνας ακτινωτής βάσης), που σημαίνει οτι ταξινομήθηκαν σωστά 276 επιπλέον κείμενα. Η μεγαλύτερη επίδοση ήταν αναμενόμενη, καθώς οι τιμές TF/IDF εμπεριέχουν περισσότερη πληροφορία συσχετισμού των λέξεων με τα κείμενα από ο,τι οι απλές τιμές συχνοτήτων.

Πυρήνας	Ακρίβεια κατηγοριοποίησης	C	γ
Γραμμικός	44,44%	0,5	0,0078125
Πολυωνυμικός	44,44%	32	0,0078125
Ακτινωτής βάσης	58,98%	32	0,000122
Σιγμοειδής	55,55%	2048	0,000122

Σχήμα 6.3: Βέλτιστες τιμές παραμέτρων για κάθε συνάρτηση πυρήνα στην κατηγοριοποίηση των καταχωρήσεων.

Εκπαίδευση του ταξινομητή και κατηγοριοποίηση καταχωρήσεων ιστολόγιων

Για την διεξαγωγή των πειραμάτων σε αυτό το σκέλος αρχικά εκπαιδεύσαμε τον ταξινομητή στο σώμα που περιέχει τις κριτικές των Pang et all καθώς και της ιστοσελίδας Rotten Tomatoes, οι οποίες στο σύνολό τους ήταν 2.200. Κατόπιν, αξιολογήσαμε την ταξινόμηση του σώματος των καταχωρήσεων, το οποίο αποτελείται απο 235 κείμενα, με χρήση του ήδη εκπαιδευμένου ταξινομητή.

Στο πρώτο πείραμα τρέξαμε τον αλγόριθμο με τις ερήμην παραμέτρους $C = 1, \gamma = 1/2200$ της βιβλιοθήκης libsvm και τον προεπιλεγμένο πυρήνα (πυρήνας ακτινωτής βάσης). Η ακρίβεια του ταξινομητή σε αυτήν την περίπτωση ήταν 55,55%. Τα αποτελέσματα αυτα θα χρησιμοποιηθούν σαν το κάτω όριο (Baseline Results) ωστε να έχουμε κάποιο ενδεικτικό μέτρο σύγκρισης για τα επόμενα πειράματα. Κατόπιν, έγινε η αναζήτηση των βέλτιστων τιμών των παραμέτρων C και γ. Για κάθε συνάρτηση πυρήνα εκτελέσαμε διαδοχικές φάσεις του αλγορίθμου εκμάθησης με εκθετικά αυξανόμενες τιμές των παραμέτρων όπως στα προηγούμενα πειράματα, καταλήγοντας στα αποτελέσματα του πίνακα 6.3.

Η μέγιστη ακρίβεια κατηγοριοποίησης που επιτυγχάνουμε είναι 58,98%. Το ποσοστό αυτό είναι αρκετά ικανοποιητικό, δεδομένου οτι το σώμα εκπαίδευσης είναι τελείως ανεξάρτητο απο το σώμα κατηγοριοποίησης.

Κατόπιν εκτελέσαμε τα παραπάνω πειράματα με δεδομένα που προέκυψαν μέσω της απεικόνισης με TF/IDF τιμές. Τα αποτελέσματα με τις βέλτιστες παραμέτρους για διαφορετικούς πυρήνες SVM φαίνονται στον πίνακα 6.4.

Πυρήνας	Ακρίβεια κατηγοριοποίησης	C	γ
Γραμμικός	55,55%	0,5	0,078125
Πολυωνυμικός	55,55%	0,5	0,5
Ακτινωτής βάσης	58,54%	32	0.078125
Σιγμοειδής	57,7%	32	0.078125

Σχήμα 6.4: Βέλτιστες τιμές παραμέτρων για κάθε συνάρτηση πυρήνα στην κατηγοριοποίηση των καταχωρήσεων με τιμές TF/IDF.

Παρατηρούμε οτι ο πυρήνας ακτινωτής βάσης με τις βέλτιστες παραμέτρους έδωσε την καλύτερη ακρίβεια ταξινόμησης. Η διαφορά στο ποσοστό μεταξύ των διαφορετικών συναρτήσεων πυρήνα είναι πολύ μικρή, όπως μικρή είναι και η διαφορά μεταξύ των αναπαραστάσεων με τιμές συχνοτήτων και τιμές TF/IDF.

6.4.5 Σχολιασμός των αποτελεσμάτων

Η ταξινόμηση με σταυρωτή επικύρωση έδωσε ικανοποιητικά αποτελέσματα αφού κατάφερε να κατηγοριοποιήσει σωστά 2066 κριτικές απο τις συνολικές 2435. Όσον αφορά την δεύτερη σειρά πειραμάτων με ξεχωριστά σώματα εκπαίδευσης και κατηγοριοποίησης παρατηρήσαμε έντονη πτώση της ακρίβειας, καθώς ταξινομήθηκαν σωστά 139 απο τις 235 καταχωρήσεις.Οι διαφορές των αποτελεσμάτων οφείλονται στο οτι το σώμα εκπαίδευσης και το σώμα κατηγοριοποίησης είναι τελείως ανεξάρτητα : Στη σταυρωτή επικύρωση το αποτέλεσμα προκύπτει από τον μέσο όρο της ακρίβειας ταξινόμησης σε κάθε βήμα επικύρωσης. Κάθε στιγμιότυπο χρησιμοποιείται για να εκπαιδεύσει ακριβώς μια φορά τον ταξινομητή, και επίσης κατηγοριοποιείται ακριβώς μια φορά. Αντιθέτως, στα πειράματα της τελευταία ενότητας η ακρίβεια προκύπτει μόνο από το σώμα των καταχωρήσεων, καθώς κανένα στιγμιότυπο του σώματος εκπαίδευσης δεν ταξινομείται. Κατέπέκταση, υπάρχει πολύ λιγότερη πληροφορία που μπορεί να χρησιμοποιήσει ο ταξινομητής στην κατηγοριοποίηση, και έτσι δικαιολογείται η διαφορά με τα προηγούμενα αποτελέσματα.

Κεφάλαιο 7

Επίλογος

7.1 Συμπεράσματα

Στην εργασία αυτή μελετήθηκαν δύο θέματα.

Στο πρώτο σκέλος, υλοποιήθηκαν μέθοδοι με τις οποίες μπορούμε να επεξεργαστούμε τα ιστολόγια και να τα χωρίσουμε στις καταχωρίσεις που περιέχουν. Οι τρείς αυτές μέθοδοι δουλεύουν συμπληρωματικά, καθώς ένα ιστολόγιο θα αναλυθεί κατα σειρά με καθεμία απο αυτές εως ότου διαχωριστεί επιτυχώς. Απο τις μεθόδους αυτές πρώτη εφαρμόζεται η μέθοδος ανάλυσης των feeds, η οποία δείχνει οτι έχει το μεγαλύτερο ποσοστό επιτυχίας που αγγίζει το 100% στις ιστοσελίδες στις οποίες εφαρμόζεται, και για αυτο τον λόγο εκτελείται πρώτη. Εάν αυτή αποτύχει, η ιστοσελίδα αποθηκεύεται στον δίσκο για γρηγορότερη προσπέλαση, και μετά χρησιμοποιούμε την μέθοδο ανάλυσης των tags. Εάν πάλι δεν επιτευχθεί ο διαχωρισμός του ιστολόγιου, εφαρμόζεται η τρίτη μέθοδος ανάλυσης με βάση τις ημερομηνίες των καταχωρήσεων. Η συνολική διαδικασία επιτυγχάνει μια αρκετά ικανοποιητική ακρίβεια.

Ο αριθμός των ιστολόγιων έχει παρατηρηθεί οτι αυξάνει εκθετικά μέρα με τη μέρα, και έτσι οι δυνατότητες της εφαρμογής είναι δυνατόν να αξιοποιηθούν σε πολλές εφαρμογές όπως για παράδειγμα η ανάλυση της γνώμης μιας μεγάλης μερίδας ανθρώπων για κάποιο προϊόν [20] [21] [22], η εξαγωγή της σημασίας κάποιου γεγονότος, ακόμα και για στατιστικές αναλύσεις [23].

Επιπλέον, τα ποσοστά σωστού διαχωρισμού καταχωρήσεων αναμένουμε να βελτιωθούν, καθώς όλο και μεγαλύτερο ποσοστό ιστοσελίδων δημιουργείται με έτοιμα εργαλεία, γεγονός το οποίο εξασφαλίζει μια ομοιομορφία μεταξύ τους και τελικά μεγαλύτερη ευκολία στον διαχωρισμό τους. Τα εργαλεία αυτά

δίνουν ολοένα και περισσότερες δυνατότητες εξατομίκευσης των ιστοσελίδων απο τους χρήστες τους, γεγονός όμως το οποίο δεν μας επηρρεάζει, καθώς ο διαχωρισμός γίνεται με βάση τις μετα-πληροφορίες του HTML κώδικα των ιστοσελίδων που είναι ανεξάρτητες απο την τελική εμφάνιση στον χρήστη. Η αυξανόμενη χρήση των rss feeds θα βοηθήσει σημαντικά καθώς η αντίστοιχη μέθοδος είναι εξαιρετικά ακριβής. Ακόμη, η απόδοση του διαχωρισμού της εφαρμογής μπορεί να βελτιωθεί με επιπλέον ευριστικές μεθόδους για την εύρεση άλλων αναγνωριστικών διαχωρισμού, καθώς και με τη χρήση μηχανικής εκμάθησης. Για παράδειγμα , ο Συγλέτος και άλλοι [24] παρουσίασαν μια μέθοδος αυτόματης προσαρμογής των κανονικών εκφράσεων η οποία θα μπορούσε να εφαρμοστεί για πιο αξιόπιστη αναγνώριση των ημερομηνιών στην αντίστοιχη μέθοδο.

Κατόπιν, αξιοποιώντας τα δεδομένα που προέκυψαν απο την ανάλυση των ιστολόγιων επιδείξαμε την χρησιμότητα της διαδικασίας σε μια συγκεκριμένη εφαρμογή η οποία είναι η ταξινόμηση κριτικών για ταινίες, με βάση την άποψη που εκφράζουν. Για το σκοπό αυτό χρησιμοποιήσαμε τον ταξινομητή svm, ο οποίος οδήγησε σε ικανοποιητικά αποτελέσματα όταν χρησιμοποποιήσαμε σταυρωτή επικύρωση. Όταν χρησιμοποιήσαμε ξεχωριστό σώμα εκπαίδευσης και ταξινόμησης τα αποτελέσματα ήταν αρκετά πιο χαμηλά. Αυτό δεν σημαίνει όμως οτι η μέθοδος χάνει την πρακτική χρησιμότητά της, καθώς όσο αυξάνει το μέγεθος του σώματος εκπαίδευσης θα βελτιώνεται και η ακρίβεια της ταξινόμησης. Η μέθοδος αυτή είναι ενδιαφέρουσα καθώς ένας καλύτερα εκπαιδευμένος ταξινομητής θα μπορούσε να χρησιμοποιηθεί για την συνεχή ταξινόμηση νέων καταχωρήσεων. Σε συνδυασμό με την εφαρμογή κατάτμησης των ιστολόγιων που υλοποιήσαμε προκύπτει ένα βασικό σύστημα αυτόματης κατάτμησης και κατηγοριοποίησης των νέων ή ανανεωμένων ιστολόγιων.

Η μηχανική μάθηση και ταξινόμηση είναι ένας τεράστιος κλάδος συνεχής έρευνας, και είναι πιθανόν να μπορούμε να βελτιώσουμε τα αποτελέσματα εαν χρησιμοποιήσουμε με καλύτερο τρόπο την σημασιολογική και συντακτική πληροφορία του κειμένου.

Το βασικό συμπέρασμα αυτής της εργασίας είναι οτι ο πλούτος της πληροφορίας που υπάρχει σε εκατομμύρια ιστοσελίδες είναι δυνατόν να αξιοποιηθεί και να μας δώσει πολύ ενδιαφέρουσες πληροφορίες, και η εργασία προσφέρει μια βάση για περαιτέρω έρευνα με στόχο την ανάπτυξη μιας εύρωστης μεθόδου ανάλυσης του αστρονομικού μεγέθους των δεδομένων που μπορεί να προκύψουν απο τα ιστολόγια.

Bibliography

[1] R. Kumar, J. Novak, *et al.*, "On the bursty evolution of blogspace," *Proc. of the 12th International World Wide Web Conference, pages 568•576*, 2003.

[2] N. Glance, M. Hurst, *et al.*, "Blogpulse: Automated trend discovery for weblogs," *Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[3] T. Fukuhara, T. Murayama, and T. Nishida, "Analyzing concerns of people using weblog articles and real world temporal data," *Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2005.

[4] P. Kolari, T. Finin, and A. Joshi, "Svms for the blogosphere: Blog identification and splog detection," *Proceedings of the 2006 AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs, pages 92•99*, 2006.

[5] M. Ceglowski, "Www::blog::identify - identify blogging tools based on url and content," 2003.

[6] Intelliseek, "3rd annual workshop on the weblogging ecosystem," *Edinburgh, Scotland*, 2006.

[7] L. Eikvil, "Information extraction from world wide web - a survey," *Technical Report 945, Norweigan Computing Center*, 1999.

[8] J. Heflin, "Towards the semantic web: Knowledge representation in a dynamic, distributed environment," *Ph.D. Thesis, University of Maryland, College Park*, 2001.

[9] W. D. T. Committee, *Document object model technical reports*, 2003. http://www.w3.org/DOM/DOMTR.

[10] S. Debnath *et al.*, "Automatic identification of informative sections of web pages," vol. 98, no. E2, pp. 3247–3259, 2005.

[11] F. Douglis and T. Ball, "Tracking and viewing changes on the web," *USENIX Annual Technical Conference*, 1996.

[12] T. M. Breuel, "Information extraction from html documents by structural matching," *PARC, Inc., Palo Alto, CA, USA*, 2001.

[13] Y. S. Tomoyuki Nanno *et al.*, "Automatic collection and monitoring of japanese weblogs," *autoblog*, 1991.

[14] O. Manabu, N. Tomoyuki, *et al.*, "Automatically collecting, monitoring and mining japanese weblogs," *International World Wide Web Conference*, 2004.

[15] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers.," *Proceedings of the Fifth Annual Workshop on Computa-tional Learning Theory*, 1992.

[16] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines. software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.," 2001.

[17] M. Salton, "Introduction to modern information retrieval," 1983.

[18] E. Kaldeli, "Training text classifiers for sentiment classification," *thesis, National Technical University of Athens*, 2005.

[19] W. S. Sarle, *Neural Network FAQ Part 2*, 2002. Available at ftp://ftp.sas.com/pub/neural/FAQ2.html.

[20] T. Wilson and J. Wiebe, "Annotating opinions in the world press," 2003.

[21] J. Wiebe, E. Breck, C. Buckley, and C. Cardie, "Recognizing and organizing opinions expressed in the world press.," 2003.

[22] W. Zhang, C. Yu, and W. Meng, "Opinion retrieval from blogs," in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, (New York, NY, USA), pp. 831–840, ACM, 2007.

[23] A. Dalli, "System for spatio-temporal analysis of online news and blogs," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*, (New York, NY, USA), pp. 929–930, ACM, 2006.

[24] G.Sigletos *et al.*, "Meta-learning beyond classification: A framework for information extraction from the web," *Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik (Croatia)*, 2003.