Ατομική Διπλωματική Εργασία

ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΤΑΧΥΔΡΟΜΕΙΟΥ

Ειρήνη Ελευθερίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δεκέμβριος 2009

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΤΑΧΥΔΡΟΜΕΙΟΥ

Ειρήνη Ελευθερίου

Επιβλέπων Καθηγητής
Μάριος Δ. Δικαιάκος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Δεκέμβριος 2009

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Μάριο Δικαιάκο για την ευκαιρία που μου έδωσε να ασχοληθώ με την παρούσα διπλωματική εργασία και την πολύτιμη καθοδήγησή του. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την πολύτιμη υποστήριξη που μου πρόσφεραν καθ' όλη την διάρκεια των σπουδών μου στο Πανεπιστήμιο Κύπρου. Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον φίλο και συμφοιτητή μου Ορέστη Σπανό για την πολύτιμη βοήθεια που μου πρόσφερε κατά την ολοκλήρωση αυτής της διπλωματικής εργασίας.

Περίληψη

Στις μέρες μας, το ηλεκτρονικό ταχυδρομείο αποκτά όλο και περισσότερους χρήστες μέρα με τη μέρα. Τα ηλεκτρονικά μηνύματα είναι ένας από του πιο οικονομικούς και πιο γρήγορους τρόπους επικοινωνίας. Για αυτό τον λόγο τα συστήματα ηλεκτρονικού ταχυδρομείου διευκολύνουν την αποθήκευση των ηλεκτρονικών μηνυμάτων σε περισσότερα του ενός αρχεία αποθήκευσης, δηλαδή mailboxes, και την οργάνωση αυτών των αποθηκών σε καταλόγους. Σαν αποτέλεσμα, ο χρήστης αναγκάζεται να ταξινομεί αυτά τα μηνύματα στους διάφορους φακέλους από μόνος του. Στόχος αυτής της διπλωματικής εργασίας είναι η δημιουργία ενός υποσυστήματος διαχείρισης ηλεκτρονικού ταχυδρομείου το οποίο χρησιμοποιεί τεχνικές Ανάκτησης Πληροφοριών για να ταξινομεί τα εισερχόμενα ή και τα υπάρχον ηλεκτρονικά μηνύματα με αυτόματο τρόπο στα κατάλληλα mailboxes που διαθέτει ο χρήστης. Ο χρήστης έχει την δυνατότητα να καθορίσει ο ίδιος πόσο βάρος επιθυμεί να δοθεί σε κάθε πεδίο του μηνύματος ανάμεσα στα πεδία From, Subject και Body. Επίσης, ο χρήστης μπορεί να επιλέξει αν θέλει να γίνει χρήση του Stemmer κατά την ταξινόμηση ή όχι. Το υποσύστημα αυτό δημιουργήθηκε σαν πρόσθετο(extension/add-on) του Mozilla Thunderbird και υλοποιεί τον αλγόριθμο Bayes για την ταξινόμηση των ηλεκτρονικών μηνυμάτων. Για την αποτίμηση του συστήματος χρησιμοποιήθηκε ένα υποσύνολο μηνυμάτων από μια μεγάλη συλλογή πραγματικών ηλεκτρονικών μηνυμάτων διαφόρων υπαλλήλων της εταιρείας Enron. Από τα πειράματα που πραγματοποιήθηκαν αποδεικνύεται ότι ο αλγόριθμος του Bayes σε κάποιες περιπτώσεις δουλεύει αρκετά καλά, ανάλογα με την τεχνική ταξινόμησης που χρησιμοποιεί κάθε χρήστης. Επίσης, παρατηρήθηκε ότι η χρήση βαρών και Stemmer δεν είναι πάντοτε αποδοτική, αλλά εξαρτάται άμεσα από την τεχνική ταξινόμησης κάθε χρήστη.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Το Ηλεκτρονικό Ταχυδρομείο

Όσο αυξάνονται οι χρήστες του διαδικτύου, τα ηλεκτρονικά μηνύματα αποτελούν ένα από τους πιο δημοφιλή και αποδοτικούς τρόπους επικοινωνίας. Χρησιμοποιείται από εταιρείες, οργανισμούς, από ανθρώπους που θέλουν να προωθήσουν κάποια προϊόντα ή ακόμα και από ανθρώπους που θέλουν να μοιράσουν πληροφορίες οι οποίες εξυπηρετούν κάποιο σκοπό. Ένας λόγος που η χρήση του ηλεκτρονικού ταχυδρομείου αυξάνεται συνεχώς, είναι ότι δεν χρειάζονται ειδικές γνώσεις για να μπορεί κάποιος να χρησιμοποιήσει το ηλεκτρονικό ταχυδρομείο. Ένας δεύτερος λόγος είναι ότι στις πλύστες περιπτώσεις η κατοχή και χρήση του ηλεκτρονικού ταχυδρομείου είναι δωρεάν. Τέλος, είναι ένας πολύ γρήγορος τρόπος επικοινωνίας. Για αυτούς τους λόγους, το διακινούμενο ηλεκτρονικό ταχυδρομείο διογκώνεται συνεχώς.

Πολλοί πελάτες ηλεκτρονικού ταχυδρομείου (email clients) έχουν αναπτυχθεί για να διευκολύνουν την διαχείριση του ηλεκτρονικού ταχυδρομείου, προσφέροντας την δυνατότητα στους χρήστες τους να μπορούν να αποθηκεύουν τα ηλεκτρονικά τους μηνύματα σε περισσότερα του ενός αρχεία αποθήκευσης και την οργάνωση τους σε καταλόγους και υποκαταλόγους. Έτσι, ο χρήστης μπορεί να ταξινομεί τα ηλεκτρονικά του μηνύματα σε πολλούς καταλόγους, καθιστώντας την μελλοντική ανεύρεση κάποιου μηνύματος πιο εύκολη. Μερικοί email clients είναι: Mozilla Thunderbird, Microsoft Office Outlook, Outlook Express, Eudora και Windows Live Mail.

Το πιο πάνω προϋποθέτει χειρονακτική ταξινόμηση των μηνυμάτων από τον χρήστη. Για διευκόλυνση αυτής της διαδικασίας πολλά συστήματα έχουν αναπτυχθεί, όμως τα περισσότερα εστιάζουν αποκλειστικά στον εντοπισμό spam μηνυμάτων και στη μεταφορά τους σε ειδικό αρχείο (spam folder).

Σκοπός αυτής της διπλωματικής εργασίας είναι η δημιουργία ενός συστήματος ταξινόμησης ηλεκτρονικών μηνυμάτων στους καταλόγους που διαθέτει ο χρήστης, ανάλογα με το περιεχόμενο τους και την στρατηγική ταξινόμησης που χρησιμοποιεί ο χρήστης. Για την πραγματοποίηση του σκοπού αυτού θα χρησιμοποιηθεί ένας προϋπάρχον αλγόριθμος ταξινόμησης, ο αλγόριθμος Bayes, ο οποίος συνήθως χρησιμοποιείται για ανίχνευση spam μηνυμάτων ή και για ταξινόμηση κειμένου σε κατηγορίες.

Αν δούμε το πιο πάνω πρόβλημα σαν ένα πρόβλημα ταξινόμησης κειμένου, κάθε ηλεκτρονικό μήνυμα θεωρείται σαν ένα έγγραφο και κάθε πιθανός κατάλογος θεωρείται σαν μια κατηγορία στην οποία πιθανόν να ανήκει. Έτσι μπορούμε να αξιοποιήσουμε τεχνικές από τον κλάδο Ανάκτησης Πληροφοριών(Information Retrieval) για να επιτύχουμε το σκοπό μας.

Το σύστημα θα αναπτυχθεί σαν πρόσθετο του Mozilla Thunderbird, ενός πελάτη ηλεκτρονικού ταχυδρομείου ανοικτού κώδικα.

1.2 Σχετική Εργασία

Για τον Thunderbird αναπτύχθηκαν τρία πρόσθετα τα οποία χρησιμοποιούν τον αλγόριθμο του Bayes για το φιλτράρισμα των ηλεκτρονικών μηνυμάτων. Το πρώτο είναι το MailClassifier το οποίο ταξινομεί τα εισερχόμενα μηνύματα χρησιμοποιώντας τον αλγόριθμο του Bayes. Το δεύτερο είναι το thunderbayes το οποίο ανιχνεύει τα μηνύματα που είναι spam κάνοντας χρήση του αλγορίθμου του Bayes για ανίχνευση spam μηνυμάτων(spambayes). Τέλος, το spamato4thunderbird είναι ένα φίλτρο για ανίχνευση spam μηνυμάτων το οποίο συνδυάζει πολλές τεχνικές για ανίχνευση spam. Είναι επίσης διαθέσιμο και για τον Outlook.

Κεφάλαιο 2

Τεχνολογίες

2.1 Ανασκόπηση

Τα τελευταία χρόνια, η κατηγοριοποίηση κειμένου είναι μια πολύ δημοφιλής εφαρμογή μηχανικής μάθησης. Κατηγοριοποίηση δεν σημαίνει μόνο κατηγοριοποίηση εγγράφων βάση κάποιου θέματος [11],αλλά μπορεί να είναι κατηγοριοποίηση κατά είδος [5], κατά το επάγγελμα του συγγραφέα [3] ή ακόμη κατά το φύλο του συγγραφέα.

Στο πεδίο των ηλεκτρονικών μηνυμάτων, οι περισσότερες εφαρμογές εστιάζουν στην αναγνώριση spam μηνυμάτων [4], όπως επίσης και αναγνώριση threads τα οποία είναι συζητήσεις μεταξύ δυο ή και περισσότερων ατόμων [12], και αυτόματη δημιουργία νέων καταλόγων [7].

Όσο για την κατηγοριοποίηση μηνυμάτων σε καταλόγους, οι Payne και Edwards (1997) χρησιμοποίησαν τους αλγορίθμους CN2 [1] και k-NN(k nearest neighbor) για την κατηγοριοποίηση των προσωπικών τους μηνυμάτων και συμφώνησαν ότι ο πρώτος είναι καλύτερος από τον δεύτερο [13]. Ο Provost δείχνει ότι ο αλγόριθμος του Bayes για κατηγοριοποίηση ηλεκτρονικών μηνυμάτων υπερτερεί του RIPPER [15]. Λίγο αργότερα ο Rennie χρησιμοποιώντας και πάλι τον αλγόριθμο Bayes υλοποίησε ένα σύστημα για κατηγοριοποίηση των ηλεκτρονικών μηνυμάτων, το οποίο προτείνει τους

τρεις πιο κατάλληλους καταλόγους για κάθε εισερχόμενο ηλεκτρονικό μήνυμα, επιτυχαίνοντας πολύ ψηλή ακρίβεια [16]. Στη συνέχεια χρησιμοποιήθηκε για πρώτη φορά ο αλγόριθμος SVM(Support Vector Machine) για κατηγοριοποίηση ηλεκτρονικών μηνυμάτων και αποδείχτηκε ότι έχει πολλά πλεονεκτήματα έναντι του αλγορίθμου Bayes [8].

Η κατηγοριοποίηση του ηλεκτρονικού ταχυδρομείου έχει κάποιες δυσκολίες που την κάνουν να διαφέρει από την κατηγοριοποίηση κειμένου. Οι χρήστες ηλεκτρονικού ταχυδρομείου δημιουργούν νέους καταλόγους και αφήνουν κάποιους άλλους αχρησιμοποίητους. Επίσης, οι κατάλογοι ηλεκτρονικών μηνυμάτων δεν αντιστοιχούν πάντα σε μόνο ένα θέμα, αλλά πολλές φορές περιέχουν μηνύματα που σχετίζονται με περισσότερα από ένα θέματα. Για παράδειγμα ένας κατάλογος με το όνομα «Προσωπικά» δυνατόν να περιέχει ηλεκτρονικά μηνύματα που σχετίζονται με διάφορα θέματα. Όσο μεγαλώνουν τα περιεχόμενα ενός καταλόγου, υπάρχει περίπτωση το θέμα που σχετίζεται με αυτόν να μεταλλαχτεί. Ακόμη, κάθε χρήστης έχει διαφορετικές τακτικές για την ταξινόμηση των μηνυμάτων του. Έτσι, μπορεί μια αυτόματη μέθοδος κατηγοριοποίησης να δουλεύει καλά για κάποιον χρήστη ενώ για κάποιον άλλο όχι.

Το πρόβλημα της ταξινόμησης ηλεκτρονικών μηνυμάτων δεν τράβηξε πολύ το ενδιαφέρων για έρευνα. Ένας πιθανός λόγος είναι ότι δεν υπήρχε δημοσίευση κάποιου αναγνωρισμένου συνόλου από πραγματικά ηλεκτρονικά μηνύματα στο οποίο να μπορεί να εφαρμοστούν διάφορες μέθοδοι ταξινόμησης και να αποτιμηθούν. Εν τούτοις, ένα τεράστιο σύνολο από πραγματικά ηλεκτρονικά μηνύματα υπαλλήλων της εταιρείας Enron δόθηκε στην δημοσιότητα μετά από το μεγάλο οικονομικό σκάνδαλο που ξέσπασε την εταιρεία. Οι Klimt και Yang στο [9] παρουσίασαν κάποια στατιστικά σχετικά με το σύνολο αυτό.

2.2 Πελάτης Ηλεκτρονικού Ταχυδρομείου

Ένας email client (πελάτης ηλεκτρονικού ταχυδρομείου) ή MUA (Mail User Agent), είναι ένα πρόγραμμα το οποίο χρησιμοποιείται για τη διαχείριση του ηλεκτρονικού ταχυδρομείου. Τα μηνύματα βρίσκονται αποθηκευμένα σε κάποιον email server (MTA

= Mail Transfer Agent), ο οποίος είναι μια εφαρμογή η οποία λαμβάνει τα εισερχόμενα μηνύματα και τα προωθεί στον παραλήπτη τους, η σε κάποιον άλλο email server.

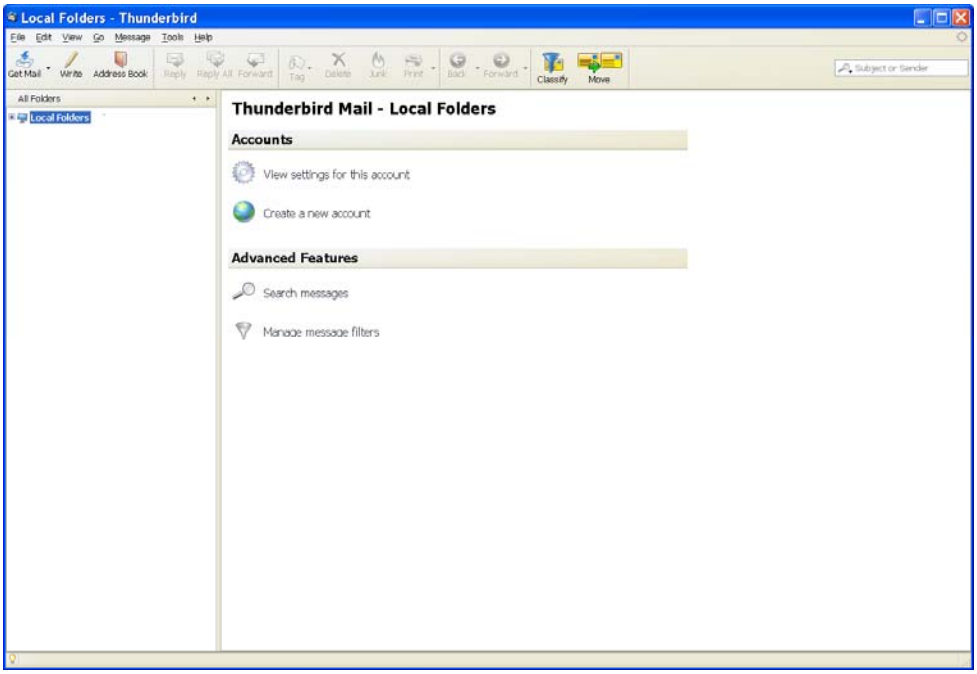
Η διαδικασία που ακολουθείται για να ανακτηθούν τα μηνύματα από τον email server είναι η εξής:
Αρχικά τα μηνύματα φτάνουν στον MTA server ο οποίος είναι υπεύθυνος για την μεταφορά των ηλεκτρονικών μηνυμάτων από ένα υπολογιστή σε κάποιον άλλο. Αν ο email client έχει πρόσβαση στον δίσκο του server, όταν ο χρήστης ενωθεί στον server αυτό, ο email client διαβάζει τα μηνύματα από το HOME directory του χρήστη. Ο MTA χρησιμοποιεί τον κατάλληλο MDA(Mail Delivery Agent) για να μεταφέρει τα ηλεκτρονικά μηνύματα στο χώρο αυτό. Ο MDA είναι υπεύθυνος για την παράδοση των ηλεκτρονικών μηνυμάτων μόλις αυτά φτάσουν στον server, κατανέμοντας τα στα mailboxes κάθε ξεχωριστού παραλήπτη. Τα μηνύματα αυτά είναι σε τυποποιημένη μορφή(π.χ. mbox). Στην περίπτωση που ο email client δεν έχει πρόσβαση στον δίσκο του server, τα μηνύματα αποθηκεύονται σε ένα μακρινό server. Για να τα ανακτήσει, ο email client συνδέεται με ένα απομακρυσμένο mailbox. Τα δυο πιο δημοφιλέστερα πρωτόκολλα τα οποία επιτρέπουν σε ένα email client να έχει πρόσβαση σε ένα mailbox για να παραλάβει τα μηνύματα, είναι το IMAP(Internet Message Access Protocol) και το POP3(Post Office Protocol).

Η διαδικασία που ακολουθείται για να σταλούν τα μηνύματα στον email server είναι η εξής:
Ο email client ενώνεται με κάποιον email server ή με κάποιον MSA(Mail submission agent) ο οποίος θα προωθήσει το μήνυμα στον παραλήπτη του. Το πρωτόκολλο για αποστολή μηνυμάτων είναι το SMTP(Simple Mail Transfer Protocol) ή το πρωτόκολλο ESMTP(extended SMTP).

Επίσης, ένας email client έχει την δυνατότητα να παρουσιάσει ένα υφιστάμενο μήνυμα ή να δημιουργήσει ένα νέο. Η δημοφιλέστερη μορφή που μπορεί να είναι το μήνυμα είναι η HTML μορφή. Ευθύνη του email client είναι να μορφοποιήσει τα headers και το σώμα(body) σύμφωνα με το RFC 5322. Τα συνημμένα αρχεία(attachments) και το περιεχόμενο που δεν είναι κείμενο(non-textual) θα τα μετατρέψει σε μορφή MIME.

2.3 Thunderbird

Ο Mozilla Thunderbird (Σχήμα 1) είναι μια εφαρμογή ελεύθερου λογισμικού παρακολούθησης ηλεκτρονικού ταχυδρομείου και RSS, η οποία διατίθεται σε λειτουργικά συστήματα Microsoft Windows, Apple Macintosh και Linux. Αναπτύχθηκε από το ίδρυμα Mozilla Foundation το 1994 και ανήκει στην κατηγορία λογισμικού ανοικτού κώδικα και διατίθεται δωρεάν.



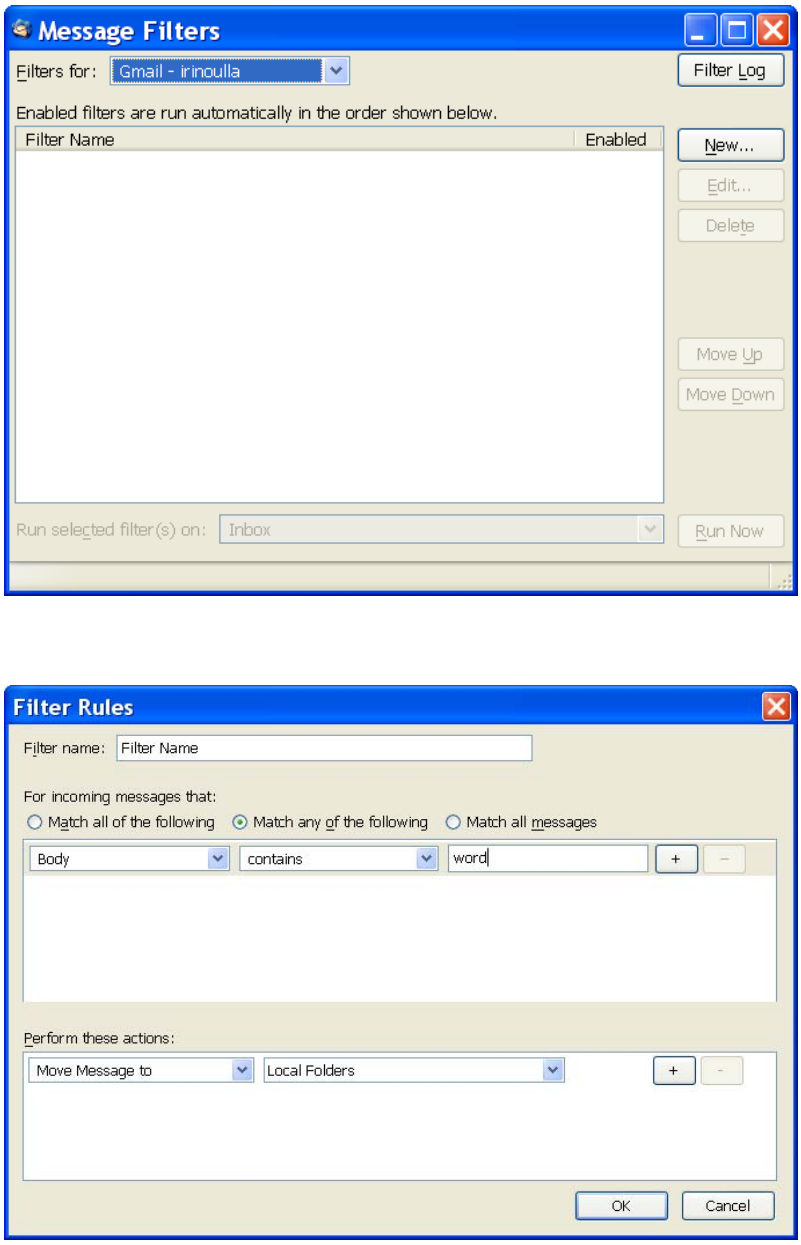
Σχήμα 1: Ο Mozilla Thunderbird

Ο Mozilla Thunderbird υποστηρίζει και τα δύο πρωτόκολλα πρόσβασης μηνυμάτων διαδικτύου IMAP και POP3 και δίνει στον χρήστη την δυνατότητα να επιλέξει ανάλογα με ποιό ταιριάζει καλύτερα στις ανάγκες του. Επίσης, παρέχει υποστήριξη για μηνύματα HTML, ομαδοποίηση μηνυμάτων, χρήση ετικετών, γρήγορη αναζήτηση, επιβεβαίωση παραλαβής και διαχείριση πολλών λογαριασμών αλληλογραφίας.

Ο Thunderbird όπως και οι περισσότεροι πελάτες ηλεκτρονικού ταχυδρομείου δίνει την δυνατότητα στο χρήστη να δημιουργήσει τοπικά τα δικά του mailboxes τα οποία μπορούμε να τα δούμε σαν φακέλους, όπου μπορεί να ταξινομεί τα εισερχόμενα ηλεκτρονικά μηνύματα σε κάθε ένα από αυτά ανάλογα με το θέμα τους ώστε να μπορεί να τα βρίσκει εύκολα στην συνέχεια. Όμως για να μην χρειάζεται να μεταφέρει ένα-ένα

τα μηνύματα στους κατάλληλους φακέλους, υπάρχουν δυο τρόποι αυτό να γίνεται αυτόματα.

Ένα άλλο εργαλείο το οποίο διαθέτει ο Thunderbird, είναι η δυνατότητα κατασκευής φίλτρων (Σχήμα 2) τα οποία θα εφαρμόζονται σε κάθε νέο ή παλιό εισερχόμενο μήνυμα. Ένα φίλτρο ηλεκτρονικού ταχυδρομείου είναι ένα κομμάτι λογισμικού το οποίο παίρνει σαν είσοδο ένα ηλεκτρονικό μήνυμα. Σαν έξοδο, μπορεί να παραδώσει το ηλεκτρονικό μήνυμα αναλλοίωτο στο ταχυδρομείο του χρήστη, μπορεί να το ανακατευθύνει για παράδοση κάπου αλλού ή ακόμα και να το πετάξει. Ο χρήστης πρέπει αρχικά να κατασκευάσει τα φίλτρα, δηλαδή τους κανόνες και τις ενέργειες τους. Τα φίλτρα μπορούν να εφαρμοστούν πάνω στα ακόλουθα χαρακτηριστικά του μηνύματος: Subject(θέμα), Body(κείμενο), From(αποστολέας), Date(ημερομηνία), Priority(Lowest, Low, Normal, High, Highest), Status(Replied, Read, New, Forwarded, Starred), To, Cc, To or Cc, Age In Days και Size. Για παράδειγμα μπορεί να ελεγχτεί αν κάποιο από τα προαναφερθέντα πεδία περιέχει κάποιες συγκεκριμένες λέξεις. Επίσης υπάρχουν διάφορες ενέργειες που μπορούν να εφαρμοστούν αν οι κανόνες ισχύουν. Αυτές είναι οι ακόλουθες: Move Message To(μετακίνηση), Copy Message To(αντιγραφή), Reply With Template(απάντηση βάση κάποιου template), Delete Message(διαγραφή), Mark As Read, Add Star, Set Priority To, Tag Message και Set Junk Status To. Κάθε φίλτρο μπορεί να ενεργοποιείται και να απενεργοποιείται.



Σχήμα 2: Κατασκευή Φίλτρων Στον Thunderbird

Με τον πιο πάνω τρόπο, ακόμα και αν οι επιλογές είναι πολλές ο χρήστης είναι περιορισμένος στο τι μπορεί να κάνει με τα φίλτρα. Ο δεύτερος τρόπος είναι η ταξινόμηση του εισερχόμενου μηνύματος με κάποιο αλγόριθμο ταξινόμησης. Αυτό μπορεί να γίνει δημιουργώντας ένα πρόσθετο στον Thunderbird το οποίο θα υλοποιεί τον αλγόριθμο.

2.4 Αποθήκευση Μηνυμάτων Στον Thunderbird

Στον Thunderbird τα ηλεκτρονικά μηνύματα είναι αποθηκευμένα στο profile folder του χρήστη σε μορφή mbox(τύπος αρχείου το οποίο χρησιμοποιείται για να κρατά συλλογές από ηλεκτρονικά μηνύματα). Σε αυτή τη μορφή, για την αποθήκευση κάθε mailbox χρησιμοποιούνται δύο αρχεία. Το πρώτο αρχείο έχει προέκταση .msf (Mail Summary File) και περιέχει ένα index σε κάθε email που βρίσκεται μέσα στο συγκεκριμένο mailbox και επίσης περιέχει τα headers(επικεφαλίδες) των μηνυμάτων. Το δεύτερο αρχείο δεν έχει προέκταση. Εδώ, τα ηλεκτρονικά μηνύματα συνενώνονται και φυλάγονται σαν απλό κείμενο(plain text). Τα attachments είναι κωδικοποιημένα ενώ τα κείμενα είναι σε 7-bit ASCII text. Κάθε ηλεκτρονικό μήνυμα σε αυτό το αρχείο ξεκινά με την λέξη “From ” (αυτή η γραμμή λέγεται From-line). Τα ηλεκτρονικά μηνύματα χωρίζονται μεταξύ τους με μια κενή γραμμή. Για να αποφύγουμε τα προβλήματα σε περιπτώσεις όπου η λέξη “From” εμφανίζεται στο σώμα του ηλεκτρονικού μηνύματος, πρέπει κατά την εισαγωγή του στο mbox αρχείο να αντικατασταθεί το “From” με κάτι άλλο, για παράδειγμα “>From”.

Το Mbox είναι ένας γενικός όρος για την περιγραφή μιας οικογένειας συσχετιζόμενων μορφών αρχείων που χρησιμοποιούνται για να φυλάνε συλλογές από ηλεκτρονικά μηνύματα. Υπάρχουν τέσσερις κύριες μορφές. Αυτές είναι mboxo, mboxrd, mboxcl, and mboxcl2. Η κάθε μια προήλθε από μια διαφορετική έκδοση Unix.

Ο Mozilla Thunderbird χρησιμοποιεί μια διαφορετική παραλλαγή της μορφής mboxrd με διαφορετικούς και πιο πολύπλοκους quoting κανόνες για τα From-lines. Ένα παράδειγμα είναι :
From - Sat Oct 17 16:24:01 2009

Οι πρώτες γραμμές του μηνύματος είναι τα headers. Τα headers περιέχουν διάφορες πληροφορίες για το μήνυμα όπως για παράδειγμα τον αποστολέα, τον παραλήπτη, το subject, την μορφή κωδικοποίησης του μηνύματος, την ημερομηνία και άλλα. Ακολουθούν τα attachments(συνημμένα αρχεία) και το body του μηνύματος. Για κάθε

ένα από αυτά καθορίζεται η κωδικοποίηση του(encoding) ώστε η εφαρμογή να αναγνωρίζει τι τύπου είναι. Τα διάφορα πεδία των headers είναι δομημένα.

Όταν χρησιμοποιείται το πρωτόκολλο IMAP για ανάκτηση των ηλεκτρονικών μηνυμάτων, αφού αποθηκεύονται μόνο τα headers, υπάρχει μόνο το .msf αρχείο. Ο χρήστης όμως έχει την επιλογή να κάνει κάποιο mailbox διαθέσιμο για χρήση όταν είναι εκτός σύνδεσης. Σε αυτή της περίπτωση τα μηνύματα αυτού του mailbox αποθηκεύονται τοπικά στον ηλεκτρονικό υπολογιστή του χρήστη και έτσι μαζί με το .msf αρχείο θα υπάρχει και το άλλο αρχείο. Ο χρήστης επίσης έχει την επιλογή να κατεβάσει όποιο μήνυμα θέλει, ακόμα και αν το mailbox στο οποίο ανήκει δεν είναι επιλεγμένο για χρήση εκτός σύνδεσης. Σε αυτή την περίπτωση θα δημιουργηθεί ένα αρχείο που θα περιέχει μόνο το συγκεκριμένο μήνυμα.

Όταν χρησιμοποιείται το πρωτόκολλο POP3 για ανάκτηση των ηλεκτρονικών μηνυμάτων, θα υπάρχουν πάντα και τα δυο αρχεία.

2.5 Πρόσθετα Thunderbird

Ένα πρόσθετο(extension) είναι ένα πρόγραμμα σχεδιασμένο για να ενσωματωθεί σε κάποιο άλλο λογισμικό με σκοπό να αλλάξει την συμπεριφορά των υπάρχων χαρακτηριστικών, ή/και ακόμα να προσθέσει επιπλέον χαρακτηριστικά σε μια εφαρμογή. Με αυτό τον τρόπο, ο χρήστης έχει την δυνατότητα να προσαρμόσει την εφαρμογή αυτή στις δικές του ανάγκες, κρατώντας το μέγεθος της μικρό. Στην περίπτωση μας η εφαρμογή αυτή είναι ο Mozilla Thunderbird.

Ένα πρόσθετο Thunderbird έχει επέκταση xpi. Το XPI(Cross-Platform Installer Module) είναι ένα zip αρχείο που χρησιμοποιείται για την εγκατάσταση πακέτων και αξιοποιεί την τεχνολογία XPInstall. Περιέχει οδηγίες εγκατάστασης και τα δεδομένα που πρόκειται να εγκατασταθούν. Οι οδηγίες εγκατάστασης βρίσκονται σε ένα manifest αρχείο το οποίο βρίσκεται στη ρίζα του XPI αρχείου. Αποτελείται από ένα chrome manifest και ένα install manifest. Το chrome manifest περιέχει πληροφορίες για το που βρίσκονται τα chrome files και τι πρέπει να κάνει με αυτά και έχει όνομα chrome.manifest. Το install manifest περιέχει πληροφορίες για το πρόσθετο οι οποίες

χρησιμοποιούνται κατά την διάρκεια της εγκατάστασης του. Με άλλα λόγια περιέχει μεταδεδομένα όπως για παράδειγμα πληροφορίες για το ποιος το δημιούργησε, με ποιες εκδόσεις της εφαρμογής είναι συμβατό κτλ.

Πιο συγκεκριμένα, το install manifest είναι ένα αρχείο με το όνομα install.rdf και έχει μορφή RDF/XML. Το RDF(Resource Description Framework) είναι μια γλώσσα γενικού σκοπού για αναπαράσταση πληροφοριών που βρίσκονται στο διαδίκτυο. Είναι ένα W3C πρότυπο(World Wide Web Consortium standard) που σχεδιάστηκε για να διαβάζεται και να γίνεται κατανοητό από τους ηλεκτρονικούς υπολογιστές. Συνήθως υλοποιείται μέσω αρχείων που έχουν XML μορφή. Ένα RDF Statement είναι μια τριάδα subject, predicate, object τα οποία προσδιορίζονται μέσω URIs(Uniform Resource Identifiers).

Πιο κάτω φαίνεται ένα απλό παράδειγμα κάποιου install manifest αρχείου.

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:em="http://www.mozilla.org/2004/em-rdf#">

     <Description about="urn:mozilla:install-manifest">

          <em:id>bayesemailclassifier@cs.ucy</em:id>
          <em:version>1.0</em:version>
          <em:name>bayesemailclassifier!</em:name>
          <em:description>An extension that use Naive Bayes
algorithm to classify Email into folders</em:description>
          <em:creator>Eirini Eleftheriou</em:creator>

     </Description>
</RDF>
```

Το αρχείο chrome manifest ενημερώνει τον Thunderbird ποια πακέτα και ποια overlays υπάρχουν στον πρόσθετο.

Τα τρία είδη chrome packages είναι:

- **Content** - Περιέχουν τα XUL έγγραφα, τα οποία περιγράφουν τον λεπτομερή σχεδιασμό του user interface.
- **Skin** – Είναι τα αρχεία CSS και οι εικόνες τα οποία ορίζουν την εμφάνιση της εφαρμογής. Ο λόγος που φυλάγονται ξεχωριστά από τα αρχεία XUL είναι για να είναι πιο εύκολη η τροποποίηση του skin (theme) κάποιας εφαρμογής.
- **Locale** – Είναι αρχεία που περιέχουν το κείμενο το οποίο βλέπει ο χρήστης. Με αυτό τον τρόπο, η εφαρμογή μπορεί να είναι διαθέσιμη σε διάφορες γλώσσες πολύ εύκολα.

Η ακόλουθη γραμμή καθορίζει που βρίσκονται τα content αρχεία για το chromePackageName.

```
content   chromePackageName   packageDirectoryLocation/
```

Το overlay είναι ένα χαρακτηριστικό της XUL το οποίο επιτρέπει στα XUL αρχεία που αποτελούν το πρόσθετο να συνδυαστούν με τα αντίστοιχα XUL αρχεία της εφαρμογής. Για να αλλάξει το user interface του Thunderbird, πρέπει να δημιουργηθεί ένα νέο overlay και στη συνέχεια αυτό να συνενωθεί με το προκαθορισμένο interface του Thunderbird. Στο αρχείο chrome manifest πρέπει να καθοριστεί η ύπαρξη του νέου overlay, δηλαδή:

```
overlay   chrome://URI-to-be-overlaid   chrome://overlay-URI
```

Έτσι, όταν φορτωθεί το chrome://URI-to-be-overlaid, θα συνενωθεί με το chrome://overlay-URI, το οποίο υπάρχει ήδη στην τοποθεσία που είναι εγκατεστημένη η εφαρμογή. Τα chrome URLs είναι ένας τρόπος για να αναφερθεί κάποιος σε πακέτα που βρίσκονται εγκατεστημένα στον Thunderbird αλλά και σε οποιαδήποτε άλλη Mozilla εφαρμογή, χωρίς να χρειάζεται να γνωρίζει που ακριβώς είναι εγκατεστημένη η εφαρμογή.

Το user interface του Thunderbird είναι γραμμένο σε XUL(XML User-interface Language) και JavaScript. Η XUL είναι μια cross-platform γλώσσα η οποία χρησιμοποιείται για περιγραφή των user interfaces των εφαρμογών. Η λειτουργικότητα τους καθορίζεται από τα JavaScript αρχεία. Ένα XUL αρχείο μπορεί να έχει ένα ή περισσότερα script αρχεία που συσχετίζεται με αυτό/ά. Η Mozilla χειρίζεται την XUL ακριβώς με τον ίδιο τρόπο όπως την HTML.

Η XUL παρέχει την δυνατότητα για δημιουργία σχεδόν όλων των element που χρησιμοποιούνται στα μοντέρνα graphical interfaces. Μερικά από αυτά είναι:

- Textbox
- Checkboxes
- Toolbars
- Pop up menu
- Tabbed dialog
- Συντομεύσεις πληκτρολογίου

Η JavaScript είναι μια αντικειμενοστρεφής(object-oriented) scripting γλώσσα προγραμματισμού η οποία τρέχει σε πολλές πλατφόρμες(cross-platform). Ονομάζεται έτσι γιατί η σύνταξη της μοιάζει πολύ με αυτήν της Java, όμως είναι διαφορετική γλώσσα προγραμματισμού και όχι απλά η Java με διερμηνέα. Επίσης, η JavaScript δεν αναπτύχθηκε από την Sun Microsystems όπως η Java, αλλά από την Netscape και πιο συγκεκριμένα από τον Brendan Eich. Το αρχικό της όνομα ήταν LiveScript.

Ουσιαστικά η JavaScript είναι μια δυναμική scripting γλώσσα και υποστηρίζει prototype based κατασκευή αντικειμένων, δηλαδή τα αντικείμενα είναι instances. Οι δυναμικές ικανότητες της συμπεριλαμβάνουν κατασκευή αντικειμένων κατά την διάρκεια εκτέλεσης του script, μεταβλητή λίστα παραμέτρων, μεταβλητές συναρτήσεων, δυναμική δημιουργία script μέσω της eval και πολλά άλλα.

Το Document Object Model (DOM), είναι ένα API για HTML και XML έγγραφα. Αναπαριστά τα έγγραφα με δομημένο(structural) τρόπο και έτσι σου επιτρέπει να τροποποιήσεις το περιεχόμενο τους. Όταν φορτωθεί ένα XML ή HTML αρχείο, τα tags

του αναλύονται και μετατρέπονται σε μια ιεραρχική δομή από κόμβους, μια για κάθε tag.

Στην Mozilla, πρόσβαση στο DOM εξασφαλίζεται μέσω του JavaScript, δηλαδή τα διάφορα DOM αντικείμενα διαθέτουν συναρτήσεις στις οποίες μπορεί κάποιος να έχει πρόσβαση μέσω του JavaScript και έτσι για παράδειγμα μπορεί κάποιος να διαχειριστεί τους διάφορους κόμβους του XUL. Δηλαδή αφού η JavaScript ορίζει την συμπεριφορά, χρησιμοποιεί τα DOM APIs για να έχει πρόσβαση στα XUL έγγραφα. Έτσι το DOM χρησιμοποιείται για δυναμική τροποποίηση του User Interface(UI).

Το XPCOM(Cross-platform Component Object Model) είναι ένα framework το οποίο επιτρέπει στους προγραμματιστές να αναπτύσσουν αρθρωτό(modular) λογισμικό το οποίο υποστηρίζεται από πολλές πλατφόρμες. Τα XPCOM components μπορούν να χρησιμοποιηθούν και να υλοποιηθούν σε C, C++, JavaScript, Java, και Python. Κατά την ώρα της εκτέλεσης, τα διάφορα components συνενώνονται ξανά μαζί. Τα Interfaces στο XPCOM ορίζονται σε IDL(Interface Description Language) διάλεκτο η οποία ονομάζεται XPIDL. Η IDL είναι μια καθοριστική γλώσσα η οποία χρησιμοποιείται για περιγραφή των interface των συστατικών κάποιου λογισμικού. Τα IDL περιγράφουν τα interface με τρόπο ώστε να επιτρέπεται η επικοινωνία μεταξύ συστατικών του λογισμικού τα οποία δεν είναι γραμμένα στην ίδια γλώσσα.

2.6 Πρωτόκολλα Μεταφοράς Ηλεκτρονικού Ταχυδρομείου

Το ηλεκτρονικά μηνύματα συνήθως στέλνονται σε ένα email server ο οποίος τα φυλάει στον mailbox του παραλήπτη. Ο χρήστης μπορεί να τα ανακτήσει με δυο τρόπους: Είτε μέσω του φυλλομετρητή του(web browser), είτε μέσω κάποιου email client ο οποίος χρησιμοποιεί κάποια πρωτόκολλα για ανάκτηση αυτών των μηνυμάτων. Τα πιο διαδεδομένα πρωτόκολλα είναι το IMAP και το POP3. Το πρωτόκολλο IMAP υποστηρίζεται από λιγότερους servers σε σχέση με το πρωτόκολλο POP3.

Το IMAP (Internet Message Access Protocol = πρωτόκολλο πρόσβασης μηνυμάτων διαδικτύου), είναι ένα από τα πιο διαδεδομένα πρωτόκολλα(μαζί με το POP3) για πρόσβαση των ηλεκτρονικών μηνυμάτων τα οποία βρίσκονται σε κάποιο

απομακρυσμένο email server. Σχεδιάστηκε το 1986 στο Stanford University από τον Mark Crispin και η πιο πρόσφατη έκδοση είναι το IMAP4. Το πρωτόκολλο IMAP λειτουργεί στη θύρα 143. Απαιτεί συνεχή πρόσβαση στον κεντρικό υπολογιστή κατά τη διάρκεια που κάποιος χρήστης έχει πρόσβαση στο ταχυδρομείο του.

Τα ηλεκτρονικά μηνύματα τα οποία είναι φυλαγμένα σε κάποιο IMAP server παραμένουν εκεί μέχρι ο χρήστης να τα διαγράψει απευθείας από αυτόν. Επιπλέον, χρησιμοποιεί flags τα οποία κρατούν την κατάσταση του μηνύματος, π.χ. αναγνωσμένο, απαντημένο κτλ. Για τους πιο πάνω λόγους, ο χρήστης έχει την δυνατότητα να τα διαχειριστεί τα μηνύματα του από πολλούς ηλεκτρονικούς υπολογιστές χρησιμοποιώντας πολλούς email clients, χωρίς να υπάρχει η οποιαδήποτε σύγχυση.

Επίσης, το IMAP παρέχει μηχανισμούς για τη δημιουργία, καταστροφή, και διαχείριση πολλαπλών γραμματοκιβωτίων στον email client. Δηλαδή ο χρήστης μπορεί να δημιουργήσει διάφορους φακέλους στους οποίους θα οργανώσει τα μηνύματα του, και θα μπορεί να τα διαχειριστεί από όποιο υπολογιστή έχει στην διάθεση του.

Τα ηλεκτρονικά μηνύματα δεν αποθηκεύονται τοπικά στον ηλεκτρονικό υπολογιστή, εκτός αν ο χρήστης το επιθυμεί. Το μόνο που αποθηκεύεται είναι τα headers των μηνυμάτων για να μπορεί ο email client να τα παρουσιάσει στον χρήστη και όταν επιλέξει ένα από αυτά, το μήνυμα θα παραληφθεί από τον email server. Δηλαδή όταν ο email client ζητήσει από τον email server να του στείλει τα νέα μηνύματα, ο email server θα στείλει μόνο τα headers του μηνύματος. Αυτό το κάνει πιο γρήγορο γιατί δεν χρειάζεται να κατεβούν όλα τα μηνύματα.

Το POP3(Post Office Protocol version 3) είναι η πιο πρόσφατη έκδοση ενός τυποποιημένου πρωτοκόλλου για τη λήψη του ηλεκτρονικού ταχυδρομείου. Ένας POP3 email server ακούει στην θύρα 110 χρησιμοποιώντας TCP/IP σύνδεση.

Το πρωτόκολλο POP3 είναι πιο απλό από το πρωτόκολλο IMAP. Όταν ένας POP server λάβει ένα ηλεκτρονικό μήνυμα, το αποθηκεύει εκεί μέχρι κάποιος email client να το ζητήσει. Όταν γίνει αυτό, ολόκληρο το μήνυμα θα σταλεί(όχι μόνο τα headers) και θα

αποθηκευτεί τοπικά στον ηλεκτρονικό υπολογιστή του χρήστη. Στη συνέχεια θα διαγραφεί από τον email server, εκτός αν καθοριστεί να διατηρείται αντίγραφο εκεί.

Για τους πιο πάνω λόγους, ο χρήστης δεν μπορεί να διαχειρίζεται το ηλεκτρονικό του ταχυδρομείο από πολλούς ηλεκτρονικούς υπολογιστές, γιατί υπάρχει περίπτωση τα μηνύματα να μην υπάρχουν πλέον στον server. Έχει όμως την δυνατότητα να διαχειρίζεται το γραμματοκιβώτιο του χωρίς να χρειάζεται να έχει συνεχή σύνδεση με τον server, αφού τα ηλεκτρονικά μηνύματα διατηρούνται τοπικά στον ηλεκτρονικό υπολογιστή του.

Ο Mozilla Thunderbird υποστηρίζει και τα δυο πρωτόκολλα για μεταφορά των ηλεκτρονικών μηνυμάτων από κάποιο server.

Για αποστολή μηνυμάτων, υπάρχει το πρωτόκολλο SMTP (Simple Mail Transfer Protocol) ή και ESMTP (extended SMTP). Το ESMTP περιλαμβάνει προσθήκες στο SMTP. Το πρωτόκολλο SMTP ακούει στην θύρα 25.

Κεφάλαιο 3

Ταξινόμηση

3.1 Ανάκτηση Πληροφοριών

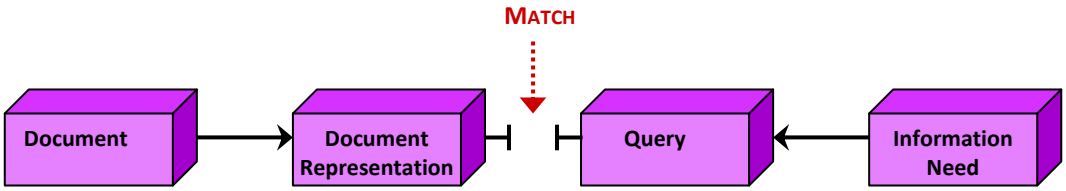
Ανάκτηση πληροφοριών είναι η διαδικασία εύρεσης υλικού αδόμητης φύσης(δηλαδή που δεν έχει σημασιολογικά φανερή δομή) και το οποίο ικανοποιεί μια ανάγκη για πληροφορία από μεγάλες συλλογές.

Όπως φαίνεται στο Σχήμα 3, στόχος είναι να βρεθεί ένα ταίριασμα μεταξύ κάποιου επερωτήματος που θέτει ο χρήστης και μιας συλλογής από έγγραφα. Η ανάγκη για πληροφορία μεταφράζεται σε μορφή επερωτήματος.

Αρχικά ξεκίνησε με επιστημονικές δημοσιεύσεις. Στη συνέχεια χρησιμοποιήθηκε για να παρέχει πρόσβαση σε πληροφορίες οι οποίες δεν είχαν δομή και τα τελευταία χρόνια, το διαδίκτυο έφερε την μεγαλύτερη καινοτομία.

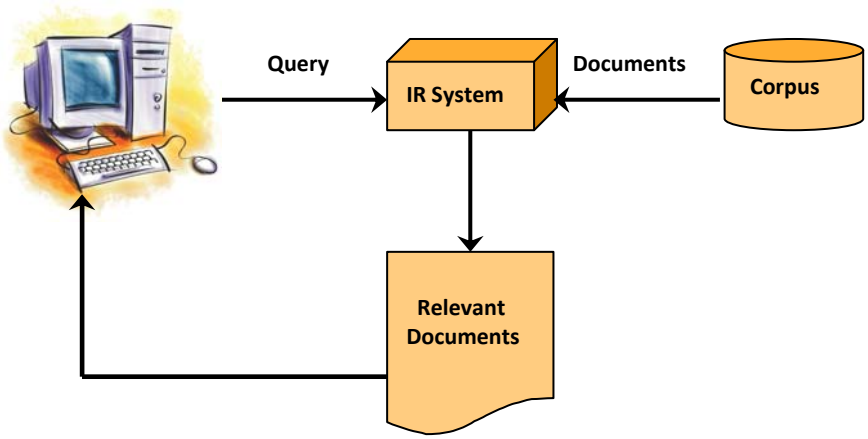
Η ανάκτηση πληροφοριών μπορεί να αντλήσει πληροφορίες από πολλά άλλα πεδία όπως για παράδειγμα βάσεις δεδομένων, τεχνητή νοημοσύνη, κατανόηση φυσικής γλώσσας και μηχανική μάθηση. Μερικές από τις χρήσεις της ανάκτησης πληροφοριών είναι: υποβοήθηση φιλτραρίσματος πληροφοριών και πλοήγησης σε πληροφοριακούς χώρους, προσωπική ανάκτηση πληροφοριών, επιχειρησιακή ανάκτηση πληροφοριών, ομαδοποίηση πληροφοριών (clustering) και ταξινόμηση (classification).

Η παρούσα διπλωματική εργασία θα ασχοληθεί με την ταξινόμηση και πιο συγκεκριμένα με την ταξινόμηση ηλεκτρονικών μηνυμάτων. Με τον όρο ταξινόμηση εννοούμε την διαδικασία κατά την οποία ένα αντικείμενο(στην περίπτωση μας ηλεκτρονικό μήνυμα) ανατίθεται σε μια ή περισσότερες προκαθορισμένες κατηγορίες/κλάσεις(καταλόγους).



Σχήμα 3: Στόχος Συστήματος Ανάκτησης Πληροφοριών

Από το Σχήμα 4 φαίνεται πως λειτουργεί ένα σύστημα ανάκτησης πληροφοριών. Σαν είσοδο δέχεται μια επερώτηση σε μορφή συμβολοσειράς και με βάση την συλλογή από έγγραφα που έχει στην διάθεση του αποφασίζει ποια από αυτά είναι σχετικά με την επερώτηση, δηλαδή με αυτό που αναζητεί ο χρήστης του.



Σχήμα 4: Περιγραφή Συστήματος Ανάκτησης Πληροφοριών

Για να το επιτύχουν το πιο πάνω πρέπει να αντιμετωπίσουν την αβεβαιότητα ως προς το τι πραγματικά αναζητεί ο χρήστης και ποιο είναι το θέμα ενός εγγράφου.

3.2 Αναπαράσταση Εγγράφων

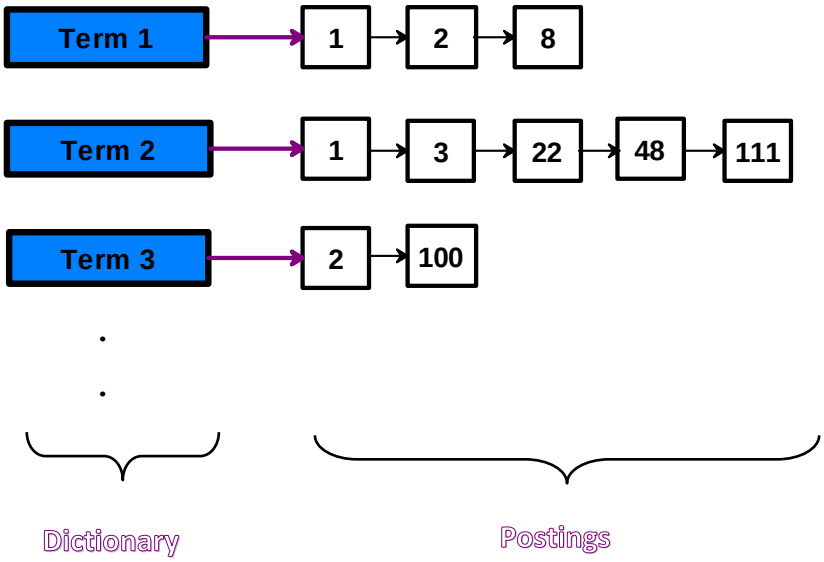
Για να διευκολύνουμε την όλη διαδικασία, χρειάζεται τα χρησιμοποιήσουμε κάποιου είδους ευρετήριο(index). Έτσι, για κάθε έγγραφο καταγράφεται κατά πόσον περιέχει κάθε όρο που περιέχεται σε κάθε έγγραφο της συλλογής. Το αποτέλεσμα θα είναι ένας δυαδικός term-document incidence matrix (μητρώο παρουσίας). Ένα παράδειγμα φαίνεται στον Πίνακας 1:

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	…
Term 1	0	1	0	1	0	1	…
Term 2	1	0	1	1	0	0	…
Term 3	0	1	0	1	1	1	…
Term 4	0	1	0	0	0	0	…
Term 5	1	0	1	0	1	0	…
…	…	…	…	…	…	…	…

Πίνακας 1: Term-Document Incidence Matrix

Το στοιχείο (*t*, *d*) είναι 1 αν το έγγραφο στη στήλη *d* περιέχει τον όρο στην γραμμή *t*, διαφορετικά είναι 0.

Ο term-document incidence matrix δεν μπορεί να χρησιμοποιηθεί σε περιπτώσεις όπου η συλλογή και τα έγγραφα είναι πολύ μεγάλα. Δεδομένου ότι ένας τέτοιος πίνακας είναι πολύ αραιός, μια πιο καλή προσέγγιση είναι να καταγράφουμε μόνο τις θέσεις που έχουν 1, δηλαδή μόνο τους όρους που εμφανίζονται. Αποτέλεσμα θα είναι η δημιουργία του λεγόμενου inverted index όπου για κάθε όρο t υπάρχει μια λίστα με όλα τα έγγραφα που περιέχουν τον t. Ένα παράδειγμα φαίνεται στο Σχήμα 5.



Σχήμα 5: Inverted Index

Λεξικό(Dictionary) είναι οι όροι. Για κάθε όρο(posting) υπάρχει μια λίστα(posting list) που περιέχει όλα τα έγγραφα στα οποία υπάρχει ο όρος αυτός.

Βήματα για την κατασκευή του inverted index:

1. Βρίσκουμε τα έγγραφα που θα γίνουν indexed.
   Εδώ πρέπει να μετατρέψουμε την κάθε ακολουθία από bytes από τις οποίες αποτελούνται τα (ψηφιακά) έγγραφα σε μια γραμμική ακολουθία χαρακτήρων, η οποία μπορεί να είναι encoded(κωδικοποιημένη) από διάφορα encoding schemes(π.χ. UTF-8).
2. Κάνουμε **tokenization** το κείμενο.

Η διαδικασία κατά την οποία μια σειρά από λέξεις μετατρέπεται σε ένα σύνολο από tokens. Τα σημεία στίξης ή κάποιοι ειδικοί χαρακτήρες δεν συμπεριλαμβάνονται στα tokens.

3. Κάνουμε **γλωσσολογική επεξεργασία** στα tokens.
4. Κάνουμε το **indexing**.

3.3 Λέξεις Αποκλεισμού

Τα έγγραφα, εκτός από τις σημαντικές λέξεις οι οποίες χαρακτηρίζουν το θέμα του εγγράφου, περιέχουν επίσης πολλές λέξεις οι οποίες δεν έχουν καμία χρησιμότητα κατά την διαδικασία ανάκτησης πληροφορίας και δεν πρέπει να λαμβάνονται υπόψη γιατί επιστρέφουν περιττές πληροφορίες, είτε επειδή είναι ασήμαντες, π.χ. άρθρα, προθέσεις, συνδέσεις κ.τ.λ., είτε επειδή είναι τόσο κοινές που τα αποτελέσματα θα ήταν υψηλότερα από αυτά που μπορεί το σύστημα να χειριστεί. Αυτές οι λέξεις ονομάζονται λέξεις αποκλεισμού.

Η κύρια στρατηγική για τον καθορισμό των λέξεων αποκλεισμού είναι να ταξινομήσουμε τους όρους βάσει του συνολικού αριθμού που εμφανίζεται ο κάθε όρος στην συλλογή εγγράφων(collection frequency) και στην συνέχεια να πάρουμε τους όρους που εμφανίζονται πιο συχνά. Αυτοί οι όροι αποτελούν την stop list και θα απορρίπτονται κατά το indexing. Ένας άλλος τρόπος είναι να μην συμπεριλάβουμε τους όρους οι οποίοι είναι γνωστοί ως λέξεις αποκλεισμού.

3.4 Κανονικοποίηση

Κανονικοποίηση είναι η διαδικασία κατά την οποία οι αντιστοιχίες μεταξύ των tokens να γίνονται παρά τις επιφανειακές διαφορές στις ακολουθίες χαρακτήρων των tokens.
Μερικές μορφές κανονικοποίησης είναι:

1. Accents and diacritics:

Εξαλείφουμε οποιαδήποτε σημεία στίξης και διαχωριστικά.

2. Capitalization/case-folding:

Κάνω όλα τα γράμματα σε μικρά (όχι κεφαλαία).

3.5 Στελέχωση Κειμένου (Stemming)

Είναι μια ευρετική (heuristic) διαδικασία κατά την οποία αφαιρείται από μια λέξη η κατάληξη της. Κάποια πειράματα έδειξαν πως το Stemming σε πολύ λίγες περιπτώσεις μπορεί να αποφέρει θετικά αποτελέσματα, και σε αυτές τις περιπτώσεις έχει πολύ λίγο αντίκτυπο στην απόδοση [6]. Κάποια άλλοι αναφέρουν πως αν το Stemming εφαρμοστεί σε κάποιες συγκεκριμένες συλλογές, μπορεί να αυξήσει την απόδοση της ανάκτησης 15-35% [10].

Ο αλγόριθμος Porter [14] είναι ο πιο διαδεδομένος αλγόριθμος για στελέχωση κειμένου. Είναι rule-based stemmer, δηλαδή καθορίζει κάποιους κανόνες οι οποίοι εφαρμόζονται στους όρους και βάση αυτών των κανόνων αφαιρούνται οι καταλήξεις τους.

3.6 Λημματοποίηση (Lemmatization)

Είναι μια διαδικασία κατά την οποία κάθε λέξη μεταφέρνεται στην άκλιτη μορφή της. Η διαφορά από το stemming είναι ότι ένας stemmer δεν μπορεί να διακρίνει ανάμεσα σε λέξεις οι οποίες έχουν διαφορετικό νόημα ανάλογα με το τι μέρος του λόγου είναι.

3.7 Κλασσικά Μοντέλα Ανάκτησης

Υπάρχουν τρία βασικά μοντέλα τα οποία χρησιμοποιούνται για την αναπαράσταση των εγγράφων και των επερωτημάτων. Αυτά είναι το Boolean μοντέλο, το Vector Space μοντέλο και το πιθανοκρατικό(probabilistic) μοντέλο.

Το Boolean μοντέλο είναι ένα μοντέλο ανάκτησης πληροφοριών του οποία τα επερωτήματα εκφράζονται σε Boolean μορφή, δηλαδή οι όροι συνδυάζονται μεταξύ τους χρησιμοποιώντας τους τελεστές AND, OR και NOT.

Στο Vector Space μοντέλο, κάθε document αναπαριστάται σαν ένα διάνυσμα όπου κάθε διάσταση αντιστοιχεί σε κάποιο όρο. Η τιμή του διανύσματος για κάποιον όρο ονομάζεται βάρος(weight) του όρου αυτού και υπάρχουν διάφοροι τρόποι για να

υπολογιστεί. Ένας τρόπος είναι το tf-idf(term frequency–inverse document frequency) weighting το οποίο δίνεται από τον πιο κάτω τύπο:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

Στον πιο πάνω τύπο πολλαπλασιάζονται δυο όροι. Ο πρώτος όρος ονομάζεται term frequency και υπολογίζεται ως ο αριθμός εμφανίσεων του όρου t στο έγγραφο d, δηλαδή:

$$\text{tf}_{t,d} = \text{αριθμός εμφανίσεων του όρου t στο έγγραφο d}$$

Ο δεύτερος όρος ονομάζεται inverse document frequency. Για να υπολογιστεί, χρειάζεται πρώτα πρέπει να υπολογιστεί το document frequency το οποίο ισούται με τον αριθμό των εγγράφων που περιέχουν τον όρο t, δηλαδή:

$$\text{df}_t = \text{αριθμός των εγγράφων που περιέχουν τον όρο t}$$

Τέλος, το inverse document frequency υπολογίζεται με τον ακόλουθο τύπο:

$$\text{idf}_t = \log \frac{N}{\text{df}_t}$$

όπου N είναι ο συνολικός αριθμός εγγράφων στην συλλογή.

Δηλαδή το $\text{tf-idf}_{t,d}$ αναθέτει στον όρο t ένα βάρος στο έγγραφο d το οποίο είναι:

1. Μεγαλύτερο όταν ο όρος t εμφανίζεται πολλές φορές μέσα σε μια μικρή συλλογή από έγγραφα.
2. Μικρότερο όταν ο όρος t εμφανίζεται λιγότερες φορές μέσα σε κάποιο έγγραφο ή εμφανίζεται σε πολλά έγγραφα.
3. Ελάχιστο όταν ο όρος t εμφανίζεται σχεδόν σε όλα τα έγγραφα.

Για να μετρήσουμε το score κάποιου εγγράφου για κάποια επερώτηση(την οποία επίσης θα αναπαραστήσουμε σαν διάνυσμα), μπορούμε να χρησιμοποιήσουμε το cosine similarity, δηλαδή:

$$\text{score(q,d)} = \cos(\vartheta) = \frac{\vec{V}(q)\cdot\vec{V}(d)}{\left|\vec{V}(q)\right|\left|\vec{V}(d)\right|}$$

όπου ο αριθμητής είναι το εσωτερικό γινόμενο των δύο διανυσμάτων $\vec{V}(q)$ και $\vec{V}(d)$, δηλαδή $\sum_{i=1}^{M} x_i y_i$ και ο παρονομαστής είναι το γινόμενο των Ευκλείδειων αποστάσεων τους, δηλαδή $\sqrt{\sum_{i=1}^{M}\overrightarrow{V_i^2}(d)}$. Στην ουσία, το cosine similarity υπολογίζει συνημίτονο της γωνίας μεταξύ των δυο διανυσμάτων.

Το vector space μοντέλο χρησιμοποιείται για scoring, ταξινόμηση(classification) εγγράφων και ομαδοποίηση(clustering) εγγράφων.

Το πιθανοκρατικό μοντέλο βασίζεται στη ιδέα ότι το πρόβλημα ανάθεσης κάποιας μετρικής σχετικότητας μεταξύ ενός εγγράφου και ενός επερωτήματος μπορεί να γίνει ένα πρόβλημα εύρεσης της πιθανότητας το έγγραφο να είναι σχετικό με το επερώτημα. Για εκτίμηση της πιθανότητας χρησιμοποιούνται οι λέξεις που εμφανίζονται στο έγγραφο και στο επερώτημα αντίστοιχα.

3.8 Μετρικές

Η αποτελεσματικότητα (effectiveness) κάποιου συστήματος, δηλαδή η ποιότητα των αποτελεσμάτων του, μπορεί να υπολογιστεί χρησιμοποιώντας τις ακόλουθες μετρικές:

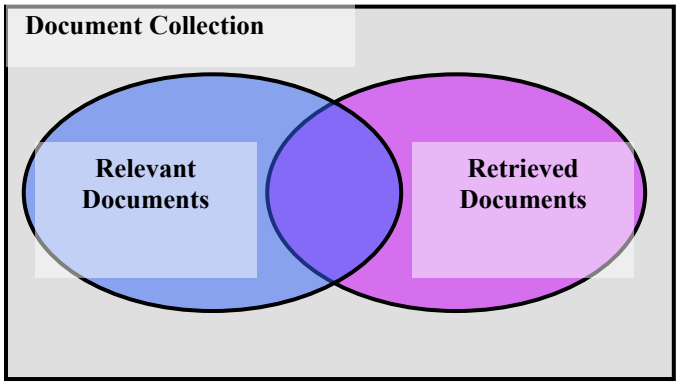
Precision (ακρίβεια): ο αριθμός των σχετικών εγγράφων που έχουν ανακτηθεί από το σύστημα δια των αριθμό των εγγράφων που έχουν ανακτηθεί.

$$precision = \frac{|\{relevant\_documents\}\cap\{documents\_retrieved\}|}{|\{documents\_retrieved\}|} = P\left(relevant\middle|retrieved\right)$$

Recall (πληρότητα): ο αριθμός των σχετικών εγγράφων που έχουν ανακτηθεί από το σύστημα δια των αριθμό των σχετικών εγγράφων (δηλαδή αυτά που θα έπρεπε να είχαν ανακτηθεί).

$$recall = \frac{|\{relevant\_documents\} \cap \{documents\_retrieved\}|}{|\{relevant\_documents\}|} = P(retieved|relevant)$$

Πιο κάτω φαίνεται το διάγραμμα Venn ανάμεσα στα δύο σύνολα.



Σχήμα 6: Διάγραμμα Venn μεταξύ των εγγράφων που έχουν ανακτηθεί και των εγγράφων που έπρεπε να είχαν ανακτηθεί

Πιο κάτω φαίνεται ένας πίνακας που περιέχει τις διάφορες περιπτώσεις που μπορεί να προκύψουν:

	Relevant	Not Relevant
Retrieved	True Positive (a)	False Positive (b)
Not Retrieved	False Negative (c)	True Negative (d)

Πίνακας 2: Μήτρα Σύγχυσης(Confusion Matrix)

Έτσι:

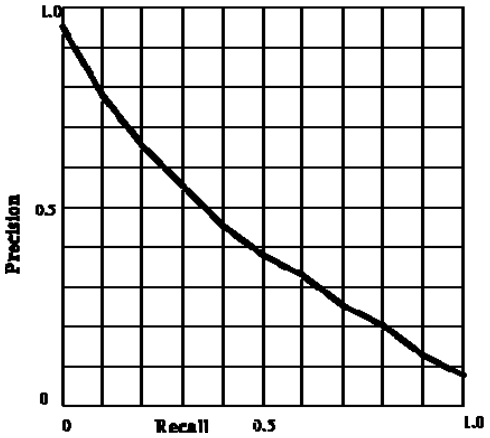
Precision = a / (a + b)

Recall = a / (a + c)

Για να μετρήσουμε την ορθότητα(accuracy) του συστήματος μετρούμε πόσες από τις ταξινομήσεις που έκανε(σαν σχετικό ή όχι) είναι ορθές, άρα:

$$\text{Accuracy} = (a + d) / (a + b + c + d)$$

Παρακάτω παρουσιάζεται μια τυπική γραφική παράσταση της ακρίβειας και της πληρότητας. Όπως φαίνεται, η γραφική παράσταση αυτή έχει κοίλο σχήμα. Συμπερασματικά, αν μεγαλώσει η ακρίβεια, μειώνεται η πληρότητα και αν μεγαλώσει η πληρότητα, μειώνεται η ακρίβεια.



Σχήμα 7: Γραφική Παράσταση Ακρίβειας Και Πληρότητας

Μια άλλη μετρική η οποία λαμβάνει υπόψη και την ακρίβεια και την πληρότητα είναι το F-measure το οποίο είναι ο αρμονικός μέσος(harmonic mean) μεταξύ της ακρίβειας και της πληρότητας. Ο γενικός τύπος είναι:

$$F_\beta = \frac{1}{\alpha \frac{1}{precision} + (1-\alpha)\frac{1}{recall}} = \frac{(1+\beta^2)\cdot precision \cdot recall}{\beta^2 precision + recall}$$

όπου $\beta^2 = \frac{1-\alpha}{\alpha}$ και $\alpha \in [0,1]$.

Αν $\beta < 1$, δίνεται περισσότερη έμφαση στην ακρίβεια.

Αν $\beta > 1$, δίνεται περισσότερη έμφαση στην πληρότητα.

Αν $\beta = 1$: $F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$

3.9 Ταξινόμηση

Δεδομένης μιας συλλογής από έγγραφα(training set) D={d_1, d_2, …, d_n} και ενός συνόλου από προκαθορισμένες κλάσεις ή κατηγορίες C={ c_1, c_2, …, c_m} όπου κάθε έγγραφο ανήκει σε κάποια από τις κατηγορίες, Ταξινόμηση ή Κατηγοριοποίηση είναι η διαδικασία ορισμού μιας συνάρτησης στόχου f: D→C που απεικονίζει κάθε έγγραφο D σε μια κλάση C.

Η Ταξινόμηση είναι μια πολύ γενική έννοια. Μερικά παραδείγματα είναι τα ακόλουθα:

- Αυτόματος εντοπισμός ηλεκτρονικών μηνυμάτων που είναι spam
- Αυτόματος εντοπισμός σεξουαλικού περιεχομένου σε αποτελέσματα κάποιας μηχανής αναζήτησης, ώστε να είναι επιλογή του χρήστη αν θα παρουσιαστούν
- Αυτόματη ταξινόμηση ηλεκτρονικού ταχυδρομείου, δηλαδή ταξινόμηση των εισερχόμενων ηλεκτρονικών μηνυμάτων στα mailboxes που διαθέτει ο χρήστης ώστε να είναι πιο εύκολη η ανεύρεση τους. Για παράδειγμα ένας φάκελος με όνομα Friends περιέχει όλα τα μηνύματα που έχουν σχέση με το συγκεκριμένο θέμα.

Υπάρχουν τρείς μέθοδοι κατηγοριοποίησης. Η πρώτη μέθοδος είναι η κατηγοριοποίηση να γίνει χειρονακτικά (manual classification). Αυτή η μέθοδος μπορεί να χρησιμοποιηθεί αν το μέγεθος της συλλογής είναι μικρό και δεν μεταβάλλεται συχνά, διαφορετικά η εφαρμογή της μεθόδου αυτής είναι αρκετά δύσκολη και ακριβή. Η δεύτερη μέθοδος είναι η αυτόματη ταξινόμηση (automatic classification) η οποία βασίζεται σε κανόνες. Ένα παράδειγμα είναι τα standing queries. Και αυτή η μέθοδος είναι ακριβή και για να πετύχει πρέπει οι κανόνες να οριστούν προσεκτικά από κάποιο

ειδικό. Η τρίτη μέθοδος ονομάζεται κατηγοριοποίηση βασισμένη σε μηχανική μάθηση (machine learning-based text classification) ή στατιστική κατηγοριοποίηση (statistical text classification) αν η μέθοδος μάθησης είναι στατιστική. Εδώ, το κριτήριο βάση του οποίου αποφασίζεται σε ποιά κλάση ανήκει κάποιο έγγραφο μαθαίνεται αυτόματα από τα δεδομένα εκπαίδευσης τα οποία πρέπει να ταξινομηθούν από τον χρήστη.

3.10 Αλγόριθμοι Ταξινόμησης

Οι πιο γνωστοί αλγόριθμοι ταξινόμησης είναι:

- Naïve Bayes Algorithm(NB)
- K-Nearest Neighbor Algorithm(kNN)
- Rocchio Algorithm
- Decision Tree Algorithm(DT)
- Artificial Neural Network(ANN)
- Support Vector Machine(SVM)

Πιο κάτω περιγράφονται δυο από τους πιο πάνω αλγόριθμους, ο αλγόριθμος του Bayes και ο αλγόριθμος του k κοντινότερου γείτονα(kNN). Ο αλγόριθμος που υλοποιήθηκε είναι ο αλγόριθμος του Bayes.

3.11 Ο Αλγόριθμος Naïve Bayes

Η βασική ιδέα είναι η εξής: για να υπολογιστούν οι πιθανότητες κάποιας κατηγορίας δεδομένου κάποιου εγγράφου, χρησιμοποιούνται οι από κοινού πιθανότητες(joint probabilities) των λέξεων και των κατηγοριών. Η από κοινού πιθανότητα δυο ενδεχομένων είναι η πιθανότητα τα δυο αυτά ενδεχόμενα να συμβούν ταυτόχρονα. Για κάθε μια από τις πιθανές κατηγορίες ο αλγόριθμος Bayes υπολογίζει την εκ των υστέρων πιθανότητα(posterior probability) ένα έγγραφο να ανήκει σε κάθε μια από αυτές. Το έγγραφο ανατίθεται στην κατηγορία που έχει την μεγαλύτερη εκ των υστέρων πιθανότητα. Για υπολογισμό των εκ των υστέρων πιθανοτήτων, χρησιμοποιείται το θεώρημα του Bayes.

Υπάρχουν δυο εκδοχές του αλγορίθμου Bayes. Η πρώτη ονομάζεται multinomial και η άλλη binomial ή multi-variate Bernoulli. Στην πρώτη λαμβάνεται υπόψη η συχνότητα της κάθε λέξης σε κάποιο έγγραφο, ενώ στην δεύτερη λαμβάνεται υπόψη μόνο η παρουσία ή η απουσία της, δηλαδή δεν μας ενδιαφέρει το πόσες φορές εμφανίζεται.

Ο αλγόριθμος Bayes βασίζεται στην θεωρία πιθανοτήτων και συγκεκριμένα στο θεώρημα του Bayes το οποίο είναι:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

όπου:

$P(h)$ → η πιθανότητα να συμβεί το h **(prior probability)**

$P(D)$ → η πιθανότητα να συμβεί το D

$P(D|h)$ → η πιθανότητα να συμβεί το D δεδομένου του h **(likelihood)**

$P(h|D)$ → η πιθανότητα να συμβεί το h δεδομένου του D **(posterior probability)**

Βάση του πιο πάνω κανόνα, μπορούμε να υπολογίσουμε την μέγιστη a posteriori πιθανότητα των δεδομένων D:

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

$$\equiv \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)} \qquad P(D) \text{ σταθερό => μπορεί να παραλειφθεί}$$

$$\equiv \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h)$$

Υποθέτοντας ότι όλες οι υποθέσεις h είναι το ίδιο πιθανές, δηλαδή $P(h_i) = P(h_j)$, μπορούμε να παραλείψουμε και το $P(h)$:

$$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} \; P(D|h)$$

Η πιο πάνω υπόθεση ονομάζεται **Maximum Likelihood Hypothesis**.

Τα έγγραφα του training set περιγράφονται από συνδέσεις των τιμών των χαρακτηριστικών τους, δηλαδή των λέξεων που τα αποτελούν, με άλλα λόγια $D = \{x_1, x_2, \ldots, x_n\}$. Θέλουμε να το αναθέσουμε σε μια κλάση $c_j \in C$.

$$\begin{aligned} C_{MAP} &\equiv \underset{c_j \in C}{\operatorname{argmax}} P(c_j | x_1, x_2, ..., x_n) \\ &\equiv \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, ..., x_n | c_j) P(c_j)}{P(x_1, x_2, ..., x_n)} \\ &\equiv \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, ..., x_n | c_j) P(c_j) \end{aligned}$$

Λόγω του ότι είναι δύσκολο να υπολογιστεί το $P(x_1, x_2, ..., x_n | c_j)$, θα κάνουμε ακόμη μια υπόθεση ότι τα χαρακτηριστικά είναι στατιστικώς ανεξάρτητα, δηλαδή αν γνωρίζουμε την τιμή ενός χαρακτηριστικού, δεν μπορούμε να πούμε τίποτα για την τιμή κάποιου άλλου χαρακτηριστικού. Αυτό λέγεται **Conditional Independence Assumption**. Έτσι:

$$\begin{aligned} P(x_1, x_2, ..., x_n | c_j) &= P(x_1 | c_j) \cdot P(x_2 | c_j) \cdot ... \cdot P(x_n | c_j) \\ &= \prod_i^n P(x_i | c_j) \end{aligned}$$

Το αποτέλεσμα είναι ο Naïve Bayes Classifier:

$$C_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_i^n P(x_i | c_j)$$

Για να αποφύγουμε να έχουμε αποτέλεσμα 0 σε περιπτώσεις όπου κάποια κλάση δεν περιέχει κάποιο χαρακτηριστικό το οποίο περιέχεται στο έγγραφο που προσπαθούμε να ταξινομήσουμε, δηλαδή $P(x_i | c_j) = 0$, θα ορίσουμε το $P(x_i | c_j)$ ως

$$P(x_i | c_j) = \frac{n_c + 1}{n + k}$$

όπου:
k: αριθμός των τιμών που μπορεί να πάρει το x_i.

Ο αλγόριθμος Bayes είναι ένας από τους πιο απλούς αλγόριθμους ταξινόμησης. Ονομάζεται Naïve Bayes, δηλαδή αφελής επειδή γίνεται η υπόθεση ότι η δεσμευμένη πιθανότητα κάποιας λέξης που εμφανίζεται σε κάποια κατηγορία είναι ανεξάρτητη από τις δεσμευμένες πιθανότητες άλλων λέξεων δοθέντος αυτής της κατηγορίας. Ακόμα και απλός, έχει πολύ δυνατά σημεία. Κάποια από αυτά είναι:

- Είναι εύκολο να κατασκευαστεί
- Είναι γρήγορος
- Εύκολα κατανοητός
- Αν και απλός, είναι πολύ αποτελεσματικός
- Ο αλγόριθμος δουλεύει πολύ καλά στην πράξη, ακόμα και αν κάνει κάποιες υποθέσεις που στην πράξη δεν ισχύουν(π.χ. ότι τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους).

Ένα μειονέκτημα του αλγορίθμου αυτού είναι ότι δεν μπορεί να χρησιμοποιηθεί σε πιο πολύπλοκες περιπτώσεις ταξινόμησης.

3.12 Ο Αλγόριθμος k Κοντινότερου Γείτονα

Ο αλγόριθμος kNN (k nearest neighbor = k κοντινότερου γείτονα) είναι επίσης ένας απλός αλγόριθμος. Είναι ένας instance-based αλγόριθμος μάθησης, δηλαδή η διαδικασία μάθησης αφορά απλά την αποθήκευση των δεδομένων εκπαίδευσης και θεωρείται ένας από τους πιο απλούς αλγόριθμους μάθησης. Τα δεδομένα εκπαίδευσης τυγχάνουν επεξεργασίας όταν εμφανιστεί ένα νέο instance για αυτό και ονομάζεται Lazy Learning. Κάθε φορά που ένα νέο instance πρόκειται να ταξινομηθεί, υπολογίζεται η ομοιότητα του με κάθε ένα από τα αποθηκευμένα δεδομένα εκπαίδευσης.

Ο αλγόριθμος KNN βασίζεται σε μια συνάρτηση απόστασης όπως είναι η Ευκλείδεια απόσταση και η απόσταση συνημίτονου, μεταξύ κάθε εγγράφου εκπαίδευσης και του εγγράφου που πρόκειται να ταξινομηθεί.

Ο αλγόριθμος KNN αναθέτει το έγγραφο που πρόκειται να ταξινομηθεί στην κατηγορία στην οποία ανήκει η πλειοψηφία των k κοντινότερων γειτόνων, όπου το k είναι ένας θετικός ακέραιος, συνήθως μικρός και δίνεται ως παράμετρος. Οι κοντινότεροι γείτονες υπολογίζονται χρησιμοποιώντας κάποια συνάρτηση απόστασης. Τα έγγραφα συνήθως αναπαριστούνται με το vector space μοντέλο, δηλαδή ως διανύσματα όπου κάθε συνιστώσα τους αντιστοιχεί σε κάποιο όρο που εμφανίζεται στο λεξικό, μαζί με κάποιος βάρος για κάθε όρο. Αν ένας όρος δεν εμφανίζεται σε κάποιο συγκεκριμένο έγγραφο, το βάρος του είναι μηδέν. Το βάρος μπορεί να ορισθεί ως:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

Στη συνέχεια υπολογίζεται η ομοιότητα(similarity) κάθε εγγράφου με το έγγραφο το οποίο πρόκειται να ταξινομηθεί. Αυτό είναι δυνατό με την χρήση κάποιας συνάρτησης που υπολογίζει την απόσταση μεταξύ δυο διανυσμάτων. Μερικά παραδείγματα είναι: cosine similarity(ομοιότητα συνημίτονου), extended cosine similarity(Jaccard coefficient), Euclidean distance(ευκλείδεια απόσταση) και Manhattan distance.

Από τους προηγούμενους υπολογισμούς θα εξαχθούν τα k έγγραφα τα οποία έχουν την μεγαλύτερη ομοιότητα με το έγγραφο που πρόκειται να ταξινομηθεί(δηλαδή την μικρότερη απόσταση). Αυτοί ονομάζονται οι κοντινότεροι γείτονες. Τέλος, το έγγραφο θα ανατεθεί στην κλάση στην οποία ανήκουν οι περισσότεροι κοντινότεροι γείτονες.

Κάποιες παραλλαγές στον αλγόριθμο μπορεί να είναι οι εξής: Αντί να λαμβάνεται υπόψη μόνο αν σε ποια κλάση ανήκουν οι k κοντινότεροι γείτονες, είναι καλύτερο να λαμβάνεται υπόψη και η απόσταση που έχουν από το έγγραφο που θα ταξινομηθεί. Με άλλα λόγια αντί το έγγραφο να ανατίθεται στην κλάση στην οποία ανήκουν οι περισσότεροι από τους k κοντινότερους γείτονες του, να ανατίθεται στη κλάση με την μεγαλύτερη ομοιότητα, δηλαδή:

$$w(d_i) = \arg\max_k \sum_{x_j \in kNN} sim(d_i, x_j) I(x_j, c_k)$$

Όπου:

d_i : Το έγγραφο που πρόκειται να ταξινομηθεί

x_j : Κάποιος από του k κοντινότερους γείτονες

c_k : Κάποια από τις κλάσεις

$I(x_j, c_k) = 1$ ανν το x_i ανήκει στην κλάση c_k, 0 διαφορετικά

$sim(d_i, x_j)$: Η συνάρτηση ομοιότητας μεταξύ του d_i και του x_j.

π.χ. μπορεί να είναι η ομοιότητα συνημίτονου.

Το πλεονέκτημα του αλγορίθμου αυτού είναι ότι είναι εύκολο να κατανοηθεί και να υλοποιηθεί. Επίσης, οι Cover και Hart στο [2] έδειξαν ότι το λάθος του kNN περιορίζεται στο μισό του λάθους του Bayes κάτω από συγκεκριμένες υποθέσεις. Ο αλγόριθμος kNN δουλεύει καλά σε περιπτώσεις multi-modal κλάσεων και σε εφαρμογές όπου κάποιο αντικείμενο μπορεί να ανήκει σε περισσότερο από μια κλάση. Επίσης, είναι πολύ αποδοτικός σε περιπτώσεις όπου τα δεδομένα εκπαίδευσης περιέχουν θόρυβο(noisy) και σε περιπτώσεις όπου τα δεδομένα εκπαίδευσης είναι πολλά.

Το σημαντικότερο μειονέκτημα του είναι ότι είναι instance-based αλγόριθμος μάθησης, οπότε δεν γίνεται οποιαδήποτε εκπαίδευση, μέχρι να φτάσει κάποιο έγγραφο για ταξινόμηση και επίσης έχει μεγάλο κόστος υπολογισμού γιατί πρέπει να υπολογίσει την απόσταση κάθε όρου του κειμένου που θα ταξινομηθεί με όλα τα έγγραφα εκπαίδευσης. Ένα άλλο σημαντικό μειονέκτημα είναι ότι χρειάζεται να καθοριστεί κάποια τιμή για το k. Το k παίζει αρκετά σημαντικό ρόλο στην αποδοτικότητα του ταξινομητή και είναι δύσκολο να προσδιοριστεί. Αν είναι πολύ μικρό, το αποτέλεσμα μπορεί να είναι ευαίσθητο σε θορυβώδη δεδομένα. Αν είναι πολύ μεγάλο, το αποτέλεσμα των κοντινότερων γειτόνων μπορεί να περιέχει πολλά έγγραφα από άλλες κατηγορίες. Το k μπορεί να οριστεί χρησιμοποιώντας διάφορες τεχνικές οι οποίες χρησιμοποιούν ευρετικά. Τέλος, πρέπει να καθοριστεί πια συνάρτηση απόστασης πρέπει να εφαρμοστεί για να προκύψουν τα καλύτερα αποτελέσματα.

Κεφάλαιο 4

Υλοποίηση Συστήματος

4.1 Περιγραφή Υλοποίησης

Αυτή η διπλωματική εργασία αφορά την υλοποίηση ενός αυτόματου φίλτρου ταξινόμησης ηλεκτρονικού ταχυδρομείου. Το σύστημα χρησιμοποιεί Τεχνικές Ανάκτησης Πληροφοριών για να ταξινομεί με αυτόματο τρόπο τα εισερχόμενα ηλεκτρονικά μηνύματα στους κατάλληλους φακέλους (mailboxes) που διαθέτει η χρήστης, ανάλογα με το περιεχόμενο τους. Πιο συγκεκριμένα, το υποσύστημα αναπτύχθηκε σαν extension(πρόσθετο) στον Mozilla Thunderbird και υλοποιεί κατάλληλους αλγόριθμους ταξινόμησης από τον κλάδο ανάκτησης πληροφοριών και επίσης χρησιμοποιεί πολλές τεχνικές από αυτό τον κλάδο. Ο Thunderbird είναι μια εφαρμογή διαχείρισης ηλεκτρονικού ταχυδρομείου ανοικτού κώδικα.

Το σύστημα που αναπτύχθηκε έκανε χρήση του αλγορίθμου Bayes για την αυτόματη ταξινόμηση των ηλεκτρονικών μηνυμάτων. Η διαδικασία είναι η εξής: αρχικά ένα μήνυμα καταφθάνει στο ηλεκτρονικό ταχυδρομείο του χρήστη. Το πρόσθετο χρησιμοποιεί κάποιο Folder Listener για να είναι σε θέση να γνωρίζει πότε ένα νέο μήνυμα προσθέτεται στα εισερχόμενα μηνύματα του χρήστη. Την δυνατότητα αυτή την προσφέρει το XPCOM(Cross-platform Component Object Model) μέσω του JavaScript και για αυτό όταν έρθει ένα νέο μήνυμα, μπορεί να το διαχειριστεί. Τα ηλεκτρονικά μηνύματα στον Thunderbird είναι αποθηκευμένα σε mbox μορφή. Δηλαδή αρχικά φυλάγονται τα headers τα οποία περιέχουν διάφορες πληροφορίες για το μήνυμα(π.χ. subject, ημερομηνία, αποστολέας κ.τ.λ.) και στη συνέχεια ακολουθούν τα attachments

και το body του μηνύματος. Από τα πιο πάνω μας ενδιαφέρουν μόνο τα πεδία From, Subject και Body.

Για να πάρω το From, το Subject και το Body του μηνύματος χρησιμοποίησα κώδικα από ένα έτοιμο λογισμικό το οποίο μετατρέπει ένα αρχείο από mbox μορφή σε eml μορφή. Το λογισμικό αυτό ονομάζεται mbox2eml. Χρησιμοποίησα μόνο ένα μέρος του κώδικα του πιο πάνω λογισμικού για να μπορώ να εξάγω από κάθε ηλεκτρονικό μήνυμα το From, το Subject κα το Body τους.

Αφού εξαχθούν τα πιο πάνω, πρέπει να γίνει μια επεξεργασία η οποία αφορά τεχνικές από τον κλάδο ανάκτησης πληροφοριών. Πιο συγκεκριμένα, πρέπει να γίνει το tokenization, η κανονικοποίηση των token, το stemming(στελέχωση) και επίσης πρέπει να φύγουν τα stop words(λέξεις αποκλεισμού). Τα πιο πάνω πρέπει να εφαρμοστούν σε όλα τα μηνύματα τα οποία ανήκουν στις κλάσεις, δηλαδή φακέλους, που λαμβάνουν μέρος στην ταξινόμηση.

Κατά την κανονικοποίηση, όλα τα γράμματα των tokens μετατράπηκαν σε μικρά, έφυγαν τα stop words και έγινε stemming των token.

Η λίστα με τα stop words που χρησιμοποιήθηκε βρίσκεται στην ιστοσελίδα http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words αλλά με μερικές εξαιρέσεις. Η λίστα αυτή περιέχει λέξεις οι οποίες είτε είναι πολύ κοινές(π.χ. “and” κτλ), είτε ασήμαντες(π.χ. άρθρα).

Ο αλγόριθμος που χρησιμοποιήθηκε για το stemming είναι ο Porter Stemmer. Ο Martin Porter, αυτός που έγραψε τον αλγόριθμο, έκδωσε υλοποιήσεις του αλγορίθμου σε πολλές γλώσσες προγραμματισμού οι οποίες είναι διαθέσιμες δωρεάν και βρίσκονται στην επίσημη ιστοσελίδα http://tartarus.org/~martin/PorterStemmer.

Οι όροι κάθε φακέλου αποτελούν το λεξικό(vocabulary). Αρχικά τοποθετούνται σε μια δομή όπου για κάθε μήνυμα υπάρχει μια λίστα με τους όρους που εμφανίζονται σε αυτό. Για να είναι πιο εύκολη η διαδικασία της ταξινόμησης, οι όροι πρέπει να ευρετηριαστούν. Για την ευρετηρίαση(indexing) χρησιμοποίησα μια τροποποιημένη

μορφή του inverted index, όπου για κάθε όρο υπάρχει μια λίστα με όλες τις κλάσεις που τον περιέχουν.

Όπως αναφέρθηκε πιο πάνω, ο αλγόριθμος που αναπτύχθηκε είναι ο αλγόριθμος Bayes. Αρχικά υπολογίζεται για κάθε κλάση C η prior probability της, P(C), βάση του πιο κάτω τύπου:

$$P(C) = \text{αριθμός μηνυμάτων στο folder C / συνολικός αριθμός μηνυμάτων.}$$

Στη συνέχεια, υπολογίζονται οι Conditional Probabilities P(t1,t2,...,tn|C) για κάθε όρο του λεξιλογίου:

$$P(t_1,t_2,...,t_n|C) = P(t_1|C)*P(t_2|C)*...*P(t_n|C)$$

όπου

$$P(t_i|C) = (\text{αριθμός εμφανίσεων του όρου } t_i \text{ στα έγγραφα εκπαίδευσης για την κλάση C} + 1)/(\text{αριθμός tokens στην κλάση C} + B),$$

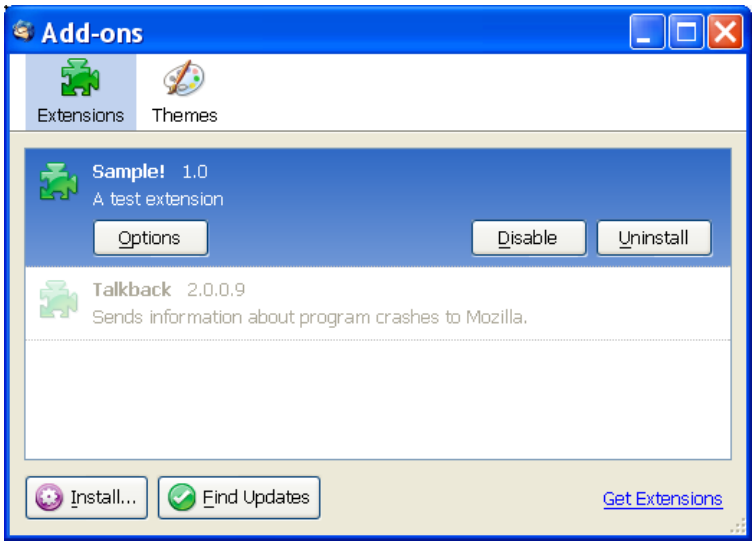
όπου
B = |V| = αριθμός όρων στο λεξιλόγιο.

Η τελική πιθανότητα $P(C| t_1,t_2,...,t_n)$ το έγγραφο d να ανήκει στην κλάση C είναι:

$$P(C| d) = P(C) * P(t_1,t_2,...,t_n|C)$$

Επειδή πολλαπλασιάζονται πολλές πιθανότητες, μια για κάθε όρο t_i, υπάρχει πιθανότητα να προκύψει floating point underflow. Για να το αποφύγουμε αυτό, αντί να πολλαπλασιάζονται οι πιθανότητες είναι καλύτερα να προσθέτονται οι αλγόριθμοι των πιθανοτήτων. Η κλάση με την μεγαλύτερη log πιθανότητα πάλι είναι η πιο πιθανή κλάση αφού ισχύει log(x*y) = log(x) + log(y) και ο λογάριθμος είναι μονοτονική συνάρτηση.

Το extension προσφέρει την δυνατότητα στον χρήστη να επιλέξει ποιοι από τους φακέλους που διαθέτει επιθυμεί να λαμβάνουν μέρος στην ταξινόμηση. Για να κάνει την επιλογή αυτή πρέπει να αλλάξει τις ρυθμίσεις του extension. Οι ρυθμίσεις βρίσκονται κάτω από το Tools, Add-ons(Σχήμα 8).



Σχήμα 8: Εγκατάσταση Πρόσθετου Στον Thunderbird. Το κουμπί Options χρησιμοποιείται για πρόσβαση στις ρυθμίσεις του προσθέτου.

Ο χρήστης έχει την δυνατότητα να καθορίσει τα βάρη που επιθυμεί για τα πεδία From, Subject και Body. Ο χρήστης, ανάλογα με τον τρόπο που εκείνος θεωρεί πιο ορθό μπορεί να καθορίσει για κάθε ένα από τα τρία πεδία κατά πόσον θα λαμβάνεται υπόψη κατά την ταξινόμηση και επίσης πόσο βάρος θα του δοθεί σε σχέση με τα άλλα. Για παράδειγμα αν κάποιος χρήστης θεωρεί πιο σημαντικό το πεδίο From από τα άλλα δυο πεδία, μπορεί να του δώσει περισσότερο βάρος. Αυτό θα έχει καλύτερα αποτελέσματα στην περίπτωση όπου τα θέματα των μηνυμάτων κάποιου χρήστη έχουν άμεση σχέση με τον αποστολέα τους. Στο Σχήμα 10-1 φαίνεται ένα screenshot από τις ρυθμίσεις αυτές.

Επίσης, μια άλλη δυνατότητα που δίνεται στον χρήστη είναι να επιλέξει ο ίδιος αν θα γίνει ή όχι χρήση του Stemmer(Σχήμα 10-2).

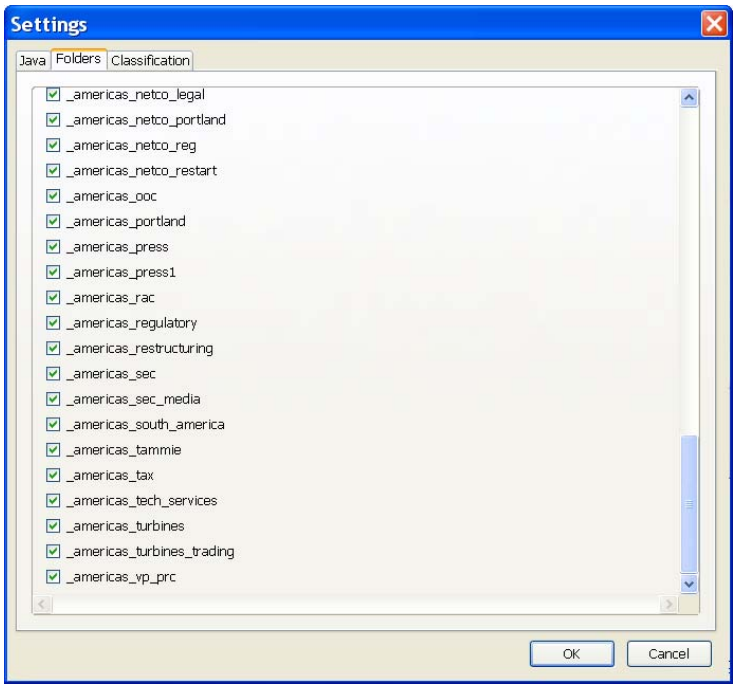
Η λειτουργικότητα του extension στον Thunderbird καθορίζεται από το JavaScript μέρος του. Το πρόσθετο που υλοποίησα τρέχει μέσω του JavaScript ένα jar αρχείο χρησιμοποιώντας interfaces του Mozilla XPCOM και συγκεκριμένα το `nsIProcess` interface το οποίο αναπαριστά μια εκτελέσιμη διαδικασία. Το jar αρχείο περιέχει κλάσεις σε java. Ο κώδικας java υλοποιεί τον αλγόριθμο που χρησιμοποιήθηκε για την ταξινόμηση καθώς επίσης και τις διάφορες επεξεργασίες που γίνονται στα μηνύματα όπως το tokenization, το stemming, τον αποκλεισμό των stop words, το indexing και κάποια άλλα. Επίσης, αυτός ο κώδικας διαβάζει τα αρχεία με τα μηνύματα και εξάγει τα από αυτά τα στοιχεία που είναι σημαντικά για την ταξινόμηση, δηλαδή τα πεδία From, Subject και Body.

Η επικοινωνία του Java μέρους με του JavaScript μέρους γίνεται με τον εξής τρόπο: Αρχικά το JavaScript μέρος όταν καλέσει το Java μέρος του δίνει ως παράμετρο τα paths των φακέλων που λαμβάνουν μέρος στην ταξινόμηση ώστε να μπορεί να διαβάσει τα mbox αρχεία με τα μηνύματα. Επίσης, τα μηνύματα που πρόκειται να ταξινομηθούν φυλάγονται από το JavaScript μέρος σε ένα αρχείο και το path του αρχείου αυτού δίνεται και αυτό στο Java μέρος σαν παράμετρος. Δηλαδή η μορφή του μοιάζει με ένα mbox αρχείο που περιέχει τα μηνύματα που θα ταξινομηθούν. Εκτός από τους πιο πάνω παραμέτρους, το JavaScript μέρος περνά στο Java μέρος κάποιους επιπλέον παραμέτρους. Αυτοί είναι: true ή false ανάλογα με το αν θα γίνει χρήση του Stemmer ή όχι, το βάρος που θα δοθεί στο κάθε πεδίο From, Subject και Body και το path του extension directory του χρήστη. Το extension directory είναι ο χώρος όπου είναι εγκατεστημένο το πρόσθετο και βρίσκεται μέσα στο profile folder του χρήστη. Αν ο χρήστης επιλέξει περισσότερα από ένα μηνύματα για ταξινόμηση, ο αλγόριθμος θα κάνει το training μόνο μια φορά και για κάθε ένα από τα μηνύματα θα υπολογίσει τις τελικές πιθανότητες για κάθε ένα από τους καταλόγους που διαθέτει ο χρήστης. Στη συνέχεια, οι πιθανότητες όλων των φακέλων συγκρίνονται μεταξύ τους και ο φάκελος με την πιο μεγάλη πιθανότητα είναι ο φάκελος στον οποίο θα μεταφερθεί το μήνυμα. Τα αποτελέσματα καταγράφονται σε ένα αρχείο το οποίο βρίσκεται στο extension directory του χρήστη. Τα αποτελέσματα έχουν την ακόλουθη μορφή:

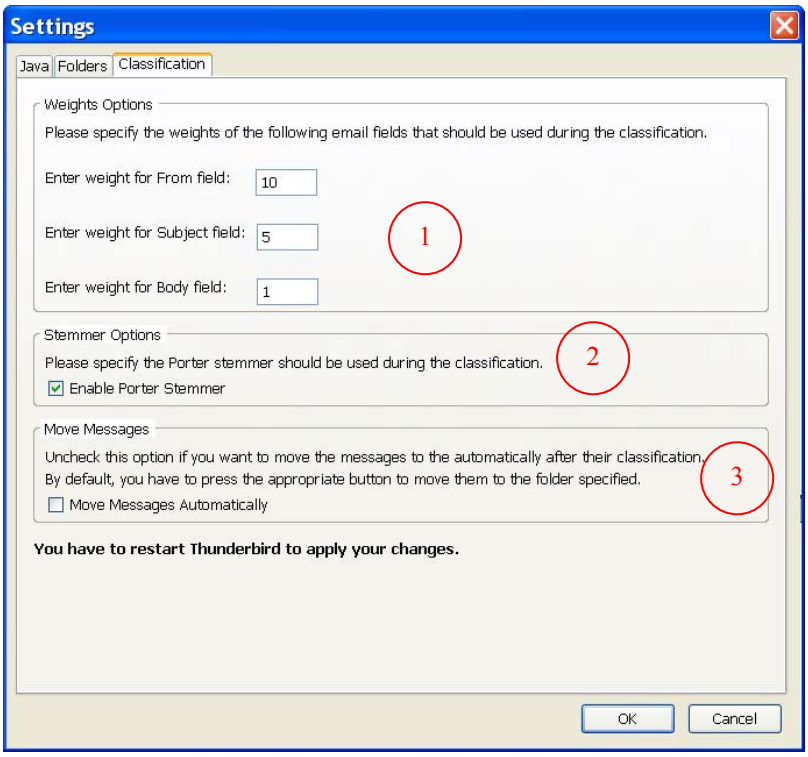
<Message-ID> destinationDirectoryName

Το Message-ID είναι ένα πεδίο του header του μηνύματος, το οποίο χαρακτηρίζει μοναδικά το συγκεκριμένο μήνυμα. Το destinationDirectoryName είναι το όνομα του

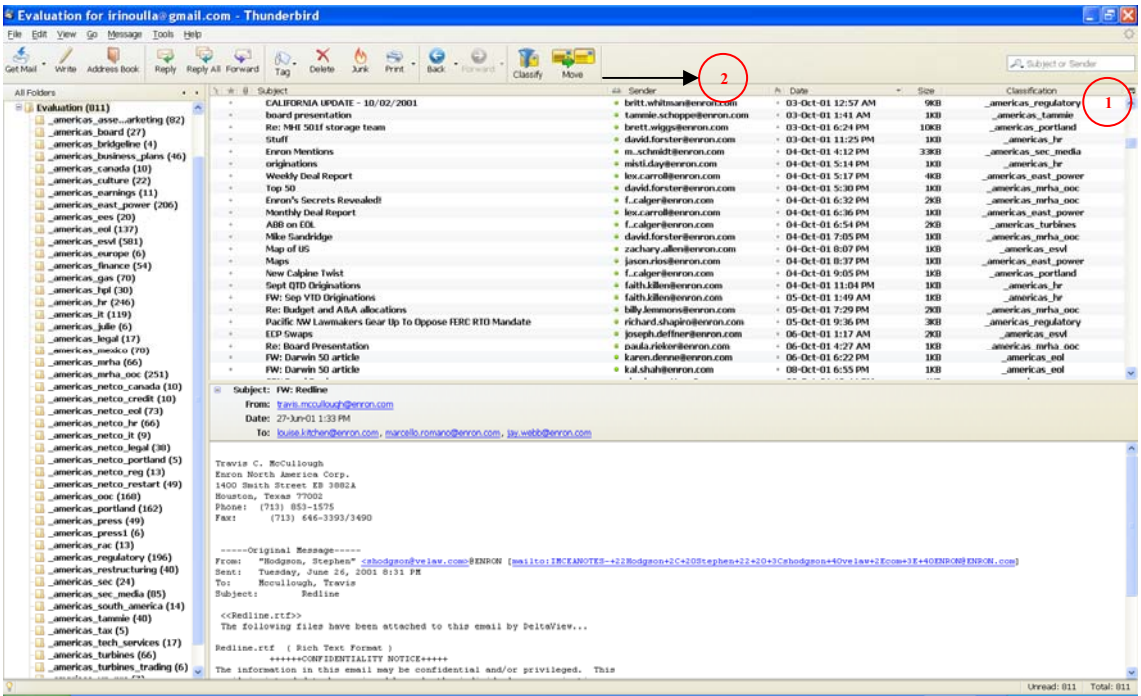
καταλόγου στο οποίο αποφασίστηκε από τον αλγόριθμο ότι ανήκει το μήνυμα. Το JavaScript μέρος με την σειρά του διαβάζει το αρχείο με το αποτέλεσμα και κάνει την μετακίνηση του μηνύματος χρησιμοποιώντας και πάλι κάποιο XPCOM interface. Οι αυτόματες μετακινήσεις γίνονται μόνο στην περίπτωση που ο χρήστης το επιλέξει από τις ρυθμίσεις του προσθέτου (Σχήμα 10-3). Στην περίπτωση που ο χρήστης δεν επιθυμεί το πιο πάνω να γίνεται αυτόματα, τα αποτελέσματα της ταξινόμησης εμφανίζονται σε μια στήλη στο κύριο παράθυρο του Thunderbird (Σχήμα 11-1). Αν ο χρήστης αποφασίσει στη συνέχεια ότι θέλει να γίνουν οι μετακινήσεις, μπορεί να επιλέξει τα μηνύματα και να πατήσεις το κουμπί Move (Σχήμα 11-2) το οποίο θα μεταφέρει τα μηνύματα αυτά στους καταλόγους στους οποίους αποφάσισε το Java μέρος και που φαίνονται στην στήλη “Classification”.



Σχήμα 9: Οι ρυθμίσεις του προσθέτου όπου ο χρήστης μπορεί να επιλέξει ποιοι από τους φακέλους θα λαμβάνονται υπόψη κατά την ταξινόμηση.



Σχήμα 10: Οι ρυθμίσεις του προσθέτου για την ταξινόμηση. Ο χρήστης μπορεί να επιλέξει πόσο βάρος θα δοθεί σε κάθε ένα από τα τρία πεδία(1) και επίσης έχει την επιλογή να διαλέξει αν θέλει να γίνει χρήση του Stemmer ή όχι(2). Τέλος, μπορεί να επιλέξει αν μετά την ταξινόμηση τα μηνύματα θα μετακινούνται αυτόματα στους κατάλληλους καταλόγους ή όχι(3).



Σχήμα 11: Ένα screenshot όπου παρουσιάζεται η στήλη με τα αποτελέσματα της ταξινόμησης.

Κεφάλαιο 5

Αποτίμηση Συστήματος

5.1 Enron Corpus

Για την αποτίμηση του συστήματος που αναπτύχθηκε χρησιμοποιήθηκε ένα μέρος μιας συλλογής από πραγματικά ηλεκτρονικά μηνύματα γνωστή ως Enron corpus ή Enron Dataset. Το Enron Email corpus είναι μια συλλογή από πολλές χιλιάδες ηλεκτρονικά μηνύματα, η οποία δημοσιεύτηκε κατά την διάρκεια της νομικής έρευνας που αφορά την ενεργειακή εταιρία Enron λόγω κάποιου οικονομικού σκανδάλου. Το ανεπεξέργαστο υλικό βρίσκεται στην σελίδα http://www.cs.cmu.edu/~enron/ από τον William Cohen. Αυτή η συλλογή από μηνύματα είναι σημαντική γιατί είναι η μοναδική αληθινή και μεγάλη συλλογή που δημοσιεύτηκε. Αποτελείται από 517,431 μηνύματα τα οποία ανήκουν σε 150 χρήστες.

Από κάποια ανάλυση που έγινε στα ηλεκτρονικά μηνύματα του Enron corpus[9], φαίνεται ότι το Enron corpus περιέχει μηνύματα από χρήστες με διαφορετικό αριθμό μηνυμάτων. Επίσης, φαίνεται ότι πολλοί χρήστες χρησιμοποιούν φακέλους για να ταξινομούν τα μηνύματα τους.

Όπως αναφέρθηκε και πιο πάνω, για την αποτίμηση(evaluation) του συστήματος χρησιμοποιήθηκε ένα μέρος του Enron corpus. Συγκεκριμένα χρησιμοποιήθηκαν

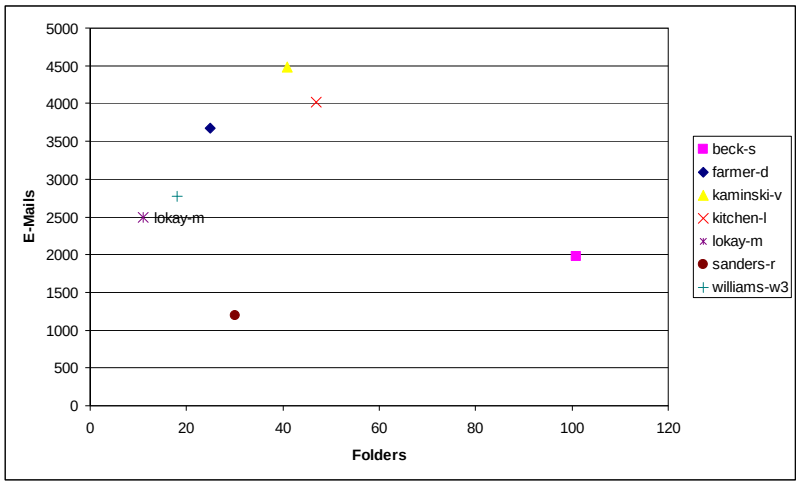
συνολικά 5597 ηλεκτρονικά μηνύματα από 7 χρήστες οι οποίοι έχουν φακέλους με αρκετά ηλεκτρονικά μηνύματα. Οι χρήστες αυτοί είναι: Sally Beck (Chief Operating Officer), Darren Farmer (Logistics Manager), Vincent Kaminski (Head of Quantitative Modeling Group), Louise Kitchen (President of EnronOnline), Michelle Lokay (Administrative Assistant), Richard Sanders (Assistant General Counsel) και William Williams III (Senior Analyst). Από αυτούς τους χρήστες αφαιρέθηκαν οι φάκελοι που δεν έχουν κάποιο συγκεκριμένο θέμα, δηλαδή αφαιρέθηκαν οι φακέλοι: all_documents, calendar, contacts, deleted_items, discussion_threads, inbox, notes_inbox, sent, sent_items και _sent_mail. Αφαιρέθηκαν επίσης οι φάκελοι που περιέχουν λιγότερα από τρία μηνύματα.

Στον πιο κάτω πίνακα(Πίνακας 3) φαίνεται ο αριθμός των φακέλων και ο αριθμός των μηνυμάτων για κάθε χρήστη:

User	Number of folders	Number of messages
beck-s	101	1971
farmer-d	25	3672
kaminski-v	41	4477
kitchen-l	47	4015
lokay-m	11	2489
sanders-r	30	1188
williams-w3	18	2769

Πίνακας 3: Αριθμός Μηνυμάτων και Φακέλων για κάθε ένα από τους χρήστες.

Στο Σχήμα 12 φαίνεται για κάθε χρήστη πόσους φακέλους και πόσα μηνύματα διαθέτει.



Σχήμα 12: Γραφική Παράσταση που παρουσιάζει πόσους φακέλους και πόσα μηνύματα διαθέτει κάθε χρήστης.

5.2 Περιγραφή Μεθοδολογίας Πειραμάτων

Για σκοπούς αξιολόγησης του συστήματος, τα μηνύματα των εφτά χρηστών χρειάστηκε να μετατραπούν στην μορφή mbox την οποία χρησιμοποιεί ο Thunderbird για να αποθηκεύει τα ηλεκτρονικά μηνύματα. Η περιγραφή της μορφής αυτής βρίσκεται στο κεφάλαιο 2.4. Ο λόγος που χρειάστηκε να γίνει αυτή η μετατροπή είναι γιατί κάθε μήνυμα της συλλογής αυτής βρίσκεται σε ένα ξεχωριστό αρχείο. Κάθε αρχείο mbox περιέχει τα μηνύματα κάποιου καταλόγου. Άρα τα αρχεία που περιέχουν τα μηνύματα κάποιου καταλόγου χρειάστηκε να συνενωθούν. Για την δημιουργία τον αρχείων αυτών υλοποιήθηκε ένα πρόγραμμα το οποίο διαβάζει τα αρχεία με τα μηνύματα και για κάθε κατάλογο δημιουργεί ένα mbox αρχείο.

Τα μηνύματα κάθε καταλόγου χωρίστηκαν σε δυο κατηγορίες. Η πρώτη περιέχει τα μηνύματα που χρησιμοποιήθηκαν για την εκπαίδευση του ταξινομητή(training) και η δεύτερη τα μηνύματα που χρησιμοποιήθηκαν για τον έλεγχο του(testing). Από κάθε κατάλογο, το 80% των μηνυμάτων του χρησιμοποιήθηκαν για training και το υπόλοιπο 20% testing. Για να ελεγχθεί καλύτερα η ποιότητα του ταξινομητή, ο διαχωρισμός αυτός έγινε με 10 τυχαίους διαφορετικούς τρόπους. Τα αποτελέσματα που φαίνονται πιο κάτω είναι ο μέσος όρος των μετρήσεων που προέκυψαν.

Για να ελεγχθεί πως τα βάρη επηρεάζουν την ορθότητα του ταξινομητή, χρησιμοποιήθηκαν 8 διαφορετικοί συνδυασμοί βαρών για το κάθε πεδίο. Αυτοί είναι:

Configuration	From	Subject	Body
0 0 1	0	0	1
0 1 0	0	1	0
1 0 0	1	0	0
0 1 1	0	1	1
1 0 1	1	0	1
1 1 0	1	1	0
1 1 1	1	1	1
10 5 1	10	5	1

Πίνακας 4: Τα βάρη που χρησιμοποιήθηκαν στα πειράματα. Η στήλη configuration αναγράφει πώς είναι κωδικοποιημένα τα βάρη στις γραφικές παραστάσεις.

Κάθε γραμμή είναι ένας συνδυασμός των βαρών για κάθε πεδίο που χρησιμοποιήθηκε κατά την διεξαγωγή των πειραμάτων. Κάθε μια από τις τρεις πρώτες γραμμές του πίνακα λαμβάνει υπόψη μόνο ένα από τα τρία πεδία From, Subject και Body. Δηλαδή ο πρώτος συνδυασμός λαμβάνει υπόψη μόνο το πεδίο From, ο δεύτερος λαμβάνει υπόψη μόνο το πεδίο Subject και ο τρίτος μόνο το πεδίο Body.

Οι επόμενες τρεις γραμμές λαμβάνουν υπόψη δυο από τα τρία πεδία. Δηλαδή η $4^{η}$ γραμμή του πίνακα λαμβάνει υπόψη τα πεδία Subject και Body, η $5^{η}$ γραμμή τα πεδία From και Body και η $6^{η}$ γραμμή τα πεδία From και Subject. Όπου υπάρχει μηδενικό κάτω από μια στήλη κάποιου πεδίου σημάνει πως το πεδίο αυτό δεν λαμβάνεται καθόλου υπόψη κατά την διαδικασία της ταξινόμησης.

Η επόμενη γραμμή(γραμμή 7) δίνει ίσο βάρος και στα τρία πεδία. Με άλλα λόγια, τα τρία πεδία συνενώνονται σε ένα. Για παράδειγμα αν ένας όρος εμφανίζεται και στο πεδίο Body και στο πεδίο Subject του μηνύματος, θεωρείται ότι εμφανίστηκε 2 φορές.

Στο τελευταίο πείραμα(γραμμή 8), τα βάρη που δόθηκαν σε κάθε πεδίο είναι ξεχωριστά. Το μικρότερο βάρος δόθηκε στο πεδίο Body, μετά ακολούθησε το πεδίο Subject και το περισσότερο βάρος δόθηκε στο πεδίο From. Η λογική με την οποία αποφασίστηκε αυτός ο συνδυασμός βαρών, είναι η ακόλουθη: Λόγω του ότι πολλές φορές ο αποστολέας έχει άμεση σχέση με το θέμα του μηνύματος αλλά αποτελείται από μόνο ένα όρο, δώσαμε το περισσότερο βάρος σε αυτόν τον όρο ώστε να ληφθεί υπόψη δραστικά. Κατ' ακρίβεια λαμβάνεται υπόψη σαν να υπήρχε 2 φορές στο πεδίο Subject.

Το πεδίο Subject περιέχει πολλές λέξεις κλειδιά και συνήθως είναι μικρότερο από το πεδίο Body. Έτσι, του δόθηκε περισσότερη έμφαση από αυτό. Τέλος, στο πεδίο Body δόθηκε η λιγότερη έμφαση γιατί περιέχει πολλούς όρους σε σχέση με τα άλλα δυο πεδία.

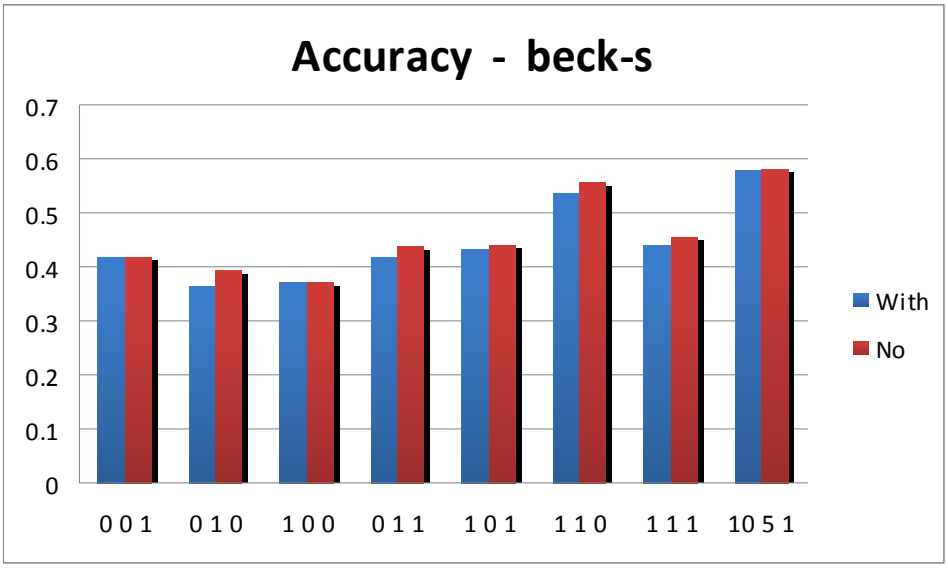
Τα αποτελέσματα ελέγχθηκαν ως εξής: δημιουργήθηκε ένα πρόγραμμα το οποίο σαν είσοδο παίρνει δυο αρχεία. Το πρώτο περιέχει τα αποτελέσματα του ταξινομητή για κάθε μήνυμα, και το δεύτερο περιέχει που πραγματικά ανήκει. Η μορφή και των δυο αρχείων είναι η ίδια με αυτήν που περιγράφηκε στο 4.1, δηλαδή:
<Message-ID> destinationDirectoryName
Το πρόγραμμα αυτό διαβάζει τα δυο αρχεία, τα συγκρίνει και επεξεργάζεται τα αποτελέσματα. Σαν έξοδο βγάζει ένα αρχείο το οποίο περιέχει τα διάφορα μετρικά(accuracy, precision, recall).

5.3 Χρήση Βαρών

Τα ηλεκτρονικά μηνύματα δεν αποτελούνται μόνο από ένα πεδίο, αλλά περιέχουν διάφορα πεδία τα οποία μπορεί να επηρεάζουν το που μπορεί να ανήκουν. Τα πεδία αυτά είναι το Subject, το Body, το πεδίο To, το πεδίο From, το πεδίο CC κ.ά. Κάθε ένα από αυτά τα πεδία είναι σημαντικό στοιχείο στον ταξινομητή και έχει ξεχωριστή σημασία από τα άλλα. Το πόση σημασία πρέπει να δίνεται στο κάθε πεδίο εξαρτάται από την τακτική ταξινόμησης του κάθε χρήστη. Για παράδειγμα κάποιος χρήστης μπορεί να έχει καταλόγους που περιέχουν μηνύματα από συγκεκριμένους αποστολείς ενώ κάποιος άλλος να έχει μηνύματα από κάποιο αποστολέα που να ανήκουν σε διαφορετικούς φακέλους. Στην πρώτη περίπτωση πρέπει να δοθεί πάρα πολλή έμφαση στο πεδίο From ενώ στην δεύτερη ελάχιστη.

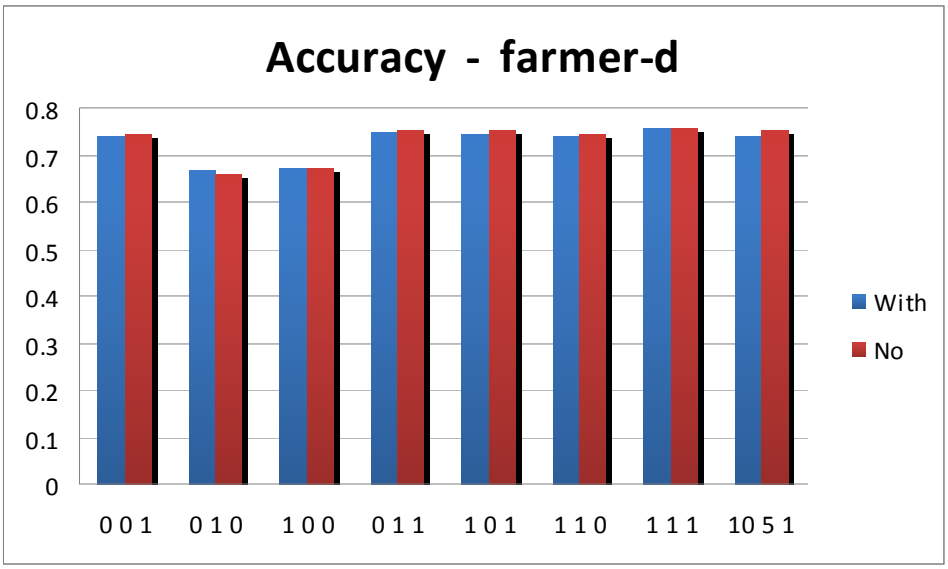
Έτσι κάθε χρήστης πρέπει να είναι ικανός να κρίνει ανάλογα με την τεχνική ταξινόμησης που χρησιμοποιεί, σε ποια από τα πιο πάνω πεδία θα ήθελε να δοθεί περισσότερη έμφαση και σε ποια λιγότερη. Το σύστημα που αναπτύχθηκε δίνει αυτή την δυνατότητα στους χρήστες του, επιτρέποντας του να καθορίσει ο ίδιο πόσο βάρος θέλει να δοθεί σε κάθε πεδίο. Τα πεδία που χρησιμοποιήθηκαν είναι το Subject, το Body, και το πεδίο From.

Στο Σχήμα 13 φαίνεται η ορθότητα του συστήματος για τον χρήστη beck-s, για κάθε ένα από τους πιο πάνω συνδυασμούς βαρών.



Σχήμα 13: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη beck-s.

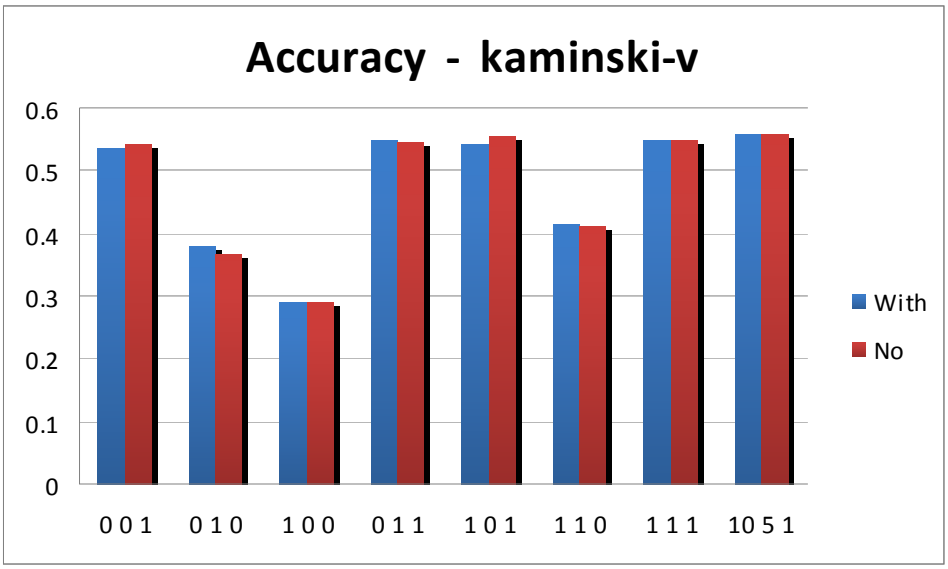
Όπως φαίνεται από την πιο πάνω γραφική παράσταση, η ορθότητα ήταν ψηλή όταν έγινε χρήση των βαρών 10 για From, 5 για Subject και 1 για Body (0.582278). Ακολουθεί ο συνδυασμός που λαμβάνει υπόψη ισότιμα τα πεδία From και Subject και δεν λαμβάνει καθόλου υπόψη το πεδίο Body(0.556962). Η χαμηλότερη ορθότητα ήταν όταν δόθηκε έμφαση μόνο στο πεδίο Subject(0.364557) με τη χρήση του Stemmer.



Σχήμα 14: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη farmer-d.

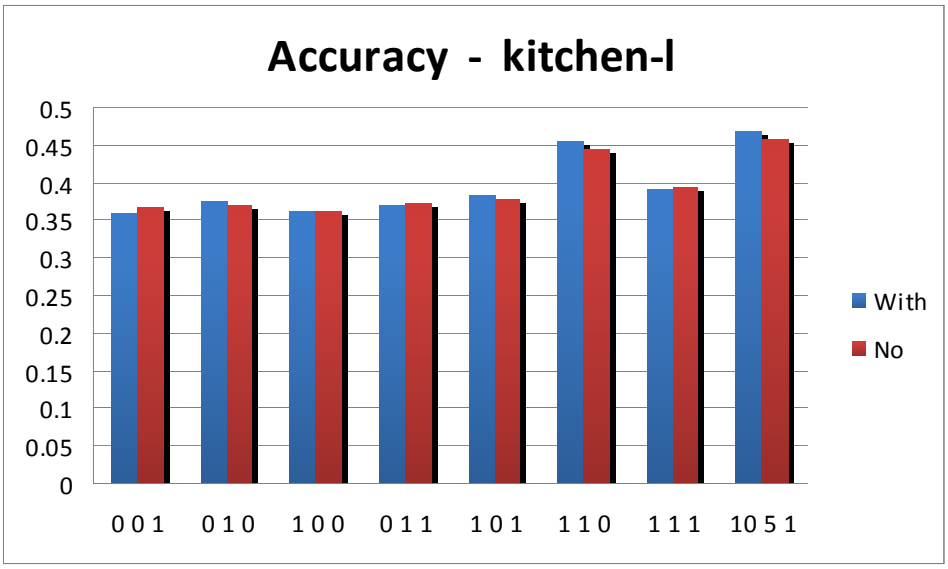
Στο Σχήμα 14 φαίνεται πως τα βάρη επηρεάζουν τον χρήστη farmer-d. Όπως βλέπουμε από την γραφική παράσταση, η μεγαλύτερη ορθότητα(0.757493) επιτυγχάνεται όταν δοθούν ίσα βάρη και στα τρία πεδία(στήλη 1 1 1), αλλά με ελάχιστη διαφορά από τις άλλες περιπτώσεις. Η μικρότερη ορθότητα(0.659401) προέκυψε όταν χρησιμοποιήθηκε μόνο το πεδίο Subject(στήλη 0 1 0), χωρίς να γίνει χρήση του Stemmer. Από αυτή την γραφική παράσταση δεν μπορούμε να διακρίνουμε ποιος συνδυασμός βαρών είναι πιο αποδοτικός για τον συγκεκριμένο χρήστη αφού η διαφορά στην ορθότητα του ταξινομητή για κάθε συνδυασμό βαρών είναι σχεδόν μηδενική.

Ακολουθεί η γραφική παράσταση για τον χρήστη kaminski-v (Σχήμα 15).



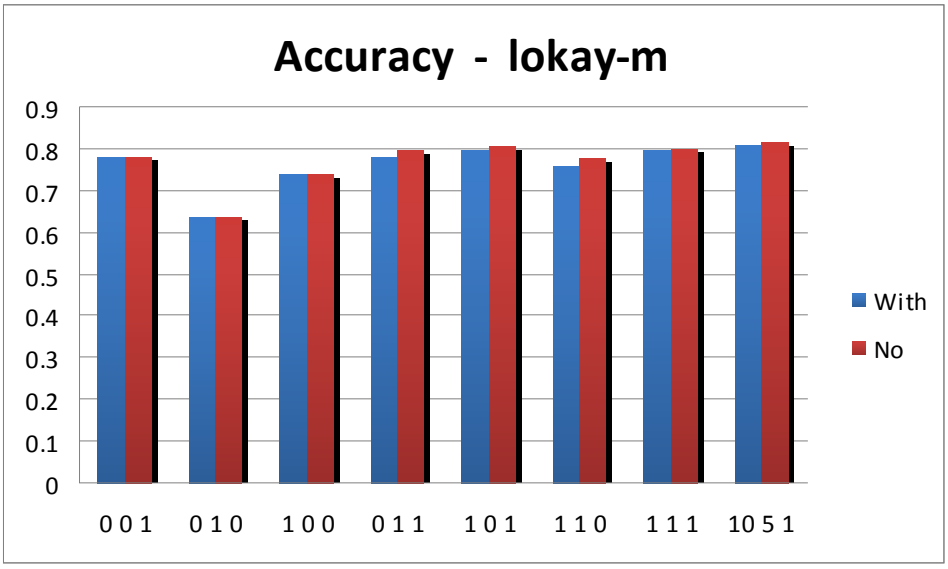
Σχήμα 15: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη kaminski-v.

Ο χρήστης αυτός παρουσίασε αρκετά χαμηλή ορθότητα(0.289385) όταν χρησιμοποιήθηκαν τα βάρη 1 0 0, δηλαδή όταν λήφθηκε υπόψη μόνο το πεδίο From. Η μεγαλύτερη ορθότητα(0.558659) παρατηρήθηκε στην τελευταία στήλη(10 5 1) αλλά με πολύ μικρή διαφορά από τις στήλες 0 0 1, 0 1 1, 1 0 1 και 1 1 1. Ούτε από αυτή τη γραφική παράσταση δεν μπορούμε να διακρίνουμε ποιος συνδυασμός βαρών είναι πιο αποδοτικός για τον συγκεκριμένο χρήστη για τον ίδιο λόγο όπως πιο πάνω.



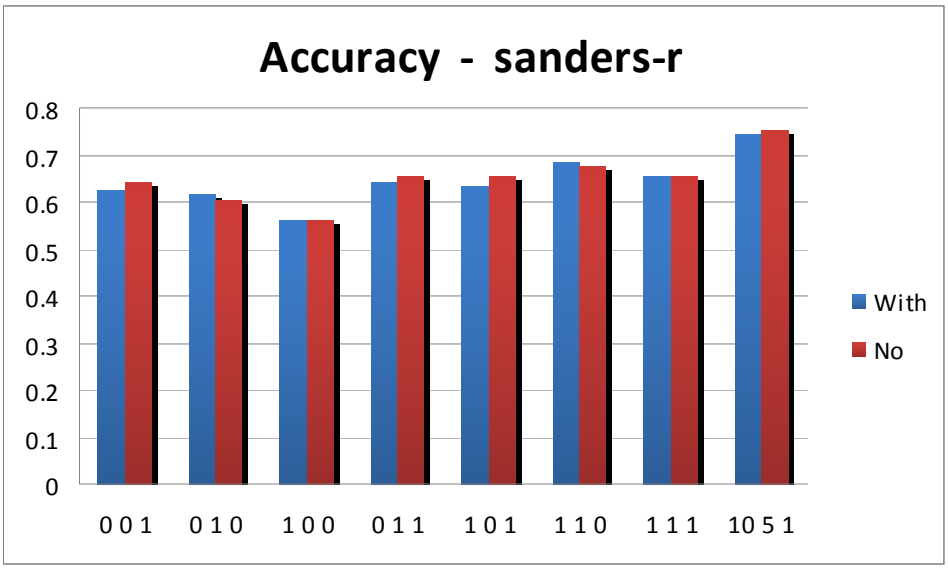
Σχήμα 16: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη kitchen-l.

Ο χρήστης kitchen-l (Σχήμα 16) παρουσίασε ιδιαίτερα ψηλή ορθότητα με τα βάρη 10 5 1 (0.468085) και 1 1 0 (0.454318) και την πιο χαμηλή (0.357947) με τα βάρη 0 0 1 όταν έγινε χρήση του Stemmer.



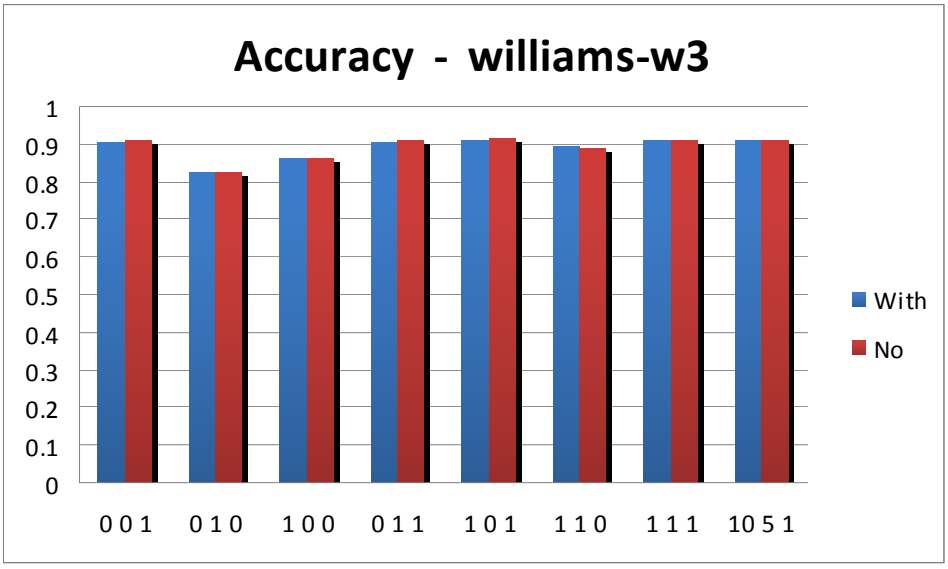
Σχήμα 17: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη lokay-m.

Η γραφική παράσταση για τον χρήστη lokay-m φαίνεται στο Σχήμα 17. Η ψηλότερη ορθότητα (0.812877) εμφανίζεται και πάλι στην στήλη 10 5 1 αλλά πάλι δεν έχει ιδιαίτερη διαφορά από τις περισσότερες από τις άλλες στήλες. Η χαμηλότερη ορθότητα εμφανίζεται στη στήλη 0 1 0, δηλαδή όταν δοθεί έμφαση μόνο στο πεδίο Subject(0.635815).



Σχήμα 18: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη sanders-r.

Ο χρήστης sanders-r (Σχήμα 18) είχε αρκετά ψηλή ορθότητα (0.753138) με τον συνδυασμό 10 5 1. Η χαμηλότερη προέκυψε με τα βάρη 1 0 0 και είναι 0.560669.



Σχήμα 19: Για κάθε βάρος φαίνεται η ορθότητα του ταξινομητή με και χωρίς Stemmer για τον χρήστη williams-w3.

Τέλος, ο χρήστης williams-w3 (Σχήμα 19) παρουσιάζει την πιο ψηλή του ορθότητα με τον συνδυασμό 1 0 1 και αυτή είναι 0.913357. Και πάλι δεν μπορούμε να προβούμε σε κάποια συμπεράσματα για τον συγκεκριμένο χρήστη όσον αφορά το ποιος συνδυασμός

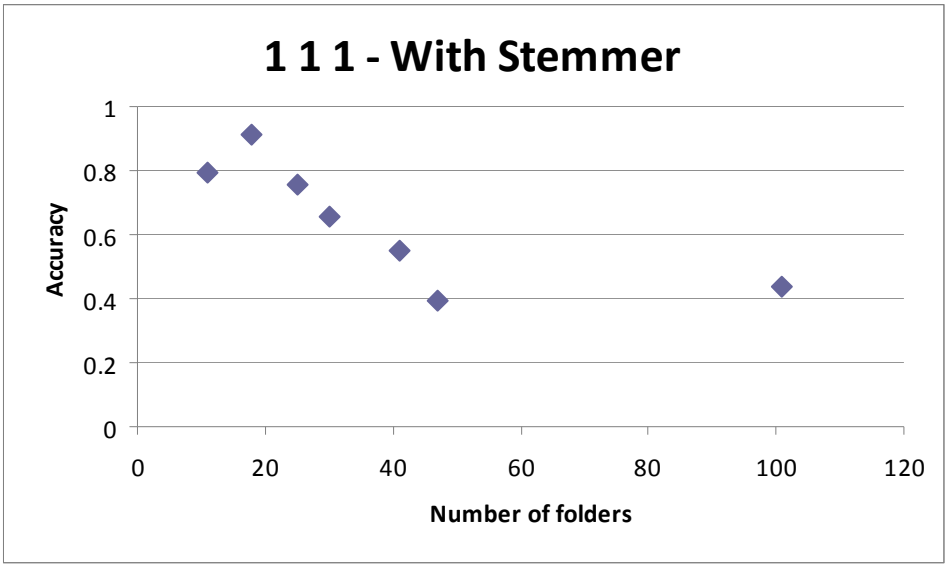
βαρών μπορεί να φέρει τα καλύτερα αποτελέσματα. Χαμηλότερη ορθότητα(0.82491) παρουσίασε στην στήλη 0 1 0.

User	Min Acc	Max Acc
beck-s	0 1 0	10 5 1
farmer-d	0 1 0	1 1 1
kaminski-v	1 0 0	10 5 1
kitchen-l	0 0 1	10 5 1
lokay-m	0 1 0	10 5 1
sanders-r	1 0 0	10 5 1
williams-w3	0 1 0	1 0 1

Πίνακας 5: Συνοπτικά Αποτελέσματα

Από τα πιο πάνω αποτελέσματα και όπως φαίνεται και από τον Πίνακας 5, σε όλους τους χρήστες η ελάχιστη ορθότητα εμφανίζεται στις περιπτώσεις όπου λαμβάνεται υπόψη μόνο ένα από τα τρία πεδία. Συγκεκριμένα, ο συνδυασμός 0 1 0 σε τέσσερις από τις εφτά περιπτώσεις είχε το χειρότερο αποτέλεσμα. Αντίθετα, σε πέντε από τις εφτά περιπτώσεις ο συνδυασμός 10 5 1 είχε σαν αποτέλεσμα την μέγιστη ορθότητα.

Στο Σχήμα 20 φαίνεται μια γραφική παράσταση του αριθμού των καταλόγων κάθε χρήστη και της ορθότητας. Χρησιμοποιήθηκαν ίσα βάρη και στα τρία πεδία και έγινε χρήση του Stemmer.



Σχήμα 20: Πως ο αριθμός των καταλόγων κάθε χρήστη επηρεάζει την ορθότητα.

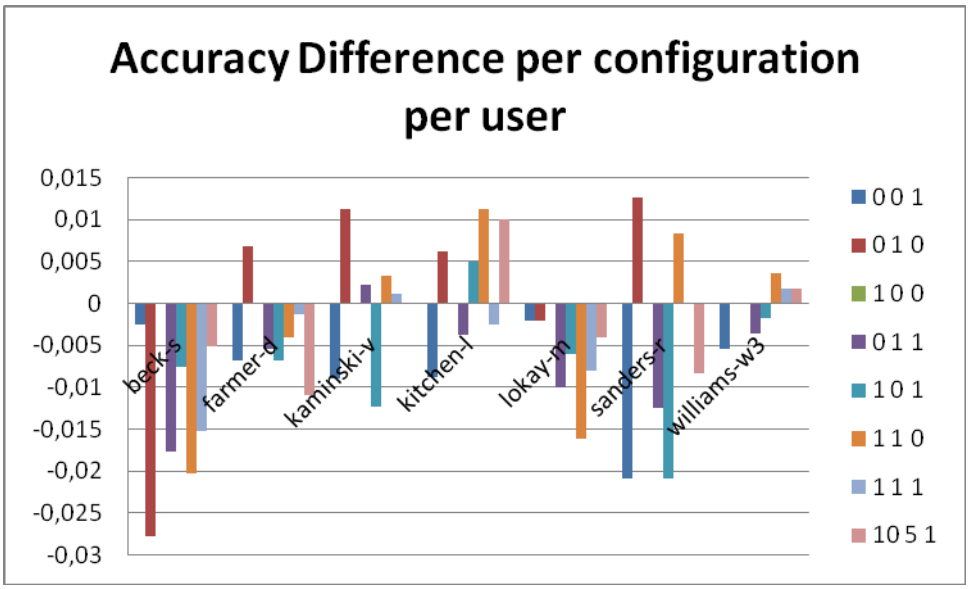
Όπως μπορούμε να διακρίνουμε, ο αριθμός των καταλόγων που έχει κάποιος χρήστης επηρεάζει άμεσα την ορθότητα του ταξινομητή. Οι χρήστες που έχουν λιγότερους καταλόγους, τείνουν να έχουν ψηλότερη ορθότητα από αυτούς που έχουν περισσότερους και οι χρήστες με περισσότερους φακέλους έχουν χαμηλότερη. Ένας λόγος είναι ότι οι χρήστες που έχουν περισσότερους φακέλους έχουν πιο πολύπλοκες τεχνικές ταξινόμησης.

5.4 Χρήση Stemmer

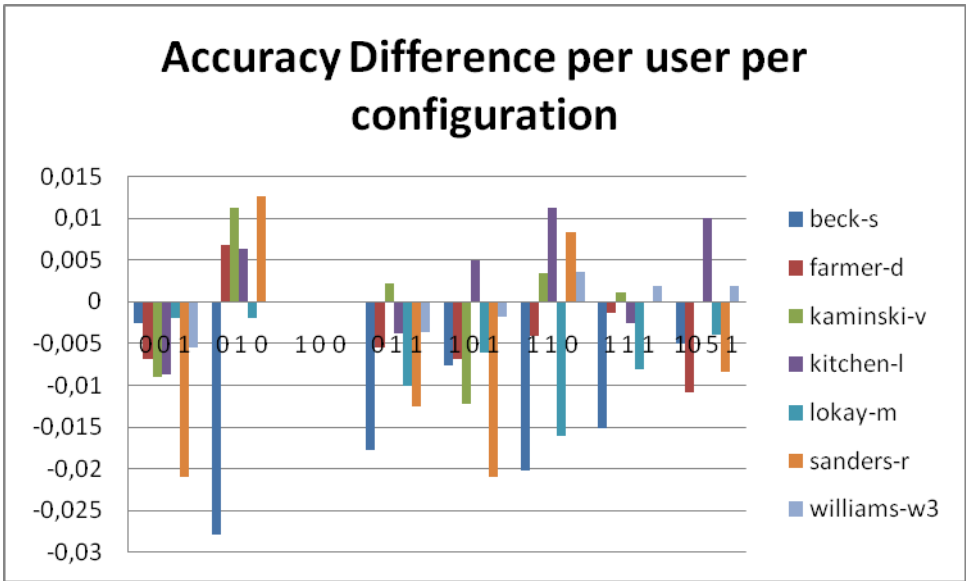
Το πρόσθετο προσφέρει την δυνατότητα στον χρήστη να επιλέξει αν κατά την διάρκεια της ταξινόμησης θα γίνει stemming(στελέχωση) στους όρους. Πιο κατω παρουσιάζονται τα αποτελέσματα πειραμάτων που έγιναν ώστε να εξεταστεί στην πράξη τον αντίκτυπο που θα έχει η χρήση του Stemmer.

Για να αξιολογήσουμε την χρήση του Stemmer αναλύθηκαν τα αποτελέσματα των πειραμάτων που έγιναν στο 5.2. Το Σχήμα 21 παρουσιάζει την διαφορά της ορθότητας που υπολογίζεται ως $\left(Accuracy_{WithStemmer} - Accuracy_{NoStemmer}\right)$. Συνολικά αρνητική επίδραση είχαμε σε 32 πειράματα, θετική επίδραση σε 14 ενώ σε 10 δεν είχαμε καμιά επίδραση με την χρήση του Stemmer. Ο συνδυασμός 1 0 0 όπως είναι φυσικό έχει σε όλα τα πειράματα μηδενική επίδραση αφού δεν εφαρμόζεται stemming στο πεδίο From.

Η μεγαλύτερη αρνητική επίδραση που καταγράφηκε είναι η περίπτωση του beck-s με τον συνδυασμό βαρών 0 1 0 που υπήρξε μια επίδραση -0.027848 δηλαδή μειώθηκε η ορθότητα σχεδόν 3%. Από την άλλη η μεγαλύτερη θετική επίδραση σημειώθηκε στον χρήστη sanders-r και συγκεκριμένα στον συνδυασμό 0 1 0 με επίδραση σχεδόν 1.5%.



Σχήμα 21: Διαφορά ορθότητας ανά διαμόρφωση βαρών ανά χρήστη

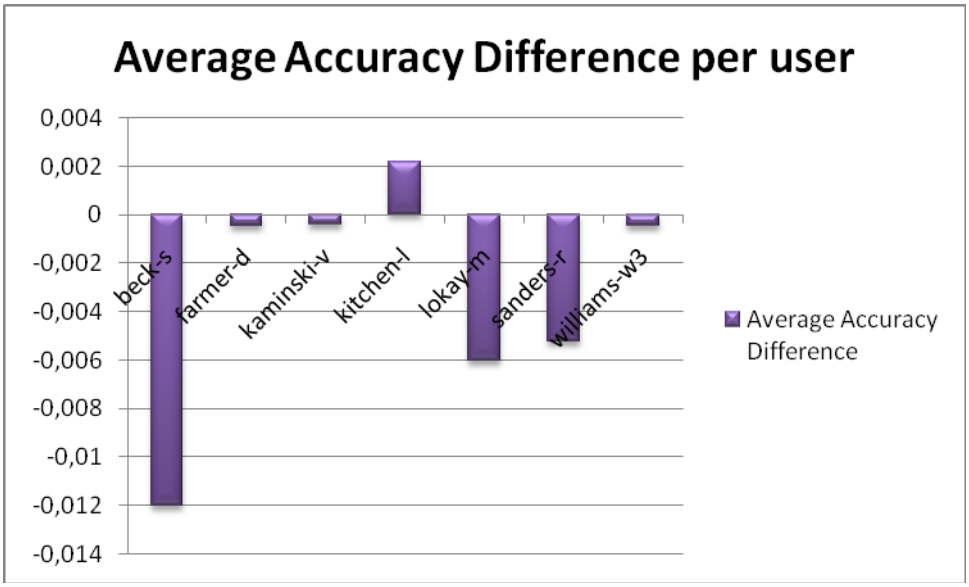


Σχήμα 22: Διαφορά ορθότητας ανά χρήστη ανά διαμόρφωση βαρών

Κάτι άλλο ενδιαφέρον που βγαίνει τα πιο πάνω πειράματα είναι να δούμε πως οι διάφοροι συνδυασμοί βαρών και η τεχνική ταξινόμησης κάθε χρήστη επηρεάζει τα αποτελέσματα της χρήσης του Stemmer. Το Σχήμα 22 δείχνει τις ίδιες πληροφορίες που δείχνει το Σχήμα 21 απλά ομαδοποιούνται ανά συνδυασμό βαρών. Όπως φαίνεται, χρησιμοποιώντας τον συνδυασμό 0 0 1, ο Stemmer έχει πάντα αρνητική επίδραση. Επίσης, χρησιμοποιώντας τον συνδυασμό 1 0 0 δεν έχουμε επίδραση. Αυτό είναι απόλυτα φυσιολογικό αφού όπως αναφέρθηκε και πιο πάνω στο πεδίο From δεν εφαρμόζεται stemming. Από τους υπόλοιπους έξι συνδυασμούς, μπορούμε να δούμε ότι

για κάθε συνδυασμό υπάρχουν χρήστες που έχουν αρνητική επίδραση και χρήστες με θετική. Επίσης από το Σχήμα 21 βλέπουμε ότι για κάθε χρήστη με εξαίρεση τον beck-s που για όλους τους συνδυασμούς έχει αρνητική επίδραση, οι υπόλοιποι έξι έχουν και αρνητική και θετική επίδραση ανάλογα με τον συνδυασμό.

Άρα σαν γενικό συμπέρασμα από τα δύο αυτά σχήματα είναι ότι η χρήση ή όχι του stemmer εξαρτάται και από την τεχνική ταξινόμησης που χρησιμοποιεί κάποιος χρήστης αλλά και από ποιο συνδυασμό χρησιμοποιεί.



Σχήμα 23: Διαφορά μέσης ορθότητας ανά χρήστη

Το Σχήμα 23 δείχνει τον μέσο όρο της ορθότητας ανά χρήστη. Όπως μπορούμε να παρατηρήσουμε μόνο ένας χρήστης και συγκεκριμένα ο kitchen-l είχε σχετική αύξηση στην ορθότητα του ενώ οι υπόλοιποι έξι χρήστες είχαν μείωση. Από αυτό φαίνεται ότι μάλλον η χρήση του Stemmer στις περισσότερες περιπτώσεις δε βοηθά.

Γενικά καλό θα ήταν να χρησιμοποιείται stemming μόνο όταν το είδος των μηνυμάτων του συγκεκριμένου χρήστη και ο συνδυασμός βαρών που θα χρησιμοποιηθεί για τα συγκεκριμένα μηνύματα είναι κατάλληλα για stemming. Αυτό μπορεί να βγει με μια ανάλυση που θα γίνεται κατά την διάρκεια της λειτουργίας του πρόσθετου όπου στην αρχή της λειτουργίας του δεν θα χρησιμοποιείται ο Stemmer αφού όπως είδαμε τις περισσότερες φορές έχει αρνητική επίδρασή αλλά ταυτόχρονα θα αναλύεται κατά πόσο η επίδραση αν χρησιμοποιούσαμε Stemmer θα ήταν θετική. Όταν μετά από μερικές

ταξινομήσεις μπορούμε να πούμε ότι έχουμε θετική επίδραση τότε θα ενεργοποιούμε τον Stemmer ώστε να λειτουργεί κατά την διάρκεια της ταξινόμησης. Καλό θα ήταν να συνεχιστεί και μετά αυτή η ανάλυση ώστε αν σε κάποια φάση αλλάξει το είδος των μηνυμάτων και δεν συνεχίσουμε να έχουμε θετική επίδραση με Stemmer να απενεργοποιείται.

Κεφάλαιο 6

Συμπεράσματα

6.1 Επίλογος

Αυτή η διπλωματική εργασία είχε ως στόχο την ανάλυση και μελέτη διαφόρων αλγόριθμων ταξινόμησης με σκοπό την δημιουργία ενός υποσυστήματος αυτόματης ταξινόμησης του ηλεκτρονικού ταχυδρομείου το οποίο χρησιμοποιεί κάποιες τεχνικές από τον κλάδο ανάκτησης πληροφοριών. Συγκεκριμένα χρησιμοποιήθηκε ο αλγόριθμος Porter Stemmer για στελέχωση κειμένου, έγινε χρήση τεχνικής αποκλεισμού λέξεων(stop words) και κανονικοποίηση κειμένου. Χρησιμοποιήθηκε το πιθανοκρατικό μοντέλο και πιο συγκεκριμένα υλοποιήθηκε ο αλγόριθμος του Bayes για την ταξινόμηση.

Ο αλγόριθμος του Bayes είναι ο trivial αλγόριθμος ταξινόμησης αλλά ακόμα και έτσι σε κάποιες περιπτώσεις έχει πολύ καλά αποτελέσματα. Αυτό εξαρτάται από την τεχνική ταξινόμησης που χρησιμοποιεί κάποιος χρήστης, όπως επίσης και από τον αριθμό των καταλόγων που έχει.

Για την αξιολόγηση του συστήματος χρησιμοποιήθηκε ένα υποσύνολο από το Enron Corpus. Συγκεκριμένα, χρησιμοποιήθηκαν τα ηλεκτρονικά μηνύματα εφτά χρηστών οι οποίοι είναι αντιπροσωπευτικοί. Διεξήχθηκαν 16 πειράματα που είναι ο συνδυασμός 8 διαφορετικών διαμορφώσεων βαρών με χρήση Stemmer και χωρίς. Τα αποτελέσματα αναλύθηκαν και έδειξαν το πώς η διαμόρφωση των βαρών και η χρήση του Stemmer

επηρεάζουν την ορθότητα του ταξινομητή. Ενδεικτικά, σε κάποια πειράματα η ορθότητα ξεπέρασε το 91% το οποίο θεωρείται ικανοποιητικό.

6.2 Μελλοντική Εργασία

Για να γίνει πιο ολοκληρωμένη η αξιολόγηση των αποτελεσμάτων, καλό θα ήταν να υλοποιηθούν και κάποιοι άλλοι αλγόριθμοι ταξινόμησης ή ακόμα και να γίνει συνδυασμός κάποιων από αυτούς. Με αυτό τον τρόπο, θα μπορεί να αποδειχτεί ποιος αλγόριθμος είναι ο πιο κατάλληλος για την ταξινόμηση των ηλεκτρονικών μηνυμάτων.

Τέλος, για να αυξηθεί η ορθότητα του ταξινομητή, μια παραλλαγή θα ήταν το stemming να χρησιμοποιείται μόνο όταν το είδος των μηνυμάτων του συγκεκριμένου χρήστη και ο συνδυασμός βαρών που θα χρησιμοποιηθεί για τα μηνύματα αυτά είναι κατάλληλα για stemming. Για να καθοριστεί κατά πόσον είναι κατάλληλα θα πρέπει να γίνεται μια παράλληλη ανάλυση κατά την διάρκεια λειτουργίας του υποσυστήματος, όπως περιγράφεται στο 5.4.

Βιβλιογραφία

[1] P. Clark and T. Niblett. The cn2 induction algorithm. Machine Learning, 3(4):261-283, 1989.

[2] T. Cover, P. Hart. Nearest neighbor pattern classification. IEEE Trans Inform Theory 13(1):21–27, 1967

[3] J. Diederich, J. L. Kindermann, E. Leopold, and G. Paass. Authorship attribution with support vector machines. Applied Intelligence, 19(1/2):109-123, 2003.

[4] H. Drucker, V. Vapnik, and D. Wu. Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5):1048-1054, 1999.

[5] A. Finn, N. Kushmerick, and B. Smyth. Genre classification and domain transfer for information filtering. In F. Crestani, M. Girolami, and C. J. van Rijsbergen, editors, Proceedings of ECIR-02, 24th European Colloquium on Information Retrieval Research, pages 353{362, Glasgow, UK, 2002.

[6] W. Frakes and R. Baeza-Yates, editors. Information retrieval, data structures and algorithms. Prentice Hall, New York, Englewood Cliffs, N.J., 1992

[7] E. Giacoletto and K. Aberer. Automatic expansion of manual email classifications based on text analysis. In Proceedings of ODBASE'03, the 2nd International Conference on Ontologies, Databases, and Applications of Semantics, pages 785-802, 2003.

[8] S. Kiritchenko and S. Matwin. Email classification with co-training. In Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, 2001.

[9] B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In Proceedings of ECML'04, 15th European Conference on Machine Learning, pages 217-226, 2004.

[10] R. Krovetz. Viewing Morphology as an Inference Process. Artificial Intelligence 118 (2000): 277–294, 1993

[11] D. D. Lewis. Representation and learning in information retrieval. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.

[12] D. D. Lewis and K. A. Knowles. Threading electronic mail: a preliminary study Information Processing and Management, 33(2):209-217, 1997.

[13] T. R. Payne and P. Edwards. Interface agents that learn: An investigation of learning issues in a mail agent interface. Applied Artificial Intelligence, 11(1):1-32, 1997.

[14] M. F. Porter. An algorithm for suffix stripping. Program 14(3):130–137. 33, 529, 1980

[15] J. Provost. Naive-bayes vs. rule-learning in classification of email. Technical Report AI-TR-99-284, University of Texas at Austin, Artificial Intelligence Lab, 1999.

[16] J. Rennie. iFile: An application of machine learning to e-mail filtering. In Proceedings of KDD'2000 Workshop on Text Mining, 2000.

[17] D. M. Christofer, R. Prabhakar, H. Schutze. An Introduction to Information Retrieval. Cambridge University Press, 2008.