

Ατομική Διπλωματική Εργασία

ΕΞΑΓΩΓΗ ΣΥΓΚΕΙΜΕΝΟΥ ΧΡΗΣΤΗ

Δημήτρης Σάββα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εξαγωγή Συγκειμένου Χρήστη

Δημήτρης Σάββα

Επιβλέπων Καθηγητής

Δρ. Μάριος Δικαιάκος

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

Μάιος 2011

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Μάριο Δικαιάκο, για την ευκαιρία που μου έδωσε να ασχοληθώ με την παρούσα διπλωματική καθώς επίσης και για την πολύτιμη καθοδήγηση του καθ' όλη τη διάρκεια του έτους. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή Δρ. Ιωάννη Κατάκη, για τη βοήθεια που μου προσέφερε κατά το δεύτερο μέρος της διπλωματικής. Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και συμφοιτητές μου για τη βοήθεια που μου προσέφεραν για την αξιολόγηση της διπλωματικής.

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας, είναι η δημιουργία ενός εργαλείου (ContextExtractor) το οποίο θα έχει τη δυνατότητα να εξάγει το συγκεκριμένο του χρήστη (user context) και να προσδιορίζει το περιβάλλον εργασίας του, χαρακτηρίζοντάς το με διάφορες παραμέτρους που θα εξάγει.

Το εργαλείο αυτό, εκτελείται στο παρασκήνιο και καταγράφει τη συμπεριφορά του χρήστη και την αλληλεπίδρασή του με το σύστημα. Συγκεκριμένα, καταγράφεται το περιεχόμενο το οποίο διαβάζει ο χρήστης από την οθόνη του σε μορφή κειμένου καθώς επίσης και μετα-πληροφορίες οι οποίες χαρακτηρίζουν το περιεχόμενο αυτό. Τέλος, το εργαλείο, αναλύει το περιεχόμενο που συλλέγει και μπορεί να προσδιορίσει το θέμα του, χωρίζοντας το σε κατηγορίες οι οποίες προσδιορίζουν το περιβάλλον στο οποίο δούλευε ο χρήστης.

Απώτερος σκοπός είναι μέσω αυτού του εργαλείου να γίνει μια ανάλυση του συγκεκριμένου του χρήστη ούτως ώστε με τη γνώση που θα αποκτήσει το εργαλείο, να είναι σε θέση να γνωρίζει το συγκεκριμένο του χρήστη και έτσι να μπορεί να χρησιμοποιηθεί σε περιπτώσεις λήψης αποφάσεων για διάφορα συμβάντα στο μέλλον τα οποία είτε θα κρίνονται σχετικά με το συγκεκριμένο και θα τους δίνεται υψηλή προτεραιότητα, είτε το αντίθετο. Μια τέτοια χρήση για παράδειγμα, είναι για το φιλτράρισμα της εισερχόμενης ηλεκτρονικής αλληλογραφίας. Με τη βοήθεια του εργαλείου, θα είναι γνωστό εάν ένα νέο εισερχόμενο μήνυμα εμπίπτει στο συγκεκριμένο του χρήστη και άρα θα ήταν σωστό να ειδοποιηθεί ο χρήστης γι' αυτό άμεσα, αλλιώς να προστεθεί κανονικά στα εισερχόμενά του. Επίσης, μια άλλη χρήση, είναι η χρησιμοποίηση του εργαλείου για αξιολόγηση της κατανομής του χρόνου στις διάφορες εργασίες του χρήστη [22], οπότε να γνωρίζει ο ίδιος τον ωφέλιμο χρόνο για την κάθε εργασία.

Περιεχόμενα

Ευχαριστίες.....	iii
Περίληψη.....	iv
Περιεχόμενα.....	v
Κεφάλαιο 1 Εισαγωγή	1
1.1 Συγκείμενο χρήστη.....	1
1.2 Σχετικές εργασίες	2
1.3 Ανάκτηση Πληροφοριών και Ανακάλυψη Γνώσης	3
1.4 Παρακολούθηση χρήστη.....	4
1.5 Στόχοι εργασίας.....	5
Κεφάλαιο 2 Επιλογές συστήματος – τεχνολογίες.....	6
2.1 Επιλογή λειτουργικού συστήματος.....	6
2.2 Επιλογή μεθόδου εξαγωγής πληροφοριών.....	7
2.3 Βιβλιοθήκη TextCaptureX	11
2.4 Windows Search.....	13
2.5 Lucene	13
2.6 Μετρικές.....	14
Κεφάλαιο 3 Ανάκτηση πληροφοριών.....	16
3.1 Ορισμός συγκεκριμένου (context) στα πλαίσια της εφαρμογής.....	16
3.2 Εξαγωγή ακατέργαστων πληροφοριών (raw data)	17
Κεφάλαιο 4 Περιβάλλον υλοποίησης	22
4.1 Γενική περιγραφή συστήματος	22
4.2 Ανάλυση περιβάλλοντος εξαγωγής συγκεκριμένου	24

4.3	Ανάλυση - επεξεργασία στιγμιότυπου	24
4.4	Cosine Similarity.....	28
4.5	Μεταβολή συγκειμένου (context shift).....	28
4.6	Ομαδοποίηση (clustering).....	29
Κεφάλαιο 5	Χαρακτηριστικά συστήματος.....	34
5.1	Ανάλυση visualization	34
5.2	Ευρετηρίαση με Lucene	36
Κεφάλαιο 6	Αξιολόγηση	38
6.1	Περιγραφή πειραμάτων.....	38
6.2	Αξιολόγηση εξαγωγής πληροφοριών.....	39
6.3	Αξιολόγηση αλγορίθμου ομαδοποίησης.....	40
6.4	Γενική αξιολόγηση συστήματος	46
Κεφάλαιο 7	Συμπεράσματα	48
7.1	Επίλογος.....	48
7.2	Μελλοντική εργασία	49
Βιβλιογραφία		50
Παράρτημα Α		A-1
Παράρτημα Β		B-1

Κεφάλαιο 1

Εισαγωγή

1.1	Συγκείμενο χρήστη	1
1.2	Σχετικές εργασίες	2
1.3	Ανάκτηση Πληροφοριών και Ανακάλυψη Γνώσης	3
1.4	Παρακολούθηση χρήστη	4
1.5	Στόχοι εργασίας	5

Στο κεφάλαιο αυτό, θα γίνει μια εισαγωγική αναφορά στο συγκείμενο χρήστη, στην ανακάλυψη γνώσης και στη μέθοδο με την οποία επιτυγχάνεται αυτή. Οι δύο αυτές έννοιες είναι οι βάσεις για την παρούσα διπλωματική. Επίσης, θα γίνει μια αναφορά στα υπάρχοντα συστήματα και τεχνολογίες στα οποία βασίστηκε η διπλωματική.

1.1 Συγκείμενο χρήστη

Συγκείμενο χρήστη (user context) ορίζεται η πληροφορία η οποία χαρακτηρίζει το περιβάλλον στο οποίο εργάζεται ένας χρήστης μια δεδομένη στιγμή. Η πληροφορία αυτή, συλλέγεται με την τεχνική παρακολούθησης χρήστη (user observation ή user logging). Συνήθως έχει τη μορφή μετα-πληροφορίας, δηλαδή πληροφορίας η οποία περιγράφει την πραγματική πληροφορία. Ένα παράδειγμα του τι είναι συγκείμενο χρήστη είναι το εξής: Ο χρήστης Α ψάχνει στο διαδίκτυο για μεταχειρισμένα αυτοκίνητα. Τη στιγμή εκείνη, οι λέξεις «αυτοκίνητο», «μεταχειρισμένο», «αυτόματο»,

«τετραθέσιο» είναι λέξεις οι οποίες ανήκουν στο συγκεκριμένο του (context) αφού μας δίνουν στοιχεία που περιγράφουν την πραγματική πληροφορία που αντλεί ο χρήστης από τον υπολογιστή.

Το συγκεκριμένο του χρήστη, συνήθως συλλέγεται μέσω παρακολούθησης χρήστη (user observation) από μια context-sensitive εφαρμογή (ή συσκευή) η οποία είναι σε θέση να καταγράψει το συγκεκριμένο για μια δεδομένη στιγμή. Επιπρόσθετα, η εφαρμογή αυτή μπορεί να υποστηρίξει το χρήστη παρέχοντας του εισηγήσεις ή άλλης μορφής υπηρεσίες [12, 23] οι οποίες βασίζονται στα δεδομένα τα οποία έχει συλλέξει και στην γνώση την οποία απέκτησε στην πορεία.

Για να καταστεί μια εφαρμογή ευαίσθητη στο συγκεκριμένο ενός χρήστη, είναι απαραίτητο με κάποιον τρόπο να παρακολουθεί τον χρήστη και να καταγράφει τις αλληλεπιδράσεις του με το σύστημα. Με το θέμα αυτό ασχολείται ένας κλάδος της Πληροφορικής που ονομάζεται Context-Aware Computing [21, 26].

Ο κλάδος του Context-Aware Computing έχει αναπτυχθεί αρκετά τα τελευταία χρόνια. Σ' αυτό συμβάλουν και η ανάπτυξη των κινητών συσκευών και η ανάγκη για περισσότερη διείσδυση στο εργασιακό περιβάλλον του χρήστη ούτως ώστε να αυξηθεί η ποιότητα παροχής υπηρεσιών σ' αυτόν.

1.2 Σχετικές εργασίες

Το συγκεκριμένο χρήστη (user context), είναι κάτι που απασχολεί εδώ και αρκετό καιρό έναν από τους παλαιότερους τομείς της Πληροφορικής, τον τομέα του Information Retrieval. Οι χρήσεις αυτού, σύμφωνα και με τη βιβλιογραφία, είναι αρκετές [6, 9, 11, 12, 13, 14, 24]. Μία σημαντική χρήση του, όπως αναφέρεται και στην εργασία [9, 24, 25], είναι για να βοηθήσει στην αναζήτηση αρχείων στην επιφάνεια εργασίας (desktop search). Στη συγκεκριμένη υλοποίηση, το εργαλείο παρακολουθούσε τα ίχνη από τα αρχεία καταγράφοντας τα system calls και αναγνωρίζοντας πότε αυτά αναφέρονται σε αρχεία του χρήστη. Μία σημαντική καινοτομία, είναι ότι μπορούσε να παρέχει αναζήτηση όχι μόνο βάσει περιεχομένου (content-only search) αλλά και βάσει συγκεκριμένου (context-enhanced search).

Στη συγκεκριμένη εργασία, υπήρχε η δυνατότητα αναγνώρισης σχέσεων μεταξύ αρχείων και αναπαράσταση τους με γράφους [4]. Οι σχέσεις αυτές, δημιουργούνταν βασισμένες στην τοπικότητα χρόνου (temporal locality). Δηλαδή, τα διάφορα αρχεία συνδέονταν αν υπήρχε κάποια σχέση μεταξύ τους σε κάποιο χρονικό διάστημα. Τέτοιες σχέσεις, είναι για παράδειγμα η διατήρηση του πρώτου διαδόχου κάποιου αρχείου. Αν για παράδειγμα μετά από την πρόσβαση στο αρχείο A υπάρξει πρόσβαση στο αρχείο B, τότε δημιουργείται μια σχέση από το A στο B η οποία δηλώνει ότι αν ζητηθεί το A (σαν αποτέλεσμα μέσα από μια αναζήτηση περιεχομένου), τότε επέστρεψε και το B. Έτσι, σε μια αναζήτηση βάσει συγκεκριμένου [5], γινόταν η αναζήτηση βάσει περιεχομένου και ακολούθως τα αποτελέσματα περνούσαν από το γράφο με τις σχέσεις μεταξύ των αρχείων. Έτσι, ανακαλύπτονταν αρχεία τα οποία αρχικά (με την αναζήτηση βάσει περιεχομένου), δεν είχαν ανακαλυφθεί αλλά, βάσει και του γράφου, σχετίζονταν σε μεγάλο βαθμό με τα αρχικά αρχεία.

Η παρούσα διπλωματική, χτίζει πάνω σε παλαιότερη διπλωματική [31] που είχε σκοπό την αυτόματη ταξινόμηση της ηλεκτρονικής αλληλογραφίας του χρήστη.

Ο σκοπός αυτής της διπλωματικής εργασίας, είναι η αναζήτηση νέων τρόπων εξαγωγής του συγκεκριμένου του χρήστη και χρησιμοποίησής του. Αυτό που εισάγει είναι η δυναμική επεξεργασία του καθώς αυτό συλλέγεται και άμεση ανάδραση από το σύστημα σε αντίθεση με τα υπάρχοντα συστήματα τα οποία έκαναν την συλλογή και επεξεργασία σε δύο διαφορετικά χρονικά στάδια.

Επιπλέον, η υλοποίηση ξεφεύγει από τον συνηθισμένο τρόπο εξαγωγής πληροφοριών μέσα από παρακολούθηση των αρχείων και προσπαθεί να το κάνει μέσω μιας όχι τόσο διαδεδομένης μεθόδου, της παρακολούθησης των παραθύρων.

1.3 Ανάκτηση Πληροφοριών και Ανακάλυψη Γνώσης

Ανάκτηση πληροφοριών, είναι ο κλάδος ο οποίος ασχολείται με την αναζήτηση πληροφοριών και μετα-πληροφοριών σε κείμενα, έγγραφα και γενικά οποιαδήποτε πηγή η οποία μπορεί να παρέχει δεδομένα. Συνήθως οι πηγές αυτές έχουν αδόμητη φύση, κάτι που κάνει την ανάκτηση πληροφοριών απ' αυτές ακόμη πιο ενδιαφέρουσα.

Παραδοσιακά, η ανάκτηση πληροφοριών γίνεται σε συλλογές από στατικά δεδομένα. Σε περίπτωση ροής δεδομένων (stream data), όπως για παράδειγμα οι επερωτήσεις στη βάση ενός συστήματος κάποιας τράπεζας οι οποίες στέλνονται συνεχώς από τους χρήστες, η πρόκληση για αναζήτηση πληροφοριών σ' αυτά είναι μεγαλύτερη αφού απαιτείται η on-the-fly εφαρμογή των οποιωνδήποτε αλγορίθμων οι οποίοι πρέπει να είναι τόσο αποδοτικοί όσο και αποτελεσματικοί ούτως ώστε να πετύχουν το στόχο τους σε χρόνο μικρότερο από αυτόν που ρέει η πληροφορία.

Οι εφαρμογές της ανάκτησης πληροφοριών είναι αρκετές καθώς υπάρχουν αρκετά πεδία που δίνουν τη δυνατότητα αλλά και απαιτούν την ανάκτηση πληροφοριών για εξυπηρέτηση αναγκών που σχετίζονται με την ποιότητα των παρεχόμενων υπηρεσιών. Μερικές τέτοιες υπηρεσίες είναι η προσωπική ανάκτηση πληροφοριών [6, 8], ομαδοποίηση, ταξινόμηση.

Ανακάλυψη γνώσης, είναι η διαδικασία αναζήτησης σε μεγάλο όγκο δεδομένων για διάφορα πρότυπα (patterns) τα οποία μπορούν να χαρακτηριστούν σαν γνώση για τα συγκεκριμένα δεδομένα.

Η παρούσα διπλωματική, θα ασχοληθεί με την προσωπική ανάκτηση πληροφοριών και με την ομαδοποίηση μέσω της γνώσης που αποκτάται από την ανακτώμενη πληροφορία. Απώτερος στόχος είναι η οργάνωση της προσωπικής πληροφορίας του χρήστη και η ανάλυση της για εξαγωγή θεματικών οντοτήτων οι οποίες να διατίθενται σε άλλα συστήματα για λήψη αποφάσεων ή παροχή υπηρεσιών.

1.4 Παρακολούθηση χρήστη

Η παρακολούθηση χρήστη, είναι μία μέθοδος ανακάλυψης γνώσης από το περιβάλλον εργασίας ενός χρήστη. Για την παρούσα διπλωματική, γίνεται παρακολούθηση της επιφάνειας εργασίας (desktop) του χρήστη [7] και ανάκτηση πληροφοριών από αυτήν.

Η παρακολούθηση χρήστη, γίνεται είτε με διάφορα εργαλεία τα οποία τρέχουν στο παρασκήνιο χωρίς να παρενοχλούν τον χρήστη είτε με παρακολούθηση. Καταγράφονται όλες οι κινήσεις και γενικά η αλληλεπίδραση του χρήστη με το σύστημα και τα αποτελέσματα της παρακολούθησης αναλύονται για εξαγωγή πληροφοριών και συμπερασμάτων [29].

1.5 Στόχοι εργασίας

Οι στόχοι της παρούσας διπλωματικής, περιλαμβάνουν την υλοποίηση ενός εργαλείου το οποίο να εξάγει επιτυχώς πληροφορίες κατά τη χρήση ενός υπολογιστή και να τις επεξεργάζεται κατάλληλα. Η επεξεργασία, πρέπει να βοηθά στην ανακάλυψη γνώσης η οποία δεν ήταν εμφανής και η οποία συμβάλλει στον προσδιορισμό της εργασίας την οποία εκτελούσε ο χρήστης κατά την περίοδο που έγινε η λήψη των πληροφοριών.

Επίσης, η εργασία επιδιώκει να παρουσιάσει ένα χρονοδιάγραμμα στο οποίο να φαίνεται πως μεταβαλλόταν η εργασία του χρήστη μεταξύ διαφόρων θεμάτων με την πάροδο του χρόνου και έτσι ο χρήστης να είναι σε θέση να αξιολογήσει την κατανομή του χρόνου του στις διάφορες εργασίες τις οποίες εκτέλεσε, με σκοπό στο μέλλον να μπορεί να αξιοποιήσει καλύτερα τον χρόνο του.

Η εργασία αυτή, είναι σημαντική για το λόγο ότι πλέον η ροή της πληροφορίας έχει αυξηθεί ραγδαία και συνεχίζει να αυξάνεται με ακατάπαυστους ρυθμούς με αποτέλεσμα ο χρήστης να μην μπορεί να την ελέγξει ή να την προσαρμόσει στις δικές του ανάγκες. Σ' αυτό, παίζει ένα μεγάλο ρόλο η εξάπλωση του διαδικτύου και η αύξηση των χώρων αποθήκευσης με αποτέλεσμα την αύξηση και των προσωπικών αρχείων. Έτσι, η καθημερινή εργασία ενός χρήστη περιπλέκεται με αποτέλεσμα ο χρήστης να μην είναι σε θέση να οργανώσει τις διάφορες εργασίες του ή να αναγνωρίσει το θέμα αυτών. Με τη βοήθεια του εργαλείου που υλοποιήθηκε για την παρούσα διπλωματική, ο χρήστης μπορεί να συμβουλευτεί το εργαλείο και να αξιολογήσει τις εργασίες του. Επίσης, το εργαλείο αυτό, μπορεί να χρησιμοποιηθεί από άλλες εφαρμογές οι οποίες παρέχουν πληροφορία στο χρήστη, με στόχο το φιλτράρισμα αυτής της πληροφορίας, ούτως ώστε να μειωθεί ο όγκος της αχρείαστης πληροφορίας. Αυτό επιτυγχάνεται με την εξαγωγή του συγκεκριμένου του χρήστη και τον έλεγχο του κατά πόσο κάποια πληροφορία σχετίζεται μ' αυτό ή όχι.

Κεφάλαιο 2

Επιλογές συστήματος – τεχνολογίες

2.1	Επιλογή λειτουργικού συστήματος	6
2.2	Επιλογή μεθόδου εξαγωγής πληροφοριών	7
2.3	Βιβλιοθήκη TextCaptureX	11
2.4	Windows Search	13
2.5	Lucene	13
2.6	Μετρικές	14

Στο κεφάλαιο αυτό, γίνεται αναφορά στα υπάρχοντα συστήματα και βιβλιοθήκες τα οποία χρησιμοποιούνται από το εργαλείο ContextExtractor. Αυτά περιλαμβάνουν επίσης το περιβάλλον στο οποίο τρέχει το εργαλείο και τον τρόπο λειτουργίας του.

2.1 Επιλογή λειτουργικού συστήματος

Μια από τις πρώτες σημαντικές αποφάσεις που έπρεπε να ληφθούν προτού υλοποιηθεί το σύστημα, ήταν η πλατφόρμα λειτουργικού συστήματος την οποία θα υποστηρίζει. Αρχικά, μελετήθηκαν οι διάφορες επιλογές και συγκεκριμένα οι δυνατές λύσεις στα περιβάλλοντα Windows και Linux. Έτσι, έπρεπε να μελετηθούν τα θετικά και αρνητικά της κάθε προσέγγισης και ανάλογα να υιοθετηθεί η καταλληλότερη για την εξυπηρέτηση του στόχου του εργαλείου. Η σημαντικότερη διαφορά των δύο, ήταν η

μέθοδος με την οποία θα γινόταν η συλλογή της πληροφορίας. Οι δύο μέθοδοι αναλύονται σε βάθος στο υποκεφάλαιο 2.2. Στο περιβάλλον Windows, η δυνατή μέθοδος ήταν με παρακολούθηση των παραθύρων ενώ στο περιβάλλον Linux ήταν δυνατή η μέθοδος με παρακολούθηση των αρχείων.

Οι δύο μέθοδοι, αναλύθηκαν παράλληλα μέχρις ότου παρθεί η τελική απόφαση. Η απόφαση αυτή, στηρίχτηκε στο αν μπορούσαν πράγματι να υλοποιηθούν οι μέθοδοι στα συστήματα και αν ναι το κατά πόσο θα επέστρεφαν τα ζητούμενα αποτελέσματα με τον ιδανικό τρόπο. Τελικά, επιλέχθηκε το περιβάλλον Windows, για το λόγο ότι η μέθοδος της παρακολούθησης παραθύρων ήταν εφικτή (με τη βοήθεια και του Windows API) ενώ στα Linux η μέθοδος παρακολούθησης αρχείων παρουσίαζε προβλήματα. Οι δύο μέθοδοι παρουσιάζονται αναλυτικότερα στο υποκεφάλαιο 2.2.

Η υποστήριξη της μεθόδου παρακολούθησης αρχείων είναι κάτι που είναι εφικτό και στο περιβάλλον Windows παρόλο που αντιμετωπίζει παρόμοια προβλήματα με αυτήν των Linux όπως αναφέρεται και στο εδάφιο 2.2.1. Το αντίθετο, δηλαδή η υποστήριξη της μεθόδου παρακολούθησης παραθύρων, στο περιβάλλον Linux είναι κάτι πολύ δύσκολο. Ο λόγος είναι ότι ενώ τα Windows έχουν έναν διαχειριστή παραθύρων (window manager) ο οποίος είναι ενσωματωμένος στον πυρήνα του λειτουργικού, τα Linux δεν έχουν προκαθορισμένο window manager δίνοντας τη δυνατότητα στο χρήστη να επιλέξει όποιον αυτός επιθυμεί. Επιπλέον υπάρχει ο διαχωρισμός μεταξύ διαχειριστή παραθύρων (π.χ. Xfce, LXDE) και περιβάλλον εργασίας (desktop environment) (π.χ. GNOME, KDE) το οποίο περιλαμβάνει και το δικό του διαχειριστή παραθύρων. Αυτό κάνει τα πράγματα ακόμη πιο δύσκολα αφού το κάθε ένα χειρίζεται διαφορετικά τα παράθυρα και επίσης δεν προσφέρουν όλα κάποιο API για το χειρισμό των παραθύρων από άλλες εφαρμογές.

2.2 Επιλογή μεθόδου εξαγωγής πληροφοριών

2.2.1 Μέθοδος παρακολούθησης αρχείων

Η πρώτη μέθοδος για εξαγωγή συγκεκριμένου η οποία μελετήθηκε, αφορούσε την παρακολούθηση των αρχείων με τα οποία αλληλεπιδρά ο χρήστης [9, 24, 25].

Ένας χρήστης κατά τη διάρκεια χρήσης του υπολογιστή, ανοίγει, δημιουργεί και επεξεργάζεται διάφορα αρχεία. Αυτά μπορεί είτε να είναι αρχεία κειμένου, τα οποία να περιέχουν αυτούσια πληροφορία, είτε οποιαδήποτε άλλα αρχεία από τα οποία μπορεί να εξαχθεί μετα-πληροφορία. Ο κύριος στόχος, ήταν η παρακολούθηση των αρχείων που περιέχουν κείμενο. Η ιδέα ήταν ότι τη στιγμή που κάποιος χρήστης αλληλεπιδρά με ένα αρχείο, μπορούμε να διαβάζουμε τα περιεχόμενα του αρχείου και να τα προσθέτουμε στο συγκεκριμένο δηλώνοντας έτσι ότι μέσα στα περιεχόμενα, περιέχεται η πληροφορία η οποία χαρακτηρίζει το θέμα με το οποίο ασχολείται ένας χρήστης.

Για παράδειγμα, αν κάποιος χρήστης διαβάζει ένα αρχείο με όνομα «cyprushistory.pdf» το οποίο περιέχει κείμενο για την ιστορία της Κύπρου, τότε λέξεις-κλειδιά από τα περιεχόμενα του κειμένου χαρακτηρίζουν το συγκεκριμένο του χρήστη. Όμως, κάθε είδος αρχείου είναι γραμμένο με διαφορετική μορφή, π.χ. ένα αρχείο τύπου «.txt» είναι γραμμένο με απλό κείμενο ενώ ένα αρχείο τύπου «.pdf» είναι γραμμένο με μια διαφορετική κωδικοποίηση. Ένα πρόβλημα που προκύπτει λοιπόν, είναι ότι θα έπρεπε το εργαλείο να είναι σε θέση να χειρίζεται διαφορετικού τύπου αρχεία.

Το μεγάλο όμως πρόβλημα που πρόκυψε με τη μέθοδο αυτή και τελικά οδήγησε στην απόρριψή της, είναι ο τρόπος με τον οποίο το λειτουργικό σύστημα χειρίζεται τα αρχεία και συγκεκριμένα τα ανοικτά αρχεία. Ο τρόπος αυτός, ορίζει δύο εναλλακτικές με τις οποίες τα διάφορα προγράμματα μπορούν να ζητούν από το λειτουργικό άδεια για αλληλεπίδραση με αρχεία. Πριν περιγράψουμε τις δύο αυτές εναλλακτικές, πρέπει να ορίσουμε τι είναι ένας file descriptor. Για να δώσουμε τον ορισμό, αναφέρουμε πρώτα ότι ένα αρχείο, μπορεί να βρίσκεται είτε στο σκληρό δίσκο, είτε στην κύρια μνήμη, είτε και στα δύο (περίπτωση κατά την οποία τα δύο αντίγραφα του συγκεκριμένου αρχείου μπορεί είτε να είναι πανομοιότυπα είτε να διαφέρουν).

Ένας file descriptor λοιπόν, είναι ένας δείκτης ο οποίος δείχνει σε κάποιο αρχείο που βρίσκεται (μόνιμα αποθηκευμένο έναντι της προσωρινής αποθήκευσης στην κύρια μνήμη) στον μαγνητικό δίσκο. Για να δουλέψει λοιπόν η συγκεκριμένη μέθοδος, πρέπει με κάποιο τρόπο να είναι γνωστό ποια αρχεία υπάρχουν ανοικτά στο περιβάλλον. Για να γίνει αυτό, πρέπει να ζητήσουμε από το λειτουργικό να μας επιστρέψει τους file descriptors που βρίσκονται στο file descriptor table, έναν πίνακα που περιέχει όλους τους file descriptors για τα ανοικτά αρχεία. Ο πίνακας όμως, δεν περιέχει μόνο τους descriptors για τα αρχεία τα οποία έχει ανοικτά ο χρήστης, αλλά και descriptors για

αρχεία τα οποία έχει ανοικτά το λειτουργικό σύστημα και χρησιμοποιεί για την λειτουργία του, καθώς επίσης και οποιαδήποτε αρχεία έχει ανοικτά οποιοδήποτε άλλο πρόγραμμα τρέχει είτε στο παρασκήνιο είτε στο προσκήνιο. Όπως είναι αντιληπτό λοιπόν, αυτό δημιουργεί το πρόβλημα του πως θα ξεχωρίσουμε τα αρχεία τα οποία έχει όντως ανοίξει ο χρήστης από το υπόλοιπα τα οποία δεν χρειαζόμαστε για την εφαρμογή μας. Εδώ είναι που κάνει την είσοδό του το εργαλείο inotify [15], το οποίο παρέχει όχι ακριβώς την απάντηση στο ερώτημα, αλλά τη μοναδική εφικτή προσέγγιση η οποία μπορεί να κάνει αυτό σε κάποιο βαθμό. Το εργαλείο inotify, δίνει τη δυνατότητα παρακολούθησης των file descriptors των αρχείων που βρίσκονται κάτω από έναν κατάλογο (directory). Έτσι μπορούμε να ορίσουμε το inotify να παρακολουθεί όλα τα αρχεία σε έναν κατάλογο (π.χ. home directory στα Linux) και να γνωρίζουμε πότε τα αρχεία ανοίγουν κτλ.

Παρόλα αυτά, εξακολουθούν να υπάρχουν αρκετά προβλήματα. Το σημαντικότερο είναι ότι το λειτουργικό σύστημα, όταν ζητηθεί το άνοιγμα ενός αρχείου, αυτό να μην ανοίγει τον file descriptor, αλλά στη συνέχεια, σε περίπτωση που το αρχείο χωράει στην κύρια μνήμη (κάτι που με τα σημερινά δεδομένα, τα πλείστα αρχεία με τα οποία αλληλεπιδρά ένας καθημερινός χρήστης χωράνε), φορτώνει τα περιεχόμενα του αρχείου στην κύρια μνήμη και ακολούθως διαγράφει τον file descriptor από το file descriptor table των ανοικτών αρχείων καθιστώντας έτσι αδύνατο την ανίχνευση των ανοικτών αρχείων. Για το λόγο αυτό, η μέθοδος αυτή εγκαταλείφθηκε και τελικά υιοθετήθηκε η μέθοδος με την παρακολούθηση των παραθύρων η οποία περιγράφεται πιο κάτω.

2.2.2 Μέθοδος παρακολούθησης παραθύρων

Η δεύτερη μέθοδος με την οποία μπορούμε να εξάγουμε πληροφορία από την αλληλεπίδραση του χρήστη με τον υπολογιστή στηρίζεται στην εξαγωγή πληροφορίας από τα παράθυρα (windows) των διαφόρων προγραμμάτων με το οποία αλληλεπιδρά ο χρήστης [10, 16].

Η ιδέα προκύπτει από το γεγονός ότι στην ουσία, τα σημερινά λειτουργικά συστήματα, παρέχουν ένα γραφικό περιβάλλον εργασίας με τη χρήση παραθύρων. Μέσα στα παράθυρα αυτά, παρουσιάζονται όλες οι ψηφιακές πληροφορίες τις οποίες αντλεί ο χρήστης συμπεριλαμβανομένου και των αρχείων του (π.χ. διάβασμα ενός αρχείου

κειμένου γίνεται μέσα από το παράθυρο του κειμενογράφου), αλλά και πιο σύνθετων εννοιών όπως είναι η ροή πληροφορίας που αντλείται μέσα από το παράθυρο ενός φυλλομετρητή (web browser), το οποίο στην παρούσα εποχή αποτελεί αναπόσπαστο κομμάτι ενός υπολογιστή και ακόμη περισσότερο μιας κινητής συσκευής.

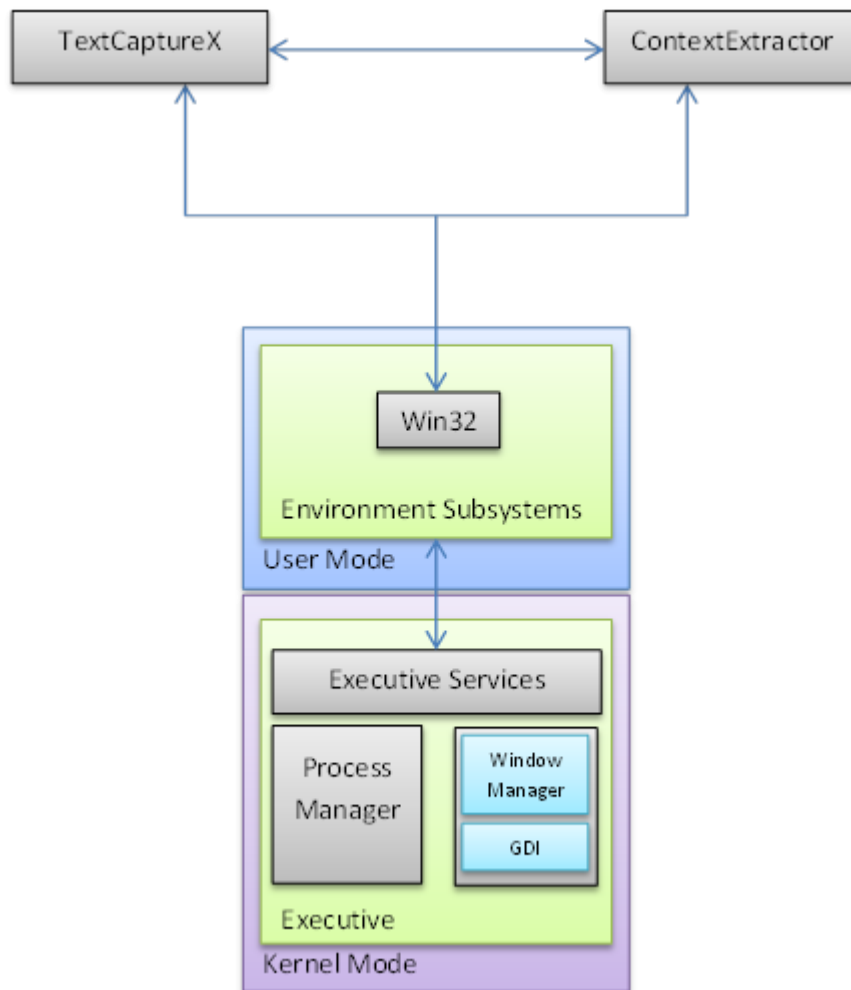
Έτσι, αυτό που γίνεται είναι να παρακολουθούμε τα παράθυρα που έχει ανοικτά ο χρήστης και να διαβάζουμε το περιεχόμενό τους. Συγκεκριμένα, το εργαλείο που υλοποιήθηκε, δεν παρακολουθεί όλα τα παράθυρα που είναι ανοικτά, αλλά το παράθυρο που κατέχει την προσοχή (focus) του χρήστη και βρίσκεται στο προσκήνιο (active window) και όχι αυτά που βρίσκονται στο παρασκήνιο (background). Η λογική αυτής της μεθοδολογίας έγκειται στο ότι το παράθυρο που βρίσκεται στο προσκήνιο έχει τις περισσότερες πιθανότητες να είναι αυτό το οποίο έχει την προσοχή του χρήστη και μέσα από αυτό ο χρήστης αντλεί την πληροφορία. Αξίζει να σημειωθεί ότι με τη μέθοδο αυτή, το εργαλείο εξάγει μόνο την πληροφορία που βρίσκεται μέσα στα ορατά (visible) περιθώρια του παραθύρου και όχι όλη την πληροφορία που βρίσκεται στο παράθυρο και μπορεί να ενεργοποιηθεί με κύλιση (scrolling) ή με άλλη ενέργεια από το χρήστη.

Ένα μειονέκτημα της μεθόδου αυτής, είναι ότι η εξαγωγή της πληροφορίας από τα παράθυρα, πρέπει να γίνεται σε καθορισμένα χρονικά διαστήματα ενώ το ιδανικό θα ήταν να μπορούσε αυτό να γίνεται κατά την λήψη συμβάντων (events) τα οποία να ενεργοποιούν την λήψη. Για παράδειγμα ένα συμβάν μπορεί να είναι η αλλαγή του περιεχομένου ενός παραθύρου. Αυτό βέβαια δημιουργεί ένα μειονέκτημα αλλά και ένα πλεονέκτημα. Το μειονέκτημα είναι ότι περιεχόμενο που αντλείται ανάμεσα στις χρονικές περιόδους που γίνεται η σύλληψη (capturing) των περιεχομένων από το ενεργό παράθυρο, χάνεται. Το πλεονέκτημα είναι ότι ένα στιγμιότυπο ενός παραθύρου που συνελήφθηκε περισσότερες από μία φορές, λόγω του τρόπου με τον οποίο γίνεται η ανάλυσή του ο οποίος περιγράφεται στο υποκεφάλαιο 4.3, δίνεται μεγαλύτερο βάρος σ' αυτό πράγμα που ισχύει δεδομένου ότι αν ένας χρήστης διαβάζει ένα κείμενο από ένα συγκεκριμένο παράθυρο για κάποια λεπτά, τότε αυτό περιγράφει καλύτερα το συγκεκριμένο του απ' ότι αν έκανε γρήγορη εναλλαγή (swapping) μεταξύ παραθύρων για μερικά δευτερόλεπτα.

2.3 Βιβλιοθήκη TextCaptureX

Η βιβλιοθήκη TextCaptureX [28] αποτελεί τον σκελετό του εργαλείου, αφού είναι υπεύθυνη για την εξαγωγή της πληροφορίας από τα παράθυρα του λειτουργικού συστήματος. Η μέθοδος που ακολουθεί ονομάζεται «hooking», γάντζωμα δηλαδή στα παράθυρα και εξαγωγή των περιεχομένων τους όταν ζητηθεί. Η βιβλιοθήκη είναι μέρος του προγράμματος ScreenScraperStudio [28], το οποίο χρησιμοποιείται για την δημιουργία της παραμέτρου strID, η οποία ορίζει για κάθε παράθυρο τα σημεία από τα οποία θα γίνει η εξαγωγή κειμένου. Κάθε φορά που καλείται η βιβλιοθήκη, πρέπει να δοθεί η παράμετρος strID για να γνωρίζει από ποιο σημείο του παραθύρου θα γίνει η εξαγωγή.

Το πώς συνδέεται το εργαλείο ContextExtractor με τη βιβλιοθήκη TextCaptureX και το Windows API, φαίνεται στο πιο κάτω σχήμα.



Σχήμα 2.1 - Αρχιτεκτονική εργαλείου ανάκτησης πληροφοριών ContextExtractor

Αρχικά, γίνεται κλήση των συναρτήσεων του Windows API για επιστροφή του «handle» (μοναδικό αναγνωριστικό του παραθύρου), του τίτλου και του «pid» (μοναδικά αναγνωριστικό της διεργασίας στην οποία ανήκει) του ενεργού παραθύρου. Ακολούθως, ελέγχεται αν το ενεργό παράθυρο ανήκει σε κάποια από τις υποστηριζόμενες εφαρμογές (Πίνακας 1) και αν ναι τότε καλείται η TextCaptureX με την παράμετρο strID για το συγκεκριμένο παράθυρο που είναι ήδη γνωστή και επιστρέφονται τα περιεχόμενά του.

Αφού επιστραφούν τα περιεχόμενα του παραθύρου, ακολουθεί η διαδικασία επεξεργασίας τους όπως περιγράφεται στα υποκεφάλαια 4.3 – 4.6.

2.4 Windows Search

Το Windows Search, είναι το προ-εγκατεστημένο πρόγραμμα το οποίο πραγματοποιεί αναζητήσεις σε έναν υπολογιστή που τρέχει λειτουργικό σύστημα Windows. Η εφαρμογή που υλοποιήθηκε, έχει τη δυνατότητα επικοινωνία με το Desktop Search των Windows και πραγματοποίηση αναζήτησης στα αρχεία του υπολογιστή.

Ο σκοπός του για την εφαρμογή, είναι να δίνει τη δυνατότητα εύρεσης εγγράφων στον υπολογιστή, τα οποία σχετίζονται με το περιβάλλον στο οποίο δουλεύει ένας χρήστης μια συγκεκριμένη στιγμή.

2.5 Lucene

Η Lucene [17] είναι μια βιβλιοθήκη που υποστηρίζει τη δημιουργία ευρετηρίων (indexes) για μια συλλογή εγγράφων. Στην περίπτωση του εργαλείου όμως, οι οντότητες τις οποίες καλούμαστε να ευρετηριάσουμε δεν αποτελούν στατικά έγγραφα αλλά δυναμικό περιεχόμενο. Παρόλα αυτά, η Lucene, μπορεί να χρησιμοποιηθεί για την ευρετηρίαση των ήδη ληφθέντων πληροφοριών όπως και έχει γίνει.

Οι δυνατότητες που προσφέρει η Lucene είναι πάρα πολλές. Στην παρούσα διπλωματική, η χρήση της έγινε μόνο για την παροχή αναζήτησης στο ιστορικό το οποίο κατέγραψε την αλληλεπίδραση του χρήστη.

Έτσι, ο χρήστης θα μπορεί να κάνει αναζήτηση όρων και το σύστημα θα επιστρέφει τα στιγμιότυπα τα οποία περιέχουν αυτούς τους όρους. Με αυτόν τον τρόπο, ένας χρήστης μπορεί εύκολα να ανατρέξει στα κείμενα τα οποία διάβαζε στο παρελθόν που αφορούσαν κάποιο θέμα (όπως ορίζεται με τις λέξεις που εισάγει στην αναζήτηση) και να τα διαβάσει ξανά.

Επίσης, ο χρήστης θα μπορεί να κάνει αναζητήσεις εύρους οι οποίες να του επιστρέφουν τα στιγμιότυπα που λήφθηκαν μέσα σε ένα χρονικό διάστημα. Με αυτόν τον τρόπο, ο χρήστης θα μπορεί να δει τι διάβαζε μια συγκεκριμένη στιγμή στο παρελθόν.

2.6 Μετρικές

Για την αξιολόγηση των αλγορίθμων ομαδοποίησης του συστήματος, οι οποίοι αναλύονται στο υποκεφάλαιο 4.6, χρησιμοποιήθηκαν κάποιες ευρέως χρησιμοποιούμενες μετρικές από το χώρο της Ανάκτησης Πληροφορίας. Οι μετρικές αυτές, εξηγούνται πιο κάτω.

Οι πρώτες δύο μετρικές, εφαρμόζονται για την αξιολόγηση ενός αλγορίθμου ο οποίος έχει σαν είσοδο μία συλλογή από αντικείμενα και σαν έξοδο ένα υποσύνολο της συλλογής αυτής η οποία υπολογίζεται εφαρμόζοντας τον αλγόριθμο στη συλλογή. Οι μετρικές εξετάζουν το κατά πόσο το υποσύνολο αυτό είναι το αναμενόμενο.

Οι επόμενες τρεις μετρικές, εφαρμόζονται για την αξιολόγηση ενός αλγορίθμου ομαδοποίησης. Ο όρος ομαδοποίηση αναλύεται στο υποκεφάλαιο 4.6. Οι μετρικές αυτές, εξετάζουν το κατά πόσο ο αλγόριθμος επέστρεψε τις ομάδες που έπρεπε να επιστρέψει αναθέτοντας τα σωστά αντικείμενα στην κάθε μια. Για σκοπούς καλύτερης διευκρίνησης, θα ορίσουμε ως κλάσεις τις ομάδες οι οποίες έχουν αναγνωρισθεί με μία αδιάψευστη μέθοδο ως οι ορθές ομάδες για μια συλλογή. Τις ομάδες τις οποίες δημιουργεί ο αλγόριθμος, θα τις αναφέρουμε ως ομάδες.

2.6.1 Precision

Ακρίβεια (precision) ονομάζουμε μία μετρική ακρίβειας η οποία μετρά το πόσο ακριβές ήταν το αποτέλεσμα που λήφθηκε. Μεγάλη ακρίβεια, σημαίνει ότι το υποσύνολο των αντικειμένων που έχει επιστραφεί, περιέχει μόνο αντικείμενα που όντως ανήκουν στο υποσύνολο, παρόλα αυτά όμως δεν σημαίνει πως ανακαλύψαμε όλα τα αντικείμενα που υπήρχαν στο σύνολο. Με λίγα λόγια, έχουμε ανακαλύψει μόνο αντικείμενα που έπρεπε, αλλά δεν τα έχουμε όλα.

2.6.2 Recall

Ανάκληση (recall) ονομάζουμε μία μετρική πληρότητας η οποία μετρά το πόσο ορθό ήταν το αποτέλεσμα που λήφθηκε. Μεγάλη ανάκληση, σημαίνει ότι το υποσύνολο των αντικειμένων που έχει επιστραφεί, περιέχει όλα τα αντικείμενα που όντως ανήκουν στο υποσύνολο, παρόλα αυτά όμως περιέχει και αντικείμενα τα οποία δεν έπρεπε γιατί δεν

ανήκουν στο υποσύνολο. Με λίγα λόγια, έχουμε ανακαλύψει όλα τα αντικείμενα που έπρεπε, αλλά έχουμε και κάποια που δεν έπρεπε.

2.6.3 Purity

Η μετρική αυτή, μετρά το βαθμό στον οποίο μια ομάδα που δημιουργήθηκε, περιέχει αντικείμενα τα οποία ανήκαν σε μία κλάση. Εάν η ομάδα περιέχει αντικείμενα από μία μόνο κλάση, τότε το purity (αγνότητα) της είναι 1. Παρόλα αυτά, η μετρική αυτή δεν εγγυάται ότι η ομάδα περιέχει όλα τα αντικείμενα της κλάσης.

2.6.4 Entropy

Η μετρική entropy, κοιτάζει πως οι διάφορες κλάσεις των αντικειμένων έχουν διαμοιραστεί σε κάθε ομάδα. Η μετρική αυτή παίρνει τιμές, από 0 μέχρι 1 με την τιμή 1 να σημαίνει πως τα αντικείμενα μιας κλάσης έχουν διαμοιραστεί στο μέγιστο βαθμό ανάμεσα στις ομάδες. Για παράδειγμα, εάν υπάρχουν A αντικείμενα σε αυτήν την κλάση και έχουν δημιουργηθεί X ομάδες, τότε κάθε ομάδα περιέχει A/X αντικείμενα.

2.6.5 F-score (ή F-measure)

Η τελευταία μετρική για την αξιολόγηση του αλγορίθμου ομαδοποίησης, ονομάζεται F-score. Η μετρική αυτή, χρησιμοποιεί τα αποτελέσματα δύο άλλων μετρικών που αναφέρθηκαν και πιο πάνω, των Precision και Recall. Σκοπός της, είναι να μετρήσει την ακρίβεια του αλγορίθμου ομαδοποίησης. Δηλαδή, αν ο αλγόριθμος δημιούργησε τον ίδιο αριθμό ομάδων με τις κλάσεις και αν κάθε ομάδα αντιστοιχεί σε μια κλάση περιλαμβάνοντας ακριβώς τα στοιχεία της κλάσης χωρίς τίποτα λιγότερο ή περισσότερο.

Κεφάλαιο 3

Ανάκτηση πληροφοριών

3.1	Ορισμός συγκειμένου (context) στα πλαίσια της εφαρμογής	16
3.2	Εξαγωγή ακατέργαστων πληροφοριών (raw data)	17

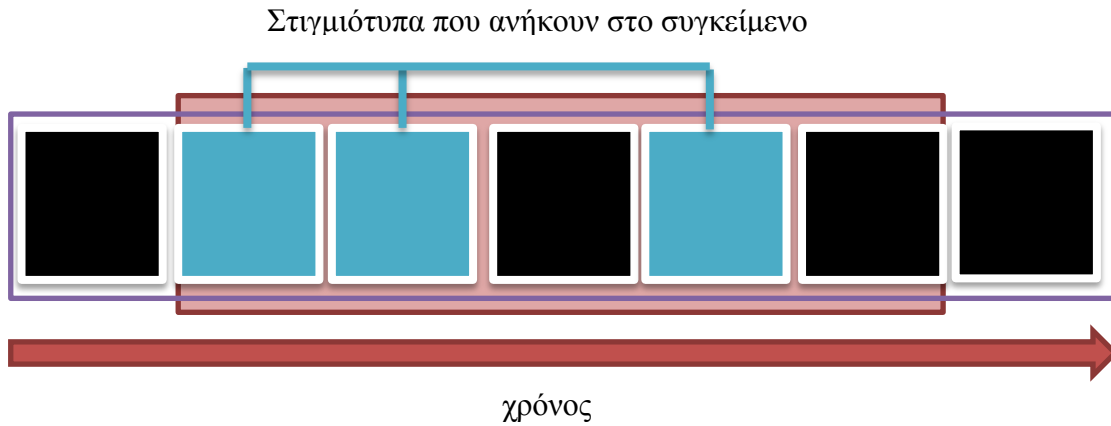
Στο κεφάλαιο αυτό, θα δοθεί ο ορισμός του συγκειμένου για την παρούσα διπλωματική και την υλοποίησή της. Επίσης, θα γίνει αναφορά στον τρόπο με τον οποίο εξάγεται η πληροφορία και αποθηκεύεται.

3.1 Ορισμός συγκειμένου (context) στα πλαίσια της εφαρμογής

Στα πλαίσια της διπλωματικής αυτής εργασίας, ορίζουμε το συγκείμενο χρήστη ως το μοντέλο που περιγράφει τη δραστηριότητα του χρήστη για μια δεδομένη στιγμή στην οποία λήφθηκε ένα στιγμιότυπο από το ενεργό παράθυρο του υπολογιστή του. Το συγκείμενο αποτελείται από μια σύνθετη δομή η οποία κτίζεται με βάση τα προηγούμενα στιγμιότυπα που έχουν ληφθεί. Το μοντέλο αυτό περιγράφεται βασιζόμενο στο XML σχήμα που υπάρχει στο Παράρτημα Α.

Το συγκείμενο χρήστη λοιπόν, αποτελείται από μια δομή, η οποία περιλαμβάνει ένα σύνολο από λέξεις οι οποίες είναι σε θέση να περιγράψουν τη δραστηριότητα του χρήστη και να χαρακτηρίσουν ως σύνολο μία τέτοια δραστηριότητα. Κάθε λέξη στο συγκείμενο, έχει διαφορετικό βάρος, πράγμα που σημαίνει ότι λέξεις με μεγαλύτερο

βάρος προσθέτουν μεγαλύτερο σημασιολογικό περιεχόμενο στο σύνολο σε σχέση με λέξεις με μικρότερο βάρος και ουσιαστικά ορίζουν το συγκεκριμένο.



Εικόνα 3.1 – Συλλογή από στιγμιότυπα το οποία απαρτίζουν το συγκεκριμένο

Όπως φαίνεται και στο πιο πάνω σχήμα, το συγκεκριμένο αποτελείται από ένα υποσύνολο από στιγμιότυπα. Τα στιγμιότυπα αυτά, συνδέονται με κάποιο τρόπο και έτσι συνολικά απαρτίζουν το συγκεκριμένο. Το συγκεκριμένο για μια δεδομένη στιγμή, αποτελείται από τους όρους που εμφανίζονται σε κάθε στιγμιότυπο το οποίο συνδέεται με το στιγμιότυπο της δεδομένης στιγμής.

3.2 Εξαγωγή ακατέργαστων πληροφοριών (raw data)

3.2.1 Ενεργό παράθυρο (active window)

Ενεργό παράθυρο σε ένα λειτουργικό σύστημα, ορίζεται ως το παράθυρο εκείνο του διαχειριστή παραθύρων, που έχει την εστίαση του πληκτρολογίου. Είναι το παράθυρο λοιπόν που βρίσκεται στο προσκήνιο και συνήθως το παράθυρο με το οποίο αλληλεπιδρά ένας χρήστης. Το ενεργό παράθυρο είναι πάντοτε μοναδικό.

3.2.2 Ανάκτηση ενεργού (active) παραθύρου

Το πρώτο βήμα που υλοποιεί το εργαλείο, είναι η ανάκτηση των πληροφοριών από το ενεργό παράθυρο. Αυτό γίνεται μέσω της βιβλιοθήκης TextCaptureX. Η πληροφορία

αυτή, στην αρχική της μορφή είναι ακατέργαστη (raw), γεγονός που σημαίνει ότι πιθανώς να εμπεριέχει πληροφορίες οι οποίες δεν είναι άμεσα διαθέσιμες αλλά μπορούν να γίνουν μετά από κάποια επεξεργασία. Για το λόγο αυτό, θα γίνει επεξεργασία της πληροφορίας και προσπάθεια για χαρακτηρισμό της σε σχέση με το περιεχόμενο αλλά και με τις υπόλοιπες πληροφορίες που έχουν ανακτηθεί μέσα στα ίδια χρονικά ή θεματικά πλαίσια.

Για παράδειγμα, ένας χρήστης διαβάζει ένα άρθρο στον κειμενογράφο του για την ιστορία του ποδοσφαίρου και επίσης ψάχνει στο διαδίκτυο για την ιστορία του ποδοσφαίρου. Αυτά τα δύο γεγονότα και κατ' επέκταση τα παράθυρα που εμφωλεύουν την πληροφορία που αντιστοιχεί σ' αυτά, συνδέονται θεματικά-σημασιολογικά και αν έχουν λάβει χώρα σε κοντινά χρονικά διαστήματα, μια παράμετρος που ορίζεται αργότερα, συνδέονται και χρονικά.

3.2.3 Επιλογή χρόνου ανάκτησης

Μία σημαντική παράμετρος του εργαλείου, εφόσον έχει επιλεγθεί η μέθοδος της εξαγωγής πληροφοριών από τα παράθυρα, είναι η παράμετρος που ορίζει το χρονικό διάστημα κατά το οποίο θα γίνεται η λήψη των περιεχομένων από το ενεργό παράθυρο. Η παράμετρος αυτή έχει οριστεί στα 5 δευτερόλεπτα.

Θεωρώντας ότι ο ανθρώπινος εγκέφαλος χρειάζεται 1 δευτερόλεπτο για να επεξεργαστεί την πληροφορία που βλέπει το μάτι και λαμβάνοντας υπόψη την μελέτη του J. Nielsen [18] όπου διαπίστωσε ότι στιγμιότυπα που διαρκούσαν λιγότερο από 4 δευτερόλεπτα δεν είχαν στην ουσία χρησιμοποιηθεί από τους χρήστες, επιλέξαμε την αμέσως μικρότερη τιμή των 5 δευτερολέπτων. Επιλέγοντας τη μικρότερη δυνατή τιμή στην οποία να λαμβάνονται τα στιγμιότυπα, διασφαλίζουμε ότι δεν χάνεται πληροφορία η οποία είχε σημασία στον χρήστη. Παρόλα αυτά, υπάρχει κίνδυνος για εξαγωγή μη χρήσιμης πληροφορίας.

Για να επιβεβαιώσουμε τον αριθμό αυτό, έγιναν διάφορα πειράματα με διαφορετικές τιμές και λαμβάνοντας υπόψη διάφορα πράγματα όπως τον όγκο των δεδομένων (σε περίπτωση πολύ μικρού αριθμού) και την πληροφορία που χάνεται (σε περίπτωση πολύ μεγάλου αριθμού), καταλήξαμε στην συγκεκριμένη τιμή. Αποτελέσματα από τα πειράματα παρουσιάζονται στο υποκεφάλαιο 6.2.

3.2.4 Ανάκτηση μόνο εμφανιζόμενων πληροφοριών

Ένα σημαντικό ερώτημα όσον αφορά την πληροφορία που εξάγεται, είναι τι ακριβώς εξάγει η εφαρμογή. Η απάντηση είναι πως η πληροφορία που εξάγεται, είναι μόνο η ορατή (visible) πληροφορία από το χρήστη [19] και όχι όλα τα περιεχόμενα ενός παραθύρου το οποία δεν είναι ορατά τη δεδομένη στιγμή αλλά μπορεί να γίνουν είτε με κύλιση είτε με κάποια άλλη ενέργεια του χρήστη ή του υπολογιστή. Αυτό προσθέτει ακόμη μεγαλύτερη ακρίβεια στο εργαλείο και επίσης κάνει την ανάλυση και αξιολόγηση ακόμη πιο εύκολη.

3.2.5 Ανάκτηση ειδικής πληροφορίας και σε μερικά εξειδικευμένα σημεία του παραθύρου. Ένα σημαντικό τέτοιο σημείο, είναι ο τίτλος

Μέχρι τώρα, αναφέραμε ότι εξάγεται το κείμενο που περιέχεται σε ένα παράθυρο και αυτό προστίθεται στο συγκεκριμένο χρήστη.

Αξίζει να αναφερθεί όμως, ότι σε παράθυρα από διαφορετικές εφαρμογές εφαρμόζεται διαφορετική μέθοδος για εξαγωγή των πληροφοριών. Έτσι, εκτός από το κυρίως κείμενο που αποτελεί τη βασική πηγή, σημαντική πληροφορία κρύβεται του παραθύρου που βρίσκεται στο λεγόμενο “titlebar”. Αυτή η πληροφορία, πιθανώς να ορίζει και τα περιεχόμενα του παραθύρου και για το λόγο αυτό δίνεται μεγάλο βάρος σε αυτήν.



Εικόνα 3.2 - Τυπικό παράθυρο φυλλομετρητή όπου εμφανίζονται α) ο τίτλος του παραθύρου, β) η διεύθυνση URL, γ) το κείμενο που περιέχει την πληροφορία και θα εξαχθεί

3.2.6 Αναπαράσταση στιγμιότυπου (snapshot)

Ένα στιγμιότυπο είναι μία αποτύπωση του περιεχομένου της οθόνης του χρήστη, για μια χρονική στιγμή. Από αυτό το στιγμιότυπο εξάγεται το κείμενο που παρουσιάζεται στο χρήστη. Κάθε στιγμιότυπο, έχει ορισμένα χαρακτηριστικά. Αυτά είναι η χρονοσφραγίδα του που δηλώνει ακριβώς τη στιγμή στην οποία συνελήφθηκε, το κείμενο το οποίο αναγνωρίστηκε, το μοναδικό αναγνωριστικό του παραθύρου από το οποίο λήφθηκε η πληροφορία (window handle), τον τίτλο του παραθύρου, το όνομα της διεργασίας στην οποία ανήκει το παράθυρο, τον αριθμό και τα ονόματα των υπόλοιπων διεργασιών που τρέχουν στο σύστημα. Το κείμενο που εξάγεται, ανάλογα με τα παράθυρο, μπορεί να αποτελείται από μία μόνο πηγή ή από πολλές. Για παράδειγμα το κείμενο από ένα παράθυρο επεξεργαστή κειμένου, αποτελείται μόνο από το σώμα του κειμένου που επεξεργάζεται ο χρήστης στο πρόγραμμα ενώ το κείμενο από μια εφαρμογή email αποτελείται από τον τίτλο του email που διαβάζει, τον αποστολέα, το σώμα του email.

Κάθε τέτοιο στιγμιότυπο, αμέσως μετά, μπαίνει σε μια κατάσταση επεξεργασίας από την εφαρμογή. Η επεξεργασία περιγράφεται αναλυτικότερα στα υποκεφάλαια 4.3 – 4.5.

3.2.7 Αποθήκευση στιγμιότυπων υπό μορφή κειμένου

Τελευταίο σημείο όσον αφορά την εξαγωγή του συγκεκριμένου, είναι η αποθήκευση των στιγμιότυπων σε ακατέργαστη μορφή σε αρχεία στο δίσκο. Αυτό γίνεται γιατί η πληροφορία αυτή είναι χρήσιμη κατά την παρουσίαση της στο χρήστη ούτως ώστε να είναι αναγνωρίσιμη απ' αυτόν.

Κάθε στιγμιότυπο αποθηκεύεται σε ξεχωριστό αρχείο με όνομα τη χρονοσφραγίδα η οποία δηλώνει τη στιγμή στην οποία λήφθηκαν τα περιεχόμενα από το παράθυρο. Αυτό γίνεται για να ξεχωρίζει το κάθε στιγμιότυπο.

Κεφάλαιο 4

Περιβάλλον υλοποίησης

4.1	Γενική περιγραφή συστήματος	22
4.2	Ανάλυση περιβάλλοντος εξαγωγής συγκειμένου	24
4.3	Ανάλυση - επεξεργασία στιγμιότυπου	24
4.4	Cosine Similarity	28
4.5	Μεταβολή συγκειμένου (context shift)	28
4.6	Ομαδοποίηση (clustering)	29

Στο κεφάλαιο αυτό, θα περιγραφεί η υλοποίηση του συστήματος. Η σειρά που ακολουθείται είναι και η σειρά με την οποία γίνεται η επεξεργασία ενός στιγμιότυπου μέχρι αυτό να μπορεί να χρησιμοποιηθεί για την ανακάλυψη μεταβολής συγκειμένου και για την ομαδοποίηση.

4.1 Γενική περιγραφή συστήματος

Η υλοποίηση των αλγορίθμων αλλά και του προγράμματος, έγινε σε γλώσσα προγραμματισμού C#. Η γλώσσα αυτή επιλέχθηκε για το λόγο ότι υποστηρίζει τον προγραμματισμό για Windows API το οποίο είναι υπεύθυνο για την επικοινωνία της εφαρμογής με τα παράθυρα για την εξαγωγή των πληροφοριών αλλά επίσης και για την επικοινωνία με τις τρέχουσες διεργασίες για αναγνώριση των παραθύρων που

χρησιμοποιούνται απ' αυτές. Η επικοινωνία αυτή είναι έμμεση, αφού χρησιμοποιείται ως μεσάζων λογισμικό η βιβλιοθήκη TextCaptureX η οποία αναλαμβάνει την επικοινωνία εφαρμογής και παραθύρων και συγκεκριμένα είναι υπεύθυνη να επιστρέφει στην εφαρμογή, το κείμενο που εξάγει από το ενεργό παράθυρο όποτε η εφαρμογή ζητά απ' αυτήν να το κάνει. Απ' εκεί και πέρα, η εφαρμογή είναι υπεύθυνη για την περαιτέρω επεξεργασία της πληροφορίας.

4.1.1 Επιλογή πληροφορίας προς ανάκτηση

Όπως αναφέρθηκε, η εξαγωγή πληροφορίας γίνεται από το ενεργό παράθυρο το οποίο ανήκει σε κάποια διεργασία. Για να γίνει επιτυχημένη εξαγωγή, το παράθυρο πρέπει να πληροί τους κανόνες σχεδίασης για παράθυρα Windows. Για να αποφευχθεί όμως η εξαγωγή πληροφορίας από άσχετα παράθυρα χωρίς ουσιώδες περιεχόμενο, η εφαρμογή δεν εξάγει πληροφορίες από όλα τα παράθυρα αλλά μόνο από παράθυρα που ανήκουν σε γνωστές εφαρμογές. Αυτές περιλαμβάνουν φυλλομετρητές, επεξεργαστές κειμένου και εξυπηρετητές ηλεκτρονικού ταχυδρομείου. Στον πιο κάτω πίνακα παρουσιάζονται οι εφαρμογές οι οποίες υποστηρίζονται από το εργαλείο καθώς επίσης και τα αναλυτικά στοιχεία τα οποία εξάγονται.

Κατηγορία	Όνομα εφαρμογής	Εξαγόμενα στοιχεία
Επεξεργαστές κειμένου	Notepad	Τίτλος εγγράφου, Περιεχόμενα
	Wordpad	
	Microsoft Word	
Pdf Viewers	Adobe Reader	
	Adobe Acrobat	
Φυλλομετρητές	Internet Explorer	Τίτλος ιστοσελίδας, διεύθυνση URL, περιεχόμενα
	Mozilla Firefox	
Εφαρμογές ηλεκτρονικού ταχυδρομείου	Mozilla Thunderbird	Τίτλος, αποστολέας, κείμενο μηνύματος
	Windows Live Mail	
	Microsoft Outlook	

Πίνακας 4.1 - Πίνακας με πληροφορίες για τα υποστηριζόμενα προγράμματα

Η επιλογή των εφαρμογών έγινε με βάση τις εφαρμογές οι οποίες μπορούν να έχουν πληροφορία υπό τη μορφή κειμένου σε βαθμό ικανοποιητικό ούτως ώστε να είναι εφικτή η επεξεργασία της και εξαγωγή αποτελεσμάτων. Παρόλα αυτά, η εφαρμογή μπορεί να επεκταθεί για υποστήριξη περισσότερων προγραμμάτων ή ακόμη να υποστηρίξει εξαγωγή από οποιοδήποτε παράθυρο. Για να γίνει αυτό, θα πρέπει να είναι γνωστό το strID το οποίο περιγράφει τη δομή του παραθύρου κάθε προγράμματος. Το strID μπορεί να ληφθεί μέσω του προγράμματος ScreenScraper Studio[28] το οποίο είναι το πρόγραμμα που περιλαμβάνει την βιβλιοθήκη TextCaptureX.

4.2 Ανάλυση περιβάλλοντος εξαγωγής συγκειμένου

Το εργαλείο που αναλαμβάνει την εξαγωγή πληροφοριών, είναι εντελώς απλό και διαφανές στο χρήστη. Ο χρήστης βλέπει απλά δύο κουμπιά, ένα για έναρξη της εξαγωγής και ένα για παύση. Όλο το σύστημα τρέχει στο παρασκήνιο χωρίς να επηρεάζει την εργασία του χρήστη.

4.3 Ανάλυση - επεξεργασία στιγμιότυπου

Όπως έχουμε αναφέρει, μετά τη λήψη ενός στιγμιότυπου, γίνεται άμεση επεξεργασία του για την μετατροπή του από την ακατέργαστη μορφή στην οποία βρίσκεται σε μία μορφή η οποία να παρέχει πλούσιες πληροφορίες που να σχετίζονται με αυτό. Στα πιο κάτω σημεία, περιγράφονται με σειρά τα βήματα που ακολουθούνται στην επεξεργασία αυτή.

4.3.1 Γλωσσολογική επεξεργασία

Γλωσσολογική επεξεργασία, ονομάζουμε την επεξεργασία ενός κειμένου με σκοπό την εξαγωγή των λέξεων που απαρτίζουν το κείμενο, υπό συγκεκριμένη μορφή.

Τα στάδια της επεξεργασίας αυτής, περιγράφονται πιο κάτω:

4.3.1.1 Λεξιλογική ανάλυση (tokenization)

Το πρώτο στάδιο της επεξεργασίας, περιλαμβάνει την λεξιλογική ανάλυση. Ένα κείμενο, είναι μία μεγάλη συμβολοσειρά από χαρακτήρες. Σκοπός της λεξιλογικής

ανάλυσης, είναι να διασπάσει το κείμενο σε λέξεις, απορρίπτοντας χαρακτήρες οι οποίοι δεν αποτελούν λέξεις όπως σημεία στίξης και ειδικούς χαρακτήρες. Έτσι, ενώ έχουμε σαν είσοδο τα εντελώς ακατέργαστα δεδομένα, παίρνουμε σαν έξοδο μία λίστα από πιθανές λέξεις που εμφανίζονται στο κείμενο.

4.3.1.2 Αποκλεισμός λέξεων (stop-word elimination)

Το δεύτερο στάδιο της επεξεργασίας, περιλαμβάνει τον αποκλεισμό λέξεων. Ο αποκλεισμός λέξεων έχει σκοπό να απαλείψει, από τη λίστα των πιθανών λέξεων που παίρνουμε από το προηγούμενο στάδιο, τις λέξεις οι οποίες θεωρούνται κοινά χρησιμοποιήσιμες λέξεις και δεν προσθέτουν σημασιολογικό περιεχόμενο στο κείμενο. Τέτοιες λέξεις είναι για παράδειγμα οι λέξεις “and”, “hello” κτλ. Οπότε, με αυτό το στάδιο, παίρνουμε μια καθαρή λίστα πλέον από τις λέξεις που ανακτήθηκαν από το αρχικό ακατέργαστο κείμενο.

4.3.1.3 Στελέχωση κειμένου (stemming)

Το τρίτο και τελευταίο στάδιο της γλωσσολογικής επεξεργασίας, αφορά την στελέχωση κειμένου. Σκοπός της είναι η αφαίρεση των καταλήξεων από τις λέξεις ούτως ώστε να γίνεται εύκολη η σύνδεση λέξεων με το ίδιο σημασιολογικό περιεχόμενο. Για παράδειγμα, αν στο αρχικό κείμενο υπήρχαν οι λέξεις “humanity” και “humans” οι οποίες ουσιαστικά είναι λέξεις με την ίδια σημασία, με την στελέχωση θα πάρουμε τις λέξεις “human” και “human”, τις οποίες εύκολα μπορούμε αργότερα να συνδέσουμε. Το τελικό αποτέλεσμα της γλωσσολογικής ανάλυσης, είναι μία λίστα από λέξεις από τις οποίες έχει αποκοπεί η κατάληξη και έμεινε η ρίζα. Ο αλγόριθμος που εφαρμόστηκε είναι ο Porter Stemmer αλγόριθμος [20] του Martin Porter.

4.3.2 Αναπαράσταση στο Vector-Space Model

Το Vector Space μοντέλο, είναι ένα μοντέλο αναπαράστασης κάθε εγγράφου ή στην περίπτωση της παρούσας διπλωματικής, κάθε στιγμιότυπου που ανακτάτε από την επιφάνεια εργασίας. Το στιγμιότυπο, αναπαρίσταται ως ένα διάνυσμα όπου κάθε διάσταση αντιστοιχεί σε κάποιον όρο-λέξη που έχει δημιουργηθεί από την γλωσσολογική επεξεργασία.

Κάθε όρος έχει κάποια τιμή στο διάνυσμα που ονομάζεται βάρος του όρου. Το βάρος αυτό υπολογίζεται με διάφορες τεχνικές. Η τεχνική που υλοποιήθηκε είναι η term frequency – inverse document frequency και περιγράφεται στο εδάφιο 4.3.3. Η τεχνική αυτή, είναι η πιο ευρέως χρησιμοποιούμενη τεχνική γι’ αυτό το μοντέλο και έτσι χρησιμοποιήθηκε στην εργασία αυτή.

Το βάρος κάθε όρου, είναι δεκαδικός αριθμός και κυμαίνεται μεταξύ 0 – 1 όπου μεγαλύτερη τιμή υποδεικνύει και μεγαλύτερο βάρος.

4.3.3 Μετρική tf-idf

Η τεχνική για υπολογισμό του βάρους κάθε λέξης η οποία υλοποιήθηκε, δίνεται από τον πιο κάτω τύπο:

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i$$

Ο όρος tf (term frequency) στον πιο πάνω τύπο, είναι η συχνότητα του όρου για ένα κείμενο. Ο τύπος για τον υπολογισμό του είναι:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

όπου $n_{i,j}$ είναι η συχνότητα εμφάνισης του όρου t_i στο d_j κείμενο. Η συχνότητα αυτή διαιρείται με τον αριθμό όλων των λέξεων στο κείμενο.

Ο όρος idf (inverse document frequency), είναι μια τιμή που δείχνει το πόσο σημαντικός είναι ο όρος για το κείμενο σε σχέση με όλα τα κείμενα που βρίσκονται στη συλλογή. Ο τύπος είναι:

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

Όπου $|D|$ είναι ο συνολικός αριθμός στιγμιότυπων στη συλλογή και $|\{j : t_i \in d_j\}|$ ο αριθμός των στιγμιότυπων στα οποία εμφανίζεται ο όρος.

Η χρησιμότητα αυτού του όρου, είναι να κατανέμει ίσα τα βάρη αποφεύγοντας την ανάθεση μεγάλου βάρους σε συχνά χρησιμοποιήσιμες λέξεις.

4.3.3.1 Εφαρμογή τεχνικής tf-idf σε δυναμικό περιεχόμενο

Η τεχνική αυτή, εφαρμόζεται παραδοσιακά σε στατικές συλλογές από αντικείμενα. Συνήθως εφαρμόζεται σε συλλογές από έγγραφα.

Η παρούσα υλοποίηση όμως, επιχειρεί μια εφαρμογή της τεχνικής σε περιεχόμενο το οποίο από τη φύση του είναι δυναμικό [1, 2]. Η τεχνική εφαρμόζεται ταυτόχρονα με την εξαγωγή των πληροφοριών και όχι αφού έχουν εξαχθεί όλες οι πληροφορίες. Αυτό όμως, δημιουργεί μια παραλλαγή στην τεχνική εφόσον πλέον στον τύπο του tf-idf, υπάρχει ένας μεταβλητός όρος, ο αριθμός των στιγμιότυπων στην συλλογή.

Ο αλγόριθμος που υλοποιήθηκε για να αντιμετωπίσει το πρόβλημα περιγράφεται πιο κάτω:

```
1 begin initialize q,  $\theta \leftarrow$  threshold  
2 do accept new s (νέο στιγμιότυπο)  
3   if q.Count >  $\theta$  (εάν η ουρά είναι γεμάτη)  
4     then q.dequeue() (διέγραψε το παλαιότερο)  
5   q.enqueue(s)  
6   v  $\leftarrow$  tfidf for q's contents (υπολόγισε το tfidf για)  
7 end (όλα τα στοιχεία της ουράς)
```

Αρχικά, η συλλογή των στιγμιότυπων είναι κενή εφόσον το πρόγραμμα θα τρέξει για πρώτη φορά. Καθώς λοιπόν το πρόγραμμα τρέχει και ανακτώνται πληροφορίες, νέα στιγμιότυπα προστίθενται στη συλλογή. Η είσοδος κάθε νέου στιγμιότυπου, επηρεάζει όχι μόνο τα βάρη του ιδίου αλλά και τα βάρη των υπολοίπων στιγμιότυπων. Αυτό συμβαίνει γιατί όπως έχουμε αναφέρει, στον τύπο που υπολογίζει το tf-idf για κάθε λέξη σε ένα στιγμιότυπο, ο όρος idf περιλαμβάνει τον αριθμό των εμφανίσεων της λέξης στα υπόλοιπα στιγμιότυπα. Οπότε για κάποιον όρο ενός παλαιότερου στιγμιότυπου, το νέο στιγμιότυπο μπορεί να έρθει και να επηρεάσει το βάρος του.

Για να γίνει αναγνώριση του συγκεκριμένου όμως για κάποιο δεδομένο χρόνο, χρειαζόμαστε εκτός από τα στιγμιότυπα που έχουν ανακτηθεί στο παρελθόν και τα στιγμιότυπα που θα ανακτηθούν στο μέλλον εφόσον και αυτά πιθανώς να περιέχουν στοιχεία που περιγράφουν το παρόν στιγμιότυπο.

Έτσι λοιπόν, για κάθε νέο στιγμιότυπο, πρέπει να υπολογίζεται ξανά ο όρος tf-idf και για τα προηγούμενα στιγμιότυπα. Όμως, αντί να υπολογίζεται για όλα τα προηγούμενα

στιγμιότυπα, γίνεται μία επιλογή των στιγμιότυπων για τα οποία θα υπολογιστεί. Η επιλογή αυτή βασίζεται στο χρονικό διάστημα το οποίο έχουν ληφθεί τα στιγμιότυπα από το παρόν στιγμιότυπο. Δηλαδή, λαμβάνονται υπόψη μόνο τα στιγμιότυπα που έχουν ληφθεί τα τελευταία X λεπτά. Η τιμή του X έχει προκαθορισθεί στα 30 λεπτά.

4.4 Cosine Similarity

Το cosine similarity, είναι μια μετρική της ομοιότητας μεταξύ δύο διανυσμάτων. Το αποτέλεσμα είναι μια τιμή μεταξύ 0 – 1 η οποία δηλώνει πόσο όμοια είναι τα διανύσματα. Όσο μεγαλύτερη είναι η τιμή τόσο πιο όμοια είναι με την τιμή 1 να δηλώνει πως το δύο διανύσματα είναι πανομοιότυπα.

Ο τύπος για υπολογισμό του cosine similarity δίνεται:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

όπου $A \cdot B$ είναι το εσωτερικό γινόμενο (dot product) των δύο διανυσμάτων και $\|A\| \|B\|$ το γινόμενο των Ευκλείδειων αποστάσεων τους.

4.5 Μεταβολή συγκειμένου (context shift)

Μεταβολή περιβάλλοντος, ονομάζουμε την αλλαγή του θέματος με το οποίο εργάζεται ένας χρήστης σε κάποιο άλλο. Η μεταβολή αυτή, μπορεί να αναπαρασταθεί σε ένα χρονοδιάγραμμα και να καταδεικνύει τις χρονικές στιγμές στις οποίες ο χρήστης έκανε μετάβαση σε κάποιο άλλο θέμα.

Για να γίνει ανίχνευση του context shift [11], χρειάζεται μια ποσοτική μετρική η οποία να περιγράφει δύο συνεχόμενα στιγμιότυπα. Εισηγούμαστε τη χρήση του cosine similarity το οποίο παρουσιάσαμε στο πιο πάνω σημείο. Ο λόγος για την επιλογή του συγκεκριμένου είναι ότι είναι μια μετρική που μπορεί να εφαρμοστεί στην περίπτωση μας και επίσης λαμβάνει υπόψη τα βάρη των όρων στα διανύσματα κανονικοποιώντας και τα μήκη των διανυσμάτων. Έχοντας αυτό, μπορούμε για κάθε νέο στιγμιότυπο, να

υπολογίζουμε την ομοιότητά του με το αμέσως προηγούμενο χρησιμοποιώντας το cosine similarity.

Μια μεγάλη τιμή ομοιότητας, δηλώνει πως το νέο στιγμιότυπο είναι αρκετά παρόμοιο (σημασιολογικά) έως πανομοιότυπο με το προηγούμενο. Αυτό σημαίνει πως το νέο στιγμιότυπο ανήκει στο ίδιο συγκείμενο με το προηγούμενο. Από την άλλη, μια μικρή τιμή, δηλώνει πως το νέο στιγμιότυπο είναι διαφορετικό από το προηγούμενο και πιθανώς ανήκει σε διαφορετικό συγκείμενο.

Το εργαλείο που υλοποιήθηκε, έχει τη δυνατότητα αναπαράστασης της μεταβολής του συγκειμένου σε σχέση με το χρόνο για μια περίοδο χρήσης του συστήματος.

4.6 Ομαδοποίηση (clustering)

4.6.1 Ορισμός

Ομαδοποίηση (clustering ή cluster analysis), ονομάζεται η ανάθεση μιας ομάδας από αντικείμενα σε υποομάδες ούτως ώστε τα αντικείμενα που βρίσκονται στην ίδια υποομάδα να έχουν κοινά γνωρίσματα ή χαρακτηριστικά βάσει κάποιων κανόνων.

Στην περίπτωση της εφαρμογής, η ομαδοποίηση γίνεται για να αναθέσουμε ένα στιγμιότυπο σε κάποια ομάδα με βάση τα περιεχόμενά του. Έτσι, κάθε ομάδα, θα περιέχει στιγμιότυπα τα οποία μοιάζουν μεταξύ τους ως προς το περιεχόμενο και έτσι για κάθε νέο στιγμιότυπο, η εφαρμογή θα είναι σε θέση να αποφασίσει σε πια ομάδα ανήκει, άρα και ποιο είναι το θέμα του. Αυτό σημαίνει ότι δυναμικά γίνεται μια αναγνώριση του συγκειμένου του χρήστη κάτι που είναι ο απώτερος στόχος της διπλωματικής αυτής.

4.6.2 Προκλήσεις ομαδοποίησης σε δυναμικό περιεχόμενο

Η ομαδοποίηση είναι μια τεχνική που εφαρμόζεται γενικά σε στατικό περιεχόμενο, δηλαδή περιεχόμενο το οποίο είναι διαθέσιμο εκ των προτέρων και δεν θα τροποποιηθεί αργότερα. Αυτό το περιεχόμενο είναι συνήθως έγγραφα κάποιας συλλογής (corpus). Στην περίπτωση μας όμως, το περιεχόμενο που έχουμε είναι δυναμικό εφόσον η αντίστοιχη οντότητα με τα έγγραφα είναι τα στιγμιότυπα τα οποία

εξάγονται σε πραγματικό χρόνο καθώς ο χρήστης χρησιμοποιεί τον υπολογιστή του. Η πρόκληση λοιπόν, ήταν να γίνει εφικτή η ομαδοποίηση ενός εισερχόμενου στιγμιότυπου σε πραγματικό χρόνο. Αυτό σημαίνει ότι οι ομάδες που δημιουργούνται από την ομαδοποίηση είναι επίσης δυναμικές αφού με την πάροδο του χρόνου νέες ομάδες θα δημιουργούνται και υπάρχουσες ομάδες θα ανανεώνονται με την προσθήκη νέων στιγμιότυπων. Για το λόγο αυτό, το διάνυσμα που περιγράφει το κέντρο μίας ομάδας, θα πρέπει να είναι ευέλικτο ούτως ώστε να μπορεί να δέχεται νέα στιγμιότυπα καθώς αυτά συλλέγονται.

4.6.3 Αναπαράσταση αντικειμένου

Τα αντικείμενα στα οποία εφαρμόζεται η ομαδοποίηση είναι τα στιγμιότυπα. Όπως έχει αναφερθεί, κάθε στιγμιότυπο αποτελείται από ένα διάνυσμα που περιέχει τις λέξεις που εμφανίζονται σ' αυτό και το βάρος της κάθε λέξης για το στιγμιότυπο. Αυτό το διάνυσμα, λαμβάνει υπόψη ο αλγόριθμος ομαδοποίησης που αναλαμβάνει την τοποθέτηση των στιγμιότυπων σε ομάδες. Στο διάνυσμα αυτό, όπως έχει αναφερθεί, περιλαμβάνονται εκτός από το κείμενο που έχει εξαχθεί και ο τίτλος του παραθύρου. Για να δοθεί περισσότερο βάρος στον τίτλο, οι λέξεις του τίτλου εισάγονται στο διάνυσμα έχοντας μεγαλύτερο ποσοστιαίο βάρος από τις λέξεις που βρίσκονται στο κείμενο.

4.6.4 Μετρική απόστασης δύο σημείων

Μία σημαντική παράμετρος για έναν αλγόριθμο ομαδοποίησης, είναι ο μετρητής απόστασης δύο αντικειμένων. Τα αντικείμενα που λαμβάνουν μέρος στην ομαδοποίηση, μπορούμε να τα θεωρήσουμε ως σημεία στον δισδιάστατο χώρο.

Απόσταση δύο σημείων στο χώρο, ονομάζουμε το πόσο μακριά είναι το ένα σημείο από το άλλο. Για να δηλώσουμε το μέτρο της απόστασης χρησιμοποιούμε κάποια μονάδα μέτρησης. Στην περίπτωση της ομαδοποίησης, η απόσταση δύο αντικειμένων, είναι η ομοιότητα (similarity) των δύο αντικειμένων και το μέτρο της απόστασης είναι το αποτέλεσμα του αλγορίθμου ο οποίος υπολογίζει την απόσταση των δύο αντικειμένων με βάσει κάποια χαρακτηριστικά που έχουν τα αντικείμενα.

Για την παρούσα διπλωματική, ο αλγόριθμος που έχει υλοποιηθεί στηρίζεται στο cosine similarity μεταξύ δύο αντικειμένων και χρησιμοποιεί συμμετρικές αποστάσεις με την έννοια ότι ένα αντικείμενο A απέχει από ένα αντικείμενο B όσο απέχει και το B από το A.

4.6.5 Αναπαράσταση ομάδας

Μια ομάδα από στιγμιότυπα χαρακτηρίζεται από ένα κέντρο (centroid) το οποίο περιλαμβάνει την τομή όλων των επί μέρους στιγμιότυπων. Όπως έχουμε αναφέρει στο εδάφιο 4.6.2, ένα στιγμιότυπο χαρακτηρίζεται από ένα διάνυσμα που περιλαμβάνει τις λέξεις που ανήκουν σ' αυτό και το βάρος της κάθε λέξης όπως αυτό υπολογίζεται από τον αλγόριθμο του tf-idf. Το κέντρο λοιπόν μιας ομάδας, είναι ένα διάνυσμα το οποίο περιλαμβάνει την τομή των διανυσμάτων των στιγμιότυπων που το απαρτίζουν. Το διάνυσμα έχει την ίδια μορφή με αυτήν ενός στιγμιότυπου, δηλαδή περιλαμβάνει πλειάδες της μορφής $\langle t, w \rangle$ όπου t είναι η λέξη και w το βάρος της.

4.6.6 K-means

Ο αλγόριθμος k-means [26], είναι ένας αλγόριθμος για ομαδοποίηση ο οποίος προσπαθεί να διαχωρίσει τα αντικείμενα σε ομάδες τοποθετώντας κάθε αντικείμενο στην ομάδα που είναι πιο κοντά σ' αυτό. Για να το πετύχει αυτό, υπολογίζει το κέντρο της κάθε ομάδας και ακολούθως ελέγχει αν το αντικείμενο είναι κοντά σ' αυτό το κέντρο.

Για το υπολογισμό του κέντρου, ο αλγόριθμος λαμβάνει υπόψη όλα τα αντικείμενα που ανήκουν ήδη στην ομάδα. Συγκεκριμένα, ο αλγόριθμος χρησιμοποιεί το διάνυσμα του αντικειμένου το οποίο υπολογίστηκε πιο πάνω μέσω της τεχνικής του cosine similarity. Αν λοιπόν το αντικείμενο είναι αρκετά όμοιο με το κέντρο, τότε ανήκει σ' αυτό.

Η βασική λειτουργία του αλγορίθμου είναι:

1. Επέλεξε K σημεία ως τα αρχικά κέντρα (centroids)
2. Ανάθεσε όλα τα σημεία στο κοντινότερο κέντρο
3. Υπολόγισε το κέντρο της κάθε ομάδας ξανά
4. Επανάλαβε τα σημεία 2-3 μέχρι τα κέντρα να μην αλλάζουν

4.6.7 Incremental k-means

Ο αλγόριθμος αυτός, αποτελεί μια παραλλαγή του k-means αλγορίθμου. Σκοπός του είναι να αντιμετωπίσει τον χειρισμό δυναμικών δεδομένων όπως αυτά που έχουμε στην περίπτωση μας.

Για τους σκοπούς της διπλωματικής εργασίας, υλοποιήθηκε μια παραλλαγή αυτού του αλγορίθμου η οποία να χειρίζεται κατάλληλα τα δεδομένα.

Ο αλγόριθμος που υλοποιήθηκε, παρουσιάζεται πιο κάτω:

```
1  begin initialize  $\eta, \theta \leftarrow \text{threshold}$ 
2     $\mu_1 \leftarrow x$ 
3    do accept new  $x$ 
4       $j \leftarrow \arg \min_j \|x - \mu_j\|$ 
5      if  $\|x - \mu_j\| < 0$ 
6        then  $\mu_j \leftarrow \mu_j + \eta x$ 
7        else add new  $\mu \leftarrow x$ 
8           $\mu \leftarrow \frac{\mu}{\|\mu\|}$ 
9      until no more patterns
10   return  $\mu_1, \mu_2, \mu_3, \dots$ 
11 end
```

Για τον υπολογισμό της απόστασης ενός στιγμιότυπου από το κέντρο μιας ομάδας, εφαρμόζεται ο εξής αλγόριθμος:

```
1  begin initialize list,  $s \leftarrow \text{snapshot}$ 
2    foreach cluster  $c$ 
3      list  $\leftarrow c \cup s$ 
4      foreach term  $t$  in list
5        if  $s$  contains  $t$ 
6          then  $w_s \leftarrow w_s + t$ 
7          if  $c$  contains  $t$ 
8            then  $w_c \leftarrow w_c + t$ 
9          sim  $\leftarrow \min(\text{sim}, (\text{cosinesimilarity } w_s, w_c))$ 
10   return sim
11 end
```

Επαναλαμβάνοντας τον πιο πάνω αλγόριθμο για όλες τις ομάδες, βρίσκουμε την κοντινότερη ομάδα στο νέο στιγμιότυπο. Έτσι, αν η απόσταση είναι μεγαλύτερη από το threshold, το αντικείμενο προστίθεται στην ομάδα αυτή αλλιώς δημιουργείται μία νέα ομάδα με το νέο στιγμιότυπο.

Ένα ενδιαφέρον σημείο που προκύπτει με την συγκεκριμένη υλοποίηση, είναι πως ένα νέο στιγμιότυπο προστίθεται σε μία υπάρχουσα ομάδα. Για να γίνει αυτό, ακολουθείται ο πιο κάτω αλγόριθμος. Ο αλγόριθμος, πρέπει να λάβει υπόψη όλα τα υπάρχοντα στιγμιότυπα στην ομάδα και να ανανεώσει το διάνυσμα του κέντρου δίνοντας το ανάλογο βάρος στο διάνυσμα του νέου στιγμιότυπου, χωρίς να υποτιμήσει τα υπάρχοντα στιγμιότυπα.

```

1 begin
2   list  $\leftarrow \mu_j \cup s$ 
3   foreach t in list
4     if c contains t
5       then if s contains t
6         then  $w_{i,j} \leftarrow ((w_{i,j} * \|s\|) + w_i) / (\|s\| + 1)$ 
7         else  $w_{i,j} \leftarrow (w_{i,j} * \|s\|) / (\|s\| + 1)$ 
8       else  $w_{i,j} \leftarrow w_i / (\|s\| + 1)$ 
9   end

```

Κεφάλαιο 5

Χαρακτηριστικά συστήματος

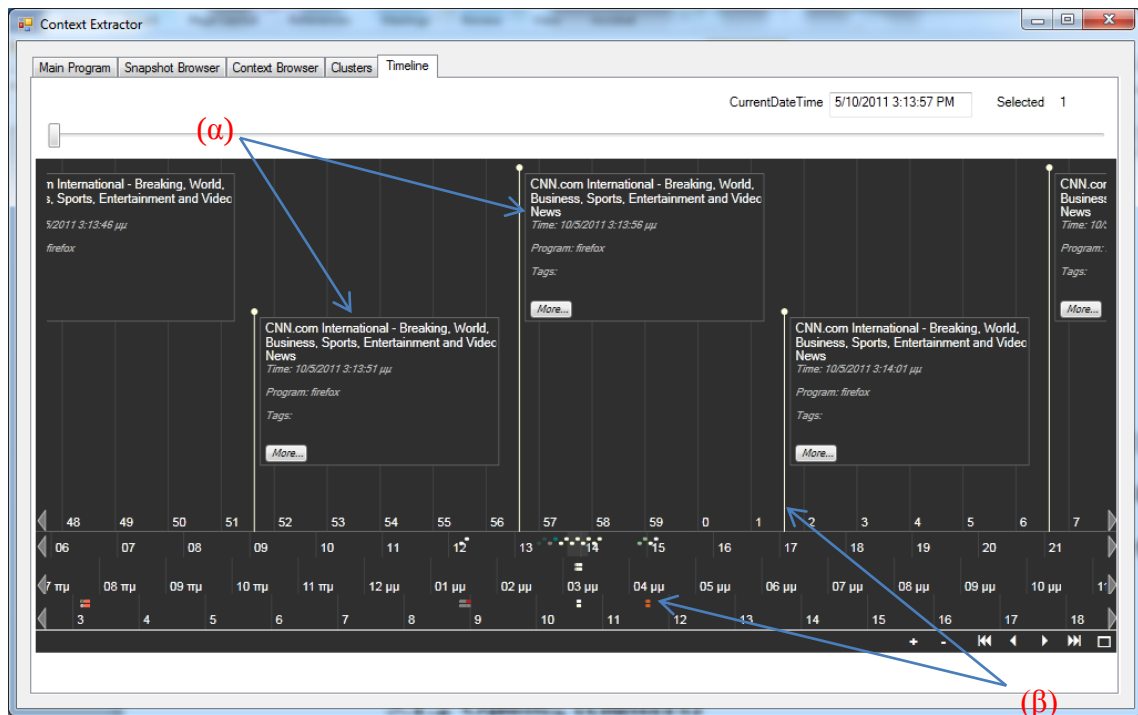
5.1	Ανάλυση visualization	34
5.2	Ευρετηρίαση με Lucene	36

Στο κεφάλαιο αυτό, θα παρουσιαστεί το γραφικό περιβάλλον του συστήματος και οι δυνατότητες που προσφέρει μέσα απ' αυτό.

5.1 Ανάλυση visualization

5.1.1 Χρονοδιάγραμμα (timeline)

Η εφαρμογή που υλοποιήθηκε, δίνει στο χρήστη τη δυνατότητα προβολής του ιστορικού των στιγμιότυπων που έχουν ληφθεί από το σύστημα, σε ένα χρονοδιάγραμμα. Ο χρήστης, μπορεί να μετακινηθεί σ' αυτό, βλέποντας τα περιεχόμενα των στιγμιότυπων για κάθε χρονική στιγμή που επιλέξει.



Εικόνα 5.1 - Εικόνα με χρονοδιάγραμμα όπου εμφανίζονται τα στιγμιότυπα (α). Τα χρώματα υποδηλώνουν τις ομάδες (β).

Στο πιο πάνω παράθυρο, εμφανίζεται το χρονοδιάγραμμα με στιγμιότυπα που λήφθηκαν. Οι άξονες κάτω, ρυθμίζουν την ημερομηνία και ώρα και πάνω φαίνονται τα στιγμιότυπα. Σ' αυτά εμφανίζεται ο τίτλος και τα περιεχόμενα.

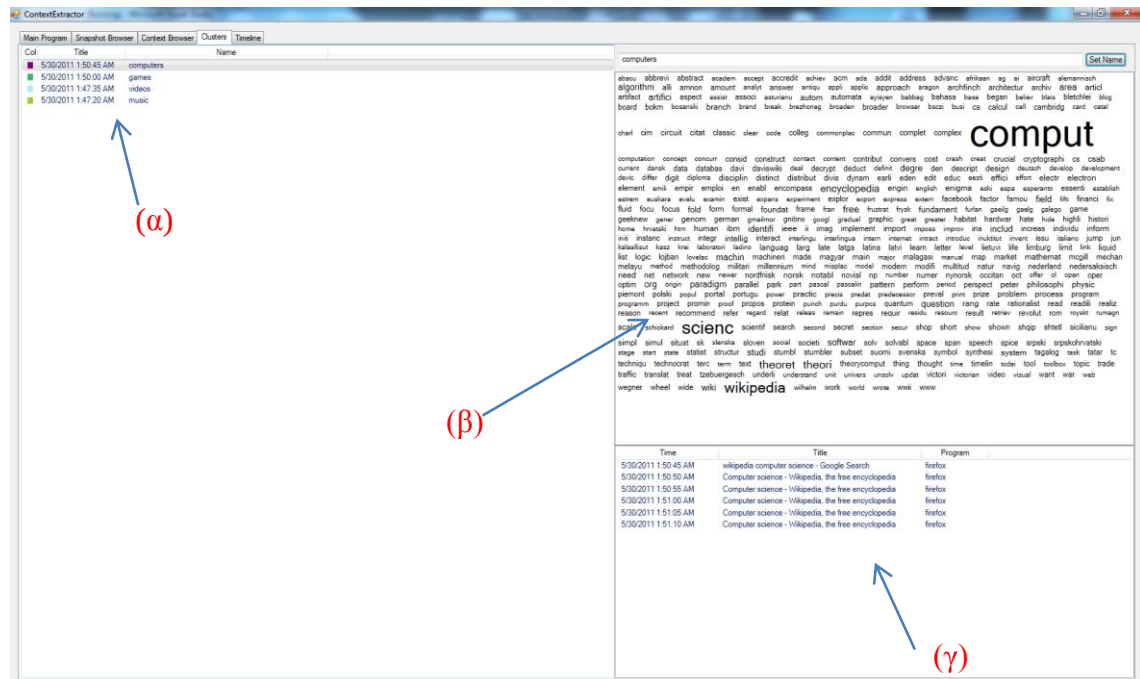
Το κάθε στιγμιότυπο, χρωματίζεται με διαφορετικό χρώμα ανάλογα με την ομάδα στην οποία ανήκει βάσει της ομαδοποίησης που έχει γίνει. Έτσι ο χρήστης μπορεί να ξεχωρίσει τα στιγμιότυπα που ανήκουν στην ίδια ομάδα.

5.1.2 Ομάδες (clusters)

Ο χρήστης έχει επίσης τη δυνατότητα να δει τη λίστα με τις ομάδες που αναγνωρίστηκαν καθώς επίσης και τα στιγμιότυπα που περιέχει η κάθε ομάδα. Οι ομάδες, εμφανίζονται υπό τη μορφή ενός tag cloud, όπου η κάθε λέξη μορφοποιείται ως προς το μέγεθος ανάλογα με το βάρος που έχει για την ομάδα. Έτσι, ο χρήστης μπορεί να αναγνωρίσει σε ποιο θέμα αναφέρεται η κάθε ομάδα.

Στην πιο κάτω εικόνα φαίνεται αριστερά η λίστα με τις ομάδες που έχουν αναγνωριστεί και δεξιά το tag cloud με τα περιεχόμενα του κέντρου τις επιλεγμένης ομάδας και από

κάτω τα στιγμιότυπα που απαρτίζουν την ομάδα. Ο χρήστης έχει τη δυνατότητα να ονομάσει την κάθε ομάδα για να είναι εύκολα αναγνωρίσιμη σ' αυτόν.

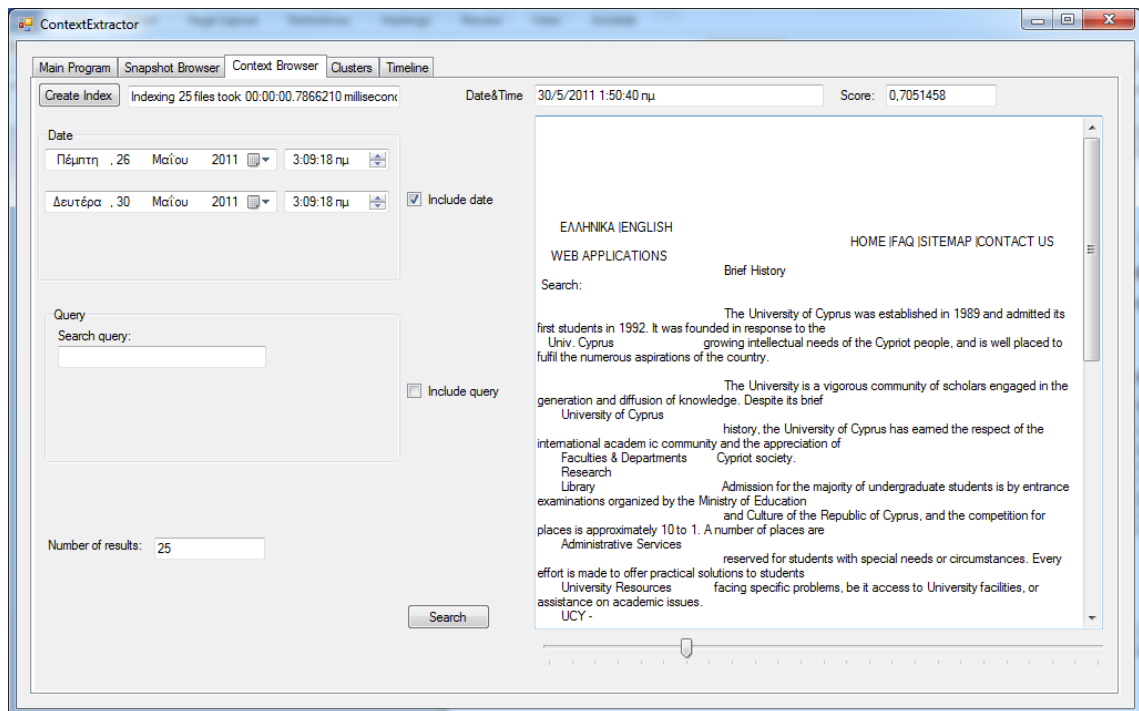


Εικόνα 5.2 - Εικόνα όπου εμφανίζονται οι ομάδες (α) και τα περιεχόμενά τους (γ). Στο (β) φαίνεται το tag cloud με τους όρους της ομάδας.

5.2 Ευρετηρίαση με Lucene

Μια ακόμα δυνατότητα που δίνεται στον χρήστη, είναι η πραγματοποίηση αναζητήσεων στο περιεχόμενο που έχει συλλεχθεί. Μέσω της Lucene, ο χρήστης μπορεί να αναζητήσει όρους ή να πραγματοποιήσει αναζήτηση εύρους για μια χρονική περίοδο.

Πιο κάτω βρίσκεται μια εικόνα από το παράθυρο που επιτρέπει στο χρήστη την αναζήτηση. Ο χρήστης ορίζει τις παραμέτρους και ακολούθως επιστρέφονται τα αποτελέσματα τα οποία μπορεί να δει μαζί με τη βαθμολογία που έβαλε στο κάθε ένα η Lucene ως προς το πόσο καλό ήταν το αποτέλεσμα για τη συγκεκριμένη αναζήτηση.



Εικόνα 5.3 - Εικόνα που δείχνει τη λειτουργία της Lucene για πραγματοποίηση αναζητήσεων στα στιγμιότυπα

Κεφάλαιο 6

Αξιολόγηση

6.1	Περιγραφή πειραμάτων	38
6.2	Αξιολόγηση εξαγωγής πληροφοριών	39
6.3	Αξιολόγηση αλγορίθμου ομαδοποίησης	40
6.4	Γενική αξιολόγηση συστήματος	46

Στο κεφάλαιο αυτό περιγράφεται η αξιολόγηση του συστήματος και του αλγορίθμου ομαδοποίησης. Αρχικά περιγράφονται τα πειράματα και τέλος δίνονται τα αποτελέσματα και οι γραφικές.

6.1 Περιγραφή πειραμάτων

Η αξιολόγηση του συστήματος, χωρίζεται σε τρία μέρη. Στο πρώτο μέρος, εξετάζεται το κομμάτι που είναι υπεύθυνο για την εξαγωγή των αδόμητων πληροφοριών από τα παράθυρα. Εξετάζεται συγκεκριμένα το κάθε πότε πρέπει να γίνεται λήψη των πληροφοριών και πως επηρεάζει η παράμετρος αυτή την μετέπειτα λειτουργία του προγράμματος. Επίσης εξετάζεται η μεταβολή συγκεκριμένου και πόσο αυτό είναι σημαντικό για την ανάλυση της εργασίας του χρήστη καθώς επίσης και για σκοπούς ανάλυσης της συμπεριφοράς του.

Στο δεύτερο μέρος, γίνεται αξιολόγηση του αλγορίθμου ομαδοποίησης ο οποίος υλοποιήθηκε στα πλαίσια της διπλωματικής για την αντιμετώπιση των δυναμικών στιγμιότυπων και την ανάθεση τους σε ομάδες.

Στο τρίτο μέρος, γίνεται γενική αξιολόγηση του συστήματος και της χρησιμότητάς του. Παρουσιάζονται επίσης τα πειραματικά αποτελέσματα από την εφαρμογή του προγράμματος σε πραγματικό περιβάλλον με πραγματικούς χρήστες. Για το σκοπό αυτό, το πρόγραμμα δόθηκε σε διάφορους χρήστες, από τους οποίους ζητήθηκε να τρέξουν το πρόγραμμα κατά τη διάρκεια μιας τυπικής χρήσης του υπολογιστή.

6.2 Αξιολόγηση εξαγωγής πληροφοριών

Στο σημείο αυτό, ελέγχθηκε η ορθότητα και αποτελεσματικότητα της επιλεγμένης μεθόδου για εξαγωγή πληροφοριών από το συγκείμενο του χρήστη. Για να γίνει αυτό, δημιουργήθηκαν εικονικές περιπτώσεις χρήσης (use cases) οι οποίες να αντικατοπτρίζουν διάφορα σενάρια χρήσης από διαφορετικές ομάδες χρηστών. Τα σενάρια αυτά έπρεπε να είναι ελεγχόμενα για να γίνει ανάλυση του κάθε στιγμιότυπου.

Ειδικά, ελέγχθηκε κατά πόσον η επιλεγμένη μέθοδος με τον τρόπο που έχει υλοποιηθεί, ανακτά τις ορθές πληροφορίες για διαφορετικές τιμές της παραμέτρου η οποία καθορίζει το διάστημα μεταξύ των ανακτήσεων.

Όπως αναφέρθηκε στο εδάφιο 3.2.3, ο χρόνος ανάκτησης ορίστηκε στα 5 δευτερόλεπτα. Κύριο μέλημα ήταν η διασφάλιση ότι δεν χάνονται σημαντικές για το χρήστη πληροφορίες.

Το συμπέρασμα από την αξιολόγηση αυτού του κομματιού είναι ότι ανακτώνται όντως οι ορθές πληροφορίες. Παρόλα αυτά, έγινε μια πιο λεπτομερής ανάλυση του τι ακριβώς είναι αυτά που βλέπει ο χρήστης στην οθόνη. Μερικά σημεία που ανιχνεύτηκαν είναι:

Παρόλο που το πρόγραμμα ανακτά ακριβώς ό,τι βλέπει ο χρήστης στην οθόνη του, χωρίς οτιδήποτε περισσότερο ή οτιδήποτε λιγότερο, πολλές φορές περιλαμβάνει θόρυβο που αργότερα θα αποδειχθεί πως δεν είναι επιθυμητός.

Η συγκεκριμένη υλοποίηση, θεωρεί ότι ο χρήστης εργάζεται κατά το μεγαλύτερο μέρος του συνολικού χρόνου, με παράθυρα τα οποία περιέχουν πληροφορία υπό τη μορφή

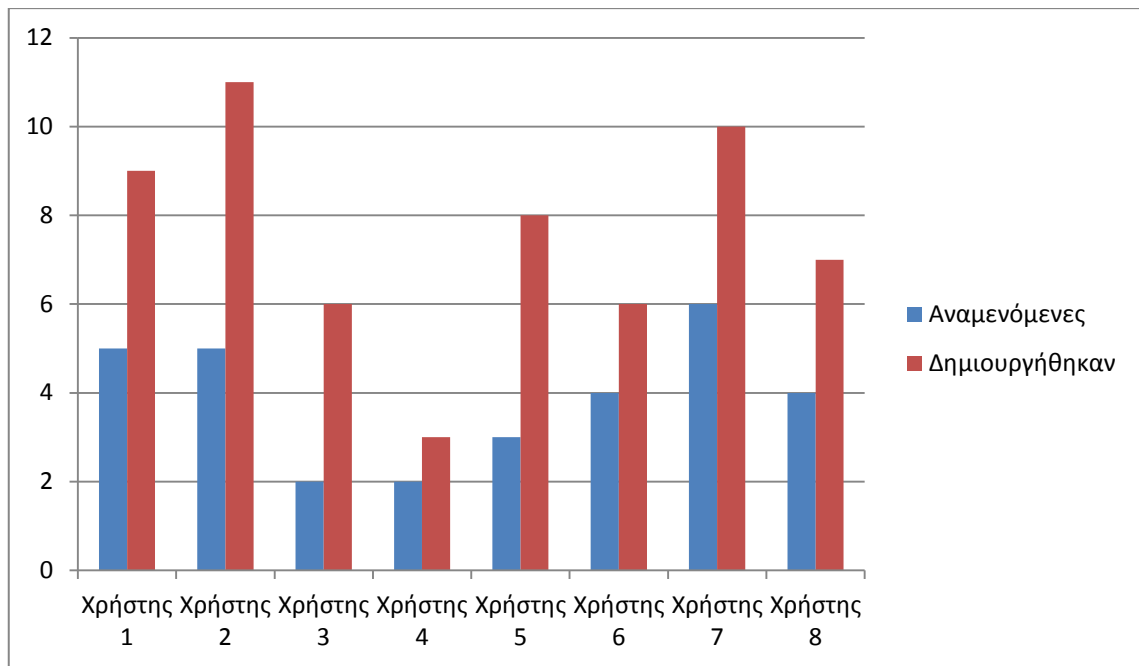
κειμένου η οποία να είναι ανιχνεύσιμη. Αυτό σημαίνει ότι για το χρονικό διάστημα στο οποίο ο χρήστης δουλεύει σε κάτι άλλο, το πρόγραμμα αδυνατεί να ορίσει το περιβάλλον εργασίας.

Όσον αφορά την παράμετρο t η οποία ορίζει το διάστημα στο οποίο γίνονται οι λήψεις των πληροφοριών, παρατηρήθηκε πως αυτό συνδέεται άμεσα με τον τρόπο εργασίας του χρήστη. Παρόλα αυτά, έπρεπε να επιλεγεί μία τιμή ούτε πολύ μικρή ούτε πολύ μεγάλη ώστε να αποφευχθούν τα πολλά λανθασμένα στιγμιότυπα αλλά ούτε και πολύ μεγάλη για να μην υπάρχουν στιγμιότυπα τα οποία να μην ανακτώνται.

6.3 Αξιολόγηση αλγορίθμου ομαδοποίησης

Για την αξιολόγηση του αλγορίθμου ομαδοποίησης [30], πραγματοποιήσαμε κάποια πειράματα με πραγματικούς χρήστες. Το πρόγραμμα δόθηκε σε οκτώ χρήστες από τους οποίους ζητήθηκε να τρέξουν το πρόγραμμα και ακολούθως να αλληλεπιδράσουν με τον υπολογιστή όπως σε μια συνηθισμένη συνεδρία.

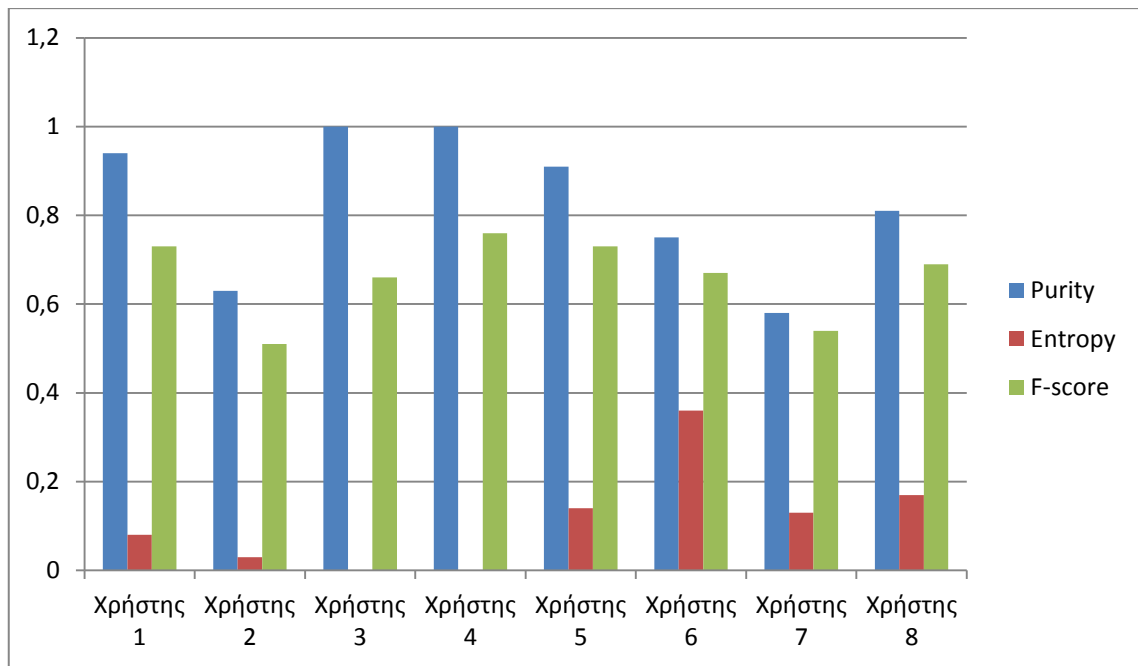
Μετά το πέρας της συνεδρίας, ζητήθηκε από τους χρήστες να συμμετάσχουν στην αξιολόγηση της ομαδοποίησης. Συγκεκριμένα οι χρήστες ρωτήθηκαν για το πόσες ομάδες αναγνώρισαν οι ίδιοι για τη συνεδρία τους και τέλος ζητήθηκε να συμπληρώσουν ένα ερωτηματολόγιο. Στον πιο κάτω πίνακα φαίνονται οι απαντήσεις των χρηστών καθώς και οι ομάδες που δημιουργήθηκαν για τον κάθε χρήστη.



Σχήμα 6.1 - Ομάδες που αναμενόταν/δημιουργήθηκαν για τους χρήστες

Πιο κάτω παρουσιάζονται αναλυτικά για τον κάθε χρήστη, το πώς διαμοιράστηκαν τα στιγμιότυπα των διαφόρων κλάσεων στις ομάδες που δημιουργήθηκαν.

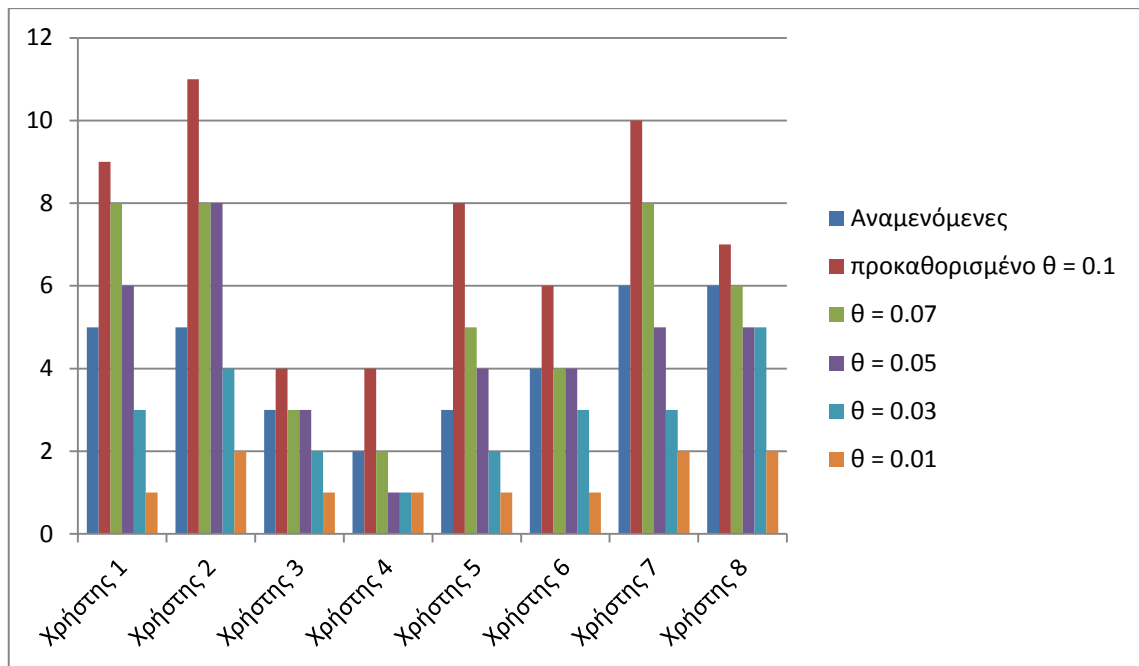
Για την αξιολόγηση του αλγορίθμου, χρησιμοποιήθηκαν οι μετρικές που περιγράφονται στο υποκεφάλαιο 2.6. Οι τρεις αυτές μετρικές εφαρμόστηκαν για τα στιγμιότυπα του κάθε χρήστη και τα αποτελέσματα παρουσιάζονται στον πιο κάτω πίνακα.



Σχήμα 6.2 - Οι μετρικές για το προκαθορισμένο κατώτατο όριο ($\theta=0.1$)

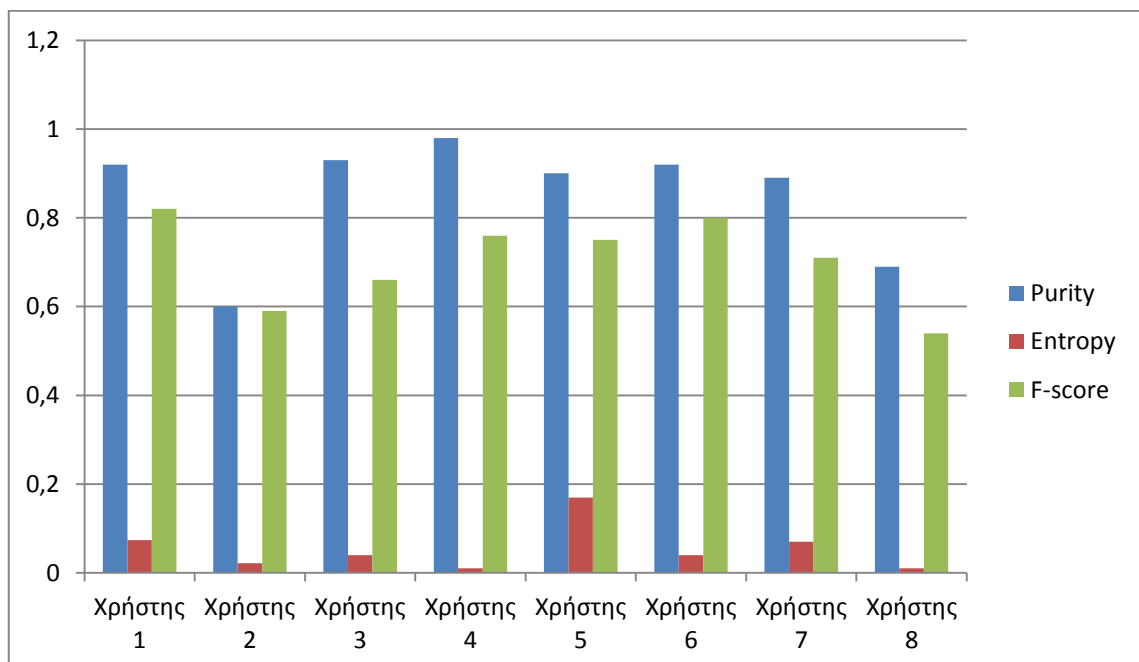
Από τους πιο πάνω πίνακες, φαίνεται πως ο αλγόριθμος δημιουργούσε περισσότερες ομάδες απ' ότι ήταν αναμενόμενο από τους χρήστες. Για το λόγο αυτό, ξανατρέξαμε τα στιγμιότυπα που είχαν συλλεχθεί, σε ένα περιβάλλον προσομοίωσης, τροποποιώντας το κατώτατο όριο (threshold) του αλγορίθμου ομαδοποίησης. Αυτό ήταν εφικτό, εφόσον φυλάγονταν τα στιγμιότυπα υπό αδόμητη μορφή. Πιο κάτω παρουσιάζονται οι πίνακες με τα αποτελέσματα.

Ο πιο κάτω πίνακας, δείχνει τις αναμενόμενες ομάδες και τις ομάδες που δημιουργήθηκαν για κάθε τιμή του κατώτατου ορίου, για τους χρήστες.

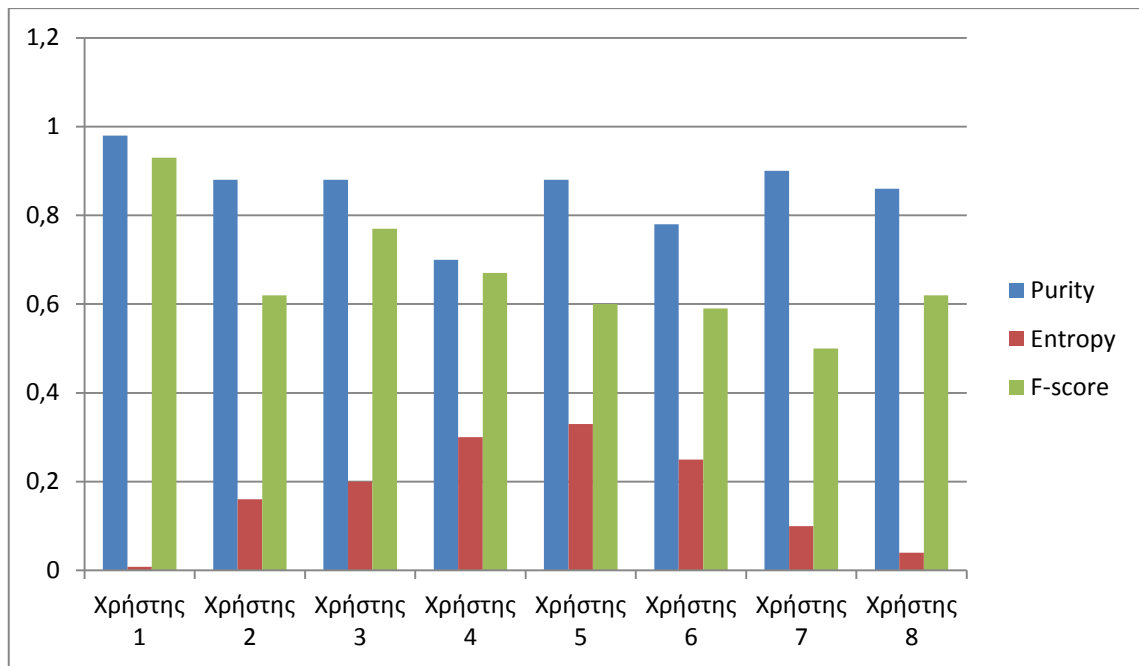


Σχήμα 6.3 - Αναμενόμενες/Ομάδες που δημιουργήθηκαν για όλα τα πειράματα

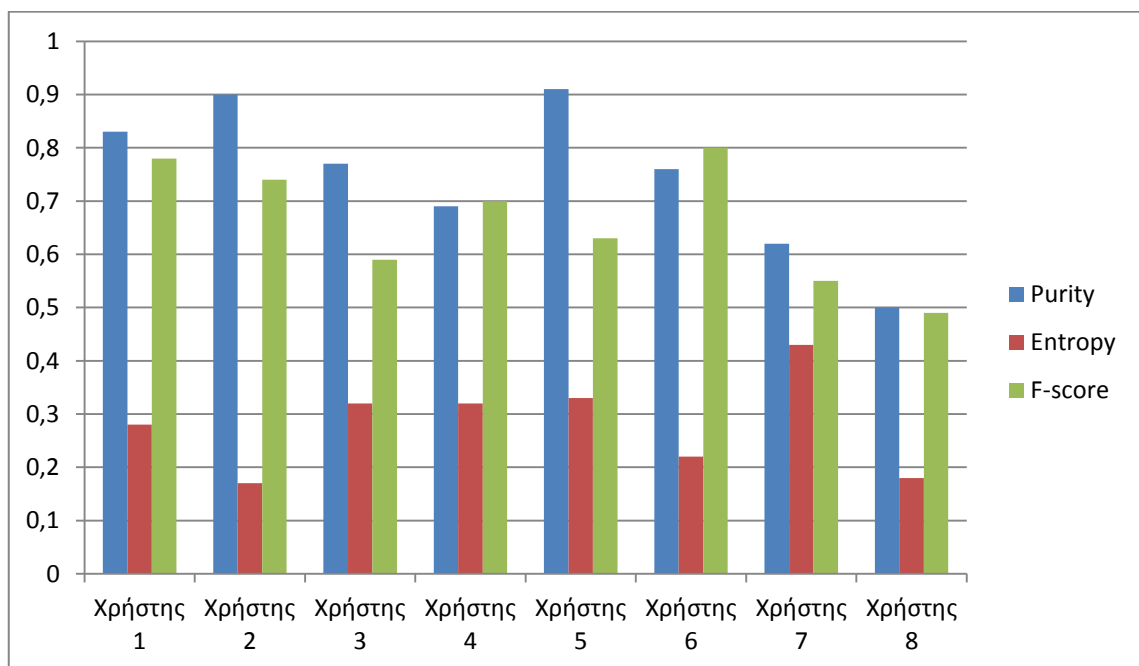
Οι πίνακες που ακολουθούν, δείχνουν για κάθε τιμή του κατώτατου ορίου (0.07, 0.05, 0.03 και 0.01) τις τιμές των μετρικών Purity, Entropy και F-Score.



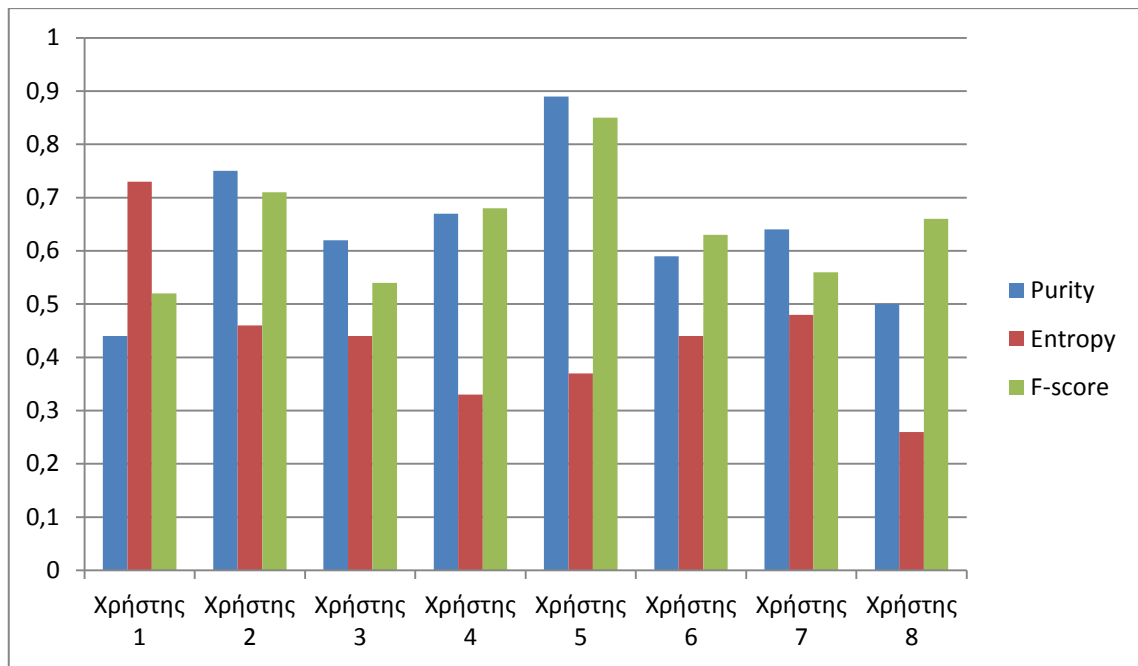
Σχήμα 6.4 - Μετρικές για $\theta=0.07$



Σχήμα 6.5 - Μετρικές για $\theta=0.05$



Σχήμα 6.6 - Μετρικές για $\theta=0.03$



Σχήμα 6.7 - Μετρικές για $\theta=0.01$

Το συμπέρασμα από αυτό το πείραμα είναι πως ο αλγόριθμος επηρεάζεται από τον πολύ θόρυβο που υπάρχει στην αδόμητη πληροφορία που αποκτάται. Παρόλο που οι μεγαλύτερες σε σημασία ομάδες αναγνωρίζονταν, το πρόγραμμα είτε δημιουργούσε περισσότερες ομάδες για το ίδιο σημασιολογικό περιεχόμενο είτε δημιουργούσε ομάδες οι οποίες δεν είχαν αναγνωριστεί από τους χρήστες.

Το πρώτο πρόβλημα οφείλεται στο γεγονός ότι η παράμετρος που ορίζει το κατώτατο όριο για τον αλγόριθμο της ομαδοποίησης, δεν μπορεί βάσει ορισμού, να είναι σταθερή για όλες τις περιπτώσεις. Παρόλο που για τους πλείστους χρήστες των πειραμάτων, το κατώτατο όριο μεταξύ 0.05 και 0.07 έδινε τα καλύτερα αποτελέσματα, εντούτοις αυτό δεν μπορεί ίσχυε για όλους τους χρήστες.

Το δεύτερο πρόβλημα, οφείλεται στο γεγονός ότι πολλά στιγμιότυπα είχαν προσωρινή σημασία για τους χρήστες. Αυτό σημαίνει πως παρόλο που την παρούσα στιγμή περιέγραφαν το περιβάλλον του χρήστη, αυτό το περιβάλλον ήταν ένα μεταβατικό περιβάλλον το οποίο δεν είχε ουσία για τον χρήστη. Αυτό βέβαια είναι κάτι που μία αυτοματοποιημένη λειτουργία δεν μπορεί να ανιχνεύσει.

6.4 Γενική αξιολόγηση συστήματος

Για τη γενικότερη αξιολόγηση του συστήματος, έχει συνταχθεί ένα ερωτηματολόγιο (Παράρτημα Α) το οποίο είχε δοθεί στους χρήστες που συμμετείχαν στον προηγούμενο πείραμα. Οι πίνακες με τις απαντήσεις φαίνονται πιο κάτω.

	Εύκολη	Μέτρια	Δύσκολη
Πόσο εύκολη ήταν η εγκατάσταση/χρήση του προγράμματος;	7	1	0

	10-20 λεπτά	30-60 λεπτά	> 30 λεπτά
Πόσο χρόνο αφιέρωσες για τη χρήση του προγράμματος;	2	6	0

	Καθόλου	Λίγο	Πολύ
Πόσο επιβάρυνε το πρόγραμμα τον υπολογιστή ή την εργασία που ήθελες να κάνεις;	4	3	1
Πόσο πιστεύεις ότι κινδυνεύουν τα προσωπικά σου δεδομένα από ένα τέτοιο πρόγραμμα;	1	6	1
Πιστεύεις ότι οι υπάρχουσες λειτουργίες είναι ικανοποιητικές;	2	6	0
Πόσο ξεκάθαρη ήταν η απεικόνιση των στιγμιότυπων και των ομάδων;	0	4	4
Πόσο ανταποκρίνονται οι ομάδες που ανακαλύφθηκαν με τις πραγματικές ομάδες;	0	5	3
Πόσο χρήσιμο πιστεύεις ότι είναι το πρόγραμμα;	3	4	1

	Όχι	Ναι
Θα χρησιμοποιούσες το πρόγραμμα ή άλλο παρόμοιο (που να προσφέρει παρόμοιες λειτουργίες) στο μέλλον;	6	2

	Όχι	1-2 ομάδες	Πάνω από 2
Υπήρχαν ομάδες που είχες ανακαλύψει αλλά δεν τις εμφάνισε το πρόγραμμα;	8	0	0

Από τις απαντήσεις των χρηστών, συμπεραίνουμε ότι το πρόγραμμα δούλεψε σε γενικές γραμμές ορθά και ικανοποιητικά. Το πρόγραμμα δεν επιβάρυνε την εργασία των χρηστών και ήταν εύκολο και απλό στη χρήση. Σε καμία περίπτωση δεν φάνηκε ότι οι χρήστες είχαν ανακαλύψει ομάδες τις οποίες να μην έχουν εμφανιστεί καθόλου. Αυτό σημαίνει πως γινόταν σωστή εξαγωγή πληροφοριών και χρήση των αλγορίθμων ομαδοποίησης.

Παρόλα αυτά, οι χρήστες δεν έδειξαν μεγάλο ενδιαφέρον για το πρόγραμμα και την ιδέα γενικότερα. Αυτό πιθανώς να οφείλεται στην έλλειψη γραφικών τα οποία να προσελκύουν τον χρήστη και είναι κάτι που ίσως διορθωθεί σε μελλοντικό στάδιο. Ένας άλλος λόγος είναι πιθανώς η αδυναμία των χρηστών να αντιληφθούν τα οφέλη που δύναται να παρέχει μια τέτοια εφαρμογή. Αυτό ίσως συμβαίνει και λόγω του γεγονότος ότι οι χρήστες είχαν υψηλότερες απαιτήσεις όσον αφορά τις παρεχόμενες λειτουργίες κάτι που επίσης μπορεί να ληφθεί υπόψη σε μελλοντικό στάδιο.

Κεφάλαιο 7

Συμπεράσματα

7.1	Επίλογος	48
7.2	Μελλοντική εργασία	49

Στο κεφάλαιο αυτό, δίνεται ο επίλογος της διπλωματικής εργασίας καθώς επίσης και οι πιθανοί στόχοι που μπορούν να υλοποιηθούν σε μελλοντική εργασία.

7.1 Επίλογος

Ο στόχος αυτής της διπλωματικής εργασίας, ήταν η ανάλυση των δυνατών τρόπων εξαγωγής του συγκειμένου του χρήστη και η υλοποίηση του πιο αποτελεσματικού με σκοπό την δημιουργία ενός συστήματος το οποίο να βρίσκεται σε θέση να απαντήσει ανά πάσα στιγμή για το συγκείμενο του χρήστη και να μπορεί να χαρακτηρίσει το τρέχον περιβάλλον του. Έχουν αναλυθεί και υλοποιηθεί διάφορες τεχνικές από τον κλάδο του Information Retrieval και προταθεί τρόποι εφαρμογής και αντιμετώπισης των υφιστάμενων μεθόδων σε δυναμικού χαρακτήρα περιεχόμενο όπως είναι το περιεχόμενο που αντλείται από τα παράθυρα με τα οποία αλληλεπιδρά ένας χρήστη κατά τη χρήση του υπολογιστή.

Ακολούθως, έγινε αξιολόγηση τόσο της αποτελεσματικότητας και αποδοτικότητας του συστήματος, όσο και της χρησιμότητάς του. Για το σκοπό αυτό, το σύστημα δούλεψε

με πραγματικούς χρήστες και πραγματική ροή δεδομένων και τα αποτελέσματα αφού συλλέχθηκαν, συγκρίθηκαν με τα θεωρητικά ορθά αποτελέσματα αργότερα.

Το γενικό συμπέρασμα της όλης εργασίας, είναι ότι υπάρχει πλούσιο contextual περιεχόμενο [3] κατά την αλληλεπίδραση ενός χρήστη με τον υπολογιστή και δουλεύοντας μόνο με ένα μέρος του, μέσα από αυτήν την εργασία, φάνηκε ότι μπορούν να ανακαλυφθούν πολλά σημαντικά στοιχεία με την επεξεργασία, που ίσως να μην είχαν καν ληφθεί υπόψη την στιγμή της σύλληψης τους.

Τέλος, φάνηκε πως το περιεχόμενο που συλλέγεται από τη χρήση του υπολογιστή, είναι πολύ βοηθητικό στον ίδιο τον χρήστη εφόσον λειτουργεί και σαν ένας προσωπικός βοηθός ο οποίος σε παρακολουθεί και ανά πάσα στιγμή μπορεί να σου θυμίσει τις ενέργειες που έκανες στο παρελθόν.

7.2 Μελλοντική εργασία

Η παρούσα διπλωματική εργασία, άγγιξε ένα πολυδιάστατο θέμα και προσπάθησε να παρουσιάσει τις προοπτικές που υπόσχεται μία περαιτέρω υλοποίηση της. Υπάρχουν ακόμη πολλές πηγές άντλησης περιεχομένου από την επιφάνεια εργασίας καθώς και μετα-πληροφορίας τα οποία μπορούν να συμβάλουν ακόμη περισσότερο στην πιο ακριβή αναπαράσταση του συγκεκριμένου. Επίσης, στον τομέα της επεξεργασίας και ανάλυσης της πληροφορίας, μπορούν να εφαρμοστούν περισσότερες μέθοδοι για σύγκριση με τις ήδη υλοποιημένες.

Τέλος, ένα σημαντικό στοιχείο, είναι η εύρεση νέων τρόπων άντλησης πληροφορίας η οποία δεν προέρχεται μόνο από το σύστημα αλλά και από το φυσικό περιβάλλον του χρήστη. Τέτοιοι τρόποι αφορούν την συλλογή πληροφοριών από αισθητήρες όπως κάμερες, μικρόφωνα αλλά και πιο προηγμένους όπως αισθητήρα ανάκτησης βλέμματος δηλαδή το που κοιτάζει ένας χρήστης ή απόσταση από οθόνη υπολογιστή.

Βιβλιογραφία

- [1] Bauer, Travis and Leake, David B. “Exploiting Information Access Patterns for Context-Based Retrieval.”, Proceedings of the 7th international conference on Intelligent user interfaces, 2002.
- [2] Bauer, Travis and Leake, David B., “WordSieve: A Method for Real-Time Context Extraction.”, Proceedings of the Third International and Interdisciplinary Conference, Context, 2001.
- [3] Budzik, Jay and Hammond, Kristian J., “User Interactions with Everyday Applications as Context for Just-in-time Information Access.”, 2000
- [4] Chahine, C. Abi, Chaignaud N., Kotowicz, JPh and Pecuchet, JP. “Context and Keyword Extraction in Plain Text using Graph Representation.”, Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference, 2008.
- [5] Challam, Vishnu Kanth Reddy., “Contextual Information Retrieval Using Ontology Based User Profiles”, 2004.
- [6] Dumais, Susan, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins., “Stuff I've Seen: A System for Personal Information Retrieval and Re-Use.”, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003.
- [7] Gyllstrom, Karl., “Passages Through Time: Chronicling Users' Information Interaction History by Recording When and What They Read.”, Proceedings of the 14th international conference on Intelligent user interfaces, 2009.
- [8] Gyllstrom, Karl, Craig Soules, and Alistair Veitch., “Activity put in context: Identifying implicit task context within the user's document interaction.”, Proceedings of the second international symposium on Information interaction in context, 2008.
- [9] Gyllstrom, Karl, Craig Soules, and Alistair Veitch., “Confluence: Enhancing Contextual Desktop Search.”, Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007.

- [10] Gyllstrom, Karl, και Craig Soules., “Seeing is retrieving: Building information context from what the user sees.”, Proceedings of the 13th international conference on Intelligent user interfaces, 2008.
- [11] Katakis, Ioannis, Grigorios Tsoumakas, and Ioannis Vlahanas., “Tracking Recurring Contexts using Ensemble Classifiers: An Application to Email Filtering.”, Knowledge and Information Systems, Vol.22, Issue 3, 2010.
- [12] Lieberman, Henry., “Letizia: An Agent That Assists Web Browsing.”, International Joint Conference on Artificial Intelligence, 2001.
- [13] Limbu, D.D., A.M. Connor, and S.G. MacDonell., “A Framework for Contextual Information Retrieval from the WWW.”, 2005.
- [14] Liu, Fangfang, and Jie Yu., “Mining User Context based on Interactive Computing for Personalized Web Search.”, Computer Engineering and Technology (ICCET), 2010 2nd International Conference, 2010.
- [15] Love, Robert, “Kernel korner: intro to inotify”, Linux Journal, Vol. 2005, Issue 139, 2005.
- [16] Lucene, <<http://lucene.apache.org>>
- [17] Nielsen, J., “How Little Do Users Read?”, 2008.
- [18] Oliver, Nuria, Greg Smith, Chintan Thakkar, and Arun C. Surendran., “SWISH: Semantic Analysis of Window Titles and Switching History.”, Proceedings of the 11th international conference on Intelligent user interfaces, 2006.
- [19] Paris, Cecile, and Stephen Wan., “Capturing the user's reading context for tailoring.”, 1998.
- [20] Porter, M.F., “An algorithm for suffix striping.” Program 14(3):130-137. 33, 529, 1980.
- [21] Schilit, B., Adams, N. and Want, R., “Context-Aware Computing Applications”, Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, 1994.
- [22] Shen, Jianqiang, Lida Li, Thomas G. Dietterich, and Jonathan L. Herlocker., “A Hybrid Learning System for Recognizing User Tasks from Desktop Activities and Email Messages.”, Proceedings of the 11th international conference on Intelligent user interfaces, 2006.

- [23] Shtykh, Roman Y., και Qun Jin., “Capturing User Contexts: Dynamic Profiling for Information Seeking Tasks.”, Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference, 2008.
- [24] Soules, Craig A. N., “Using context to assist in personal file retrieval.”, 2006.
- [25] Soules, Craig A. N., and Gregory R. Ganger., “Connections: Using Context to Enhance File Search.”, Proceedings of the twentieth ACM symposium on Operating systems principles, 2005.
- [26] Steinbach Michael, Karypis George, Kumar Vipin, “A Comparison of Document Clustering Techniques”, KDD Workshop on Text Mining, 2000.
- [27] Tanter, Eric, Gybels, Denker, Marcus and Bergel, Alexandre, “Context-Aware Aspects”, Proceedings of Software Composition 2006, LNCS 4089, Springer Verlag, pp. 227-242, 2006.
- [28] “TextCaptureX”, Deskperience, <[www. http://screen-scraper.deskperience.com/features-sdk](http://screen-scraper.deskperience.com/features-sdk)>
- [29] Widyanoro, Dwi H., Thomas R. Ioerger, and John Yen., “Learning User Interest Dynamics with a Three-Descriptor Representation.”, Journal of the American Society for Information Science and Technology, Vol. 52, Issue 3, 2001.
- [30] Zhao, Ying, and George Karypis., “Evaluation of Hierarchical Clustering Algorithms for Document Datasets.”, Proceedings of the eleventh international conference on Information and knowledge management, 2002.
- [31] Ελευθερίου, Ειρήνη. «Σύστημα Αυτόματης Ταξινόμησης Ηλεκτρονικού Ταχυδρομείου.», σ. 1-65, 2009.

Παράρτημα Α

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="context">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="snapshot" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="time" type="xs:string" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="-[0-9]"/>
                    <xs:length value="20"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="title" type="xs:string" use="required"/>
              <xs:element name="handle" type="xs:integer" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minExclusive value="0"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="process" type="xs:string" use="required"/>
              <xs:choice>
                <xs:element name="url" type="xs:string" minOccurs="0"/>
                <xs:sequence>
                  <xs:element name="mailto" type="xs:string" minOccurs="0"/>
                  <xs:element name="mailsender" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:choice>
              <xs:element name="runningprocesses" use="required">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="p" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="numofwords" type="xs:integer">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="0"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="words">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="word" minOccurs="1" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="w" type="xs:string" use="required">
                            <xs:simpleType>
                              <xs:restriction base="xs:string">
                                <xs:pattern value="[a-z]"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="f" type="xs:integer" use="required">
                            <xs:simpleType>
                              <xs:restriction base="xs:integer">
                                <xs:minInclusive value="1"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="t" type="xs:decimal" use="required"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="similarity" type="xs:decimal" use="required"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Παράρτημα Β

- 1) Πόσο εύκολη ήταν η εγκατάσταση/χρήση του προγράμματος;
Α) Εύκολη Β) Μέτρια Γ) Δύσκολη
- 2) Πόσο χρόνο αφιέρωσες για τη χρήση του προγράμματος;
Α) 10-20 λεπτά Β) 30-60 λεπτά Γ) Περισσότερα από 60 λεπτά
- 3) Πόσο επιβάρυνε το πρόγραμμα τον υπολογιστή ή την εργασία που ήθελες να κάνεις;
Α) Καθόλου Β) Λίγο Γ) Πολύ
- 4) Πόσο πιστεύεις ότι κινδυνεύουν τα προσωπικά σου δεδομένα από ένα τέτοιο πρόγραμμα;
Α) Καθόλου Β) Λίγο Γ) Πολύ
- 5) Θα χρησιμοποιούσες το πρόγραμμα στο μέλλον;
Α) Όχι Β) Ναι
- 6) Πιστεύεις ότι οι υπάρχουσες λειτουργίες είναι ικανοποιητικές;
Α) Καθόλου Β) Λίγο Γ) Πολύ
- 7) Πόσο ξεκάθαρη ήταν η απεικόνιση των στιγμιότυπων και των ομάδων;
Α) Καθόλου Β) Λίγο Γ) Πολύ
- 8) Πόσο ανταποκρίνονται οι ομάδες που ανακαλύφθηκαν με τις πραγματικές ομάδες;
Α) Καθόλου Β) Λίγο Γ) Πολύ
- 9) Υπήρχαν ομάδες που είχες ανακαλύψει αλλά δεν τις εμφάνισε το πρόγραμμα;
Α) Όχι Β) 1-2 ομάδες Γ) Πάνω από 2
- 10) Πόσο χρήσιμο πιστεύεις ότι είναι το πρόγραμμα;
Α) Καθόλου Β) Λίγο Γ) Πολύ