ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάκτηση Πληροφορίας Στον Ιστό με Χρήση Ταξινόμησης Όψεων και Συσταδοποίησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΠΑΝΑΓΙΩΤΑΣ Γ. ΓΕΩΡΓΙΟΥ

Επιβλέπων : Τίμος Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2009

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

Ανάκτηση Πληροφορίας Στον Ιστό με Χρήση Ταξινόμησης Όψεων και Συσταδοποίησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΠΑΝΑΓΙΩΤΑΣ Γ. ΓΕΩΡΓΙΟΥ

Επιβλέπων : Τίμος Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7η Ιουλίου 2009.

.....	.....	.....
Τίμος Σελλής	Ιωάννης Βασιλείου	Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2009

.....

ΠΑΝΑΓΙΩΤΑ ΓΕΩΡΓΙΟΥ

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

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Τίμο Σελλή για την επίβλεψη της διπλωματικής μου εργασίας και κυρίως γιατί με την προσωπικότητά του αποτέλεσε για μένα πηγή έμπνευσης.

Ευχαριστώ επίσης θερμά τους συνεπιβλέποντες Θεόδωρο Δαλαμάγκα και Γιώργο Γιαννόπουλο για την πολύτιμη καθοδήγηση και βοήθειά τους καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής αυτής εργασίας.

Τέλος, νιώθω την ανάγκη να ευχαριστήσω θερμά τους γονείς και τον αδερφό μου που στάθηκαν δίπλα μου και με στήριξαν όλα αυτά τα χρόνια.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας εφαρμογής ανάκτησης πληροφοριών στον Ιστό, με χρήση ταξινόμησης μέσω όψεων και συσταδοποίησης, η οποία θα αποτελεί συνέχεια της διπλωματικής εργασίας με τίτλο "Εργαλείο Συλλογής και Οργάνωσης Γνώσης με Μηχανισμούς Μετα-Αναζήτησης στον Ιστό" (Αργύρης Κόλλιας, Μάρτιος 2009) και θα επεκτείνει το εργαλείο χαρτογράφησης σκέψεων Freemind προσθέτοντας επιπλέον λειτουργικότητες σε αυτό.

Η εφαρμογή επικεντρώνεται στην ανάκτηση επιστημονικών δημοσιεύσεων από τον Ιστό και στη δημιουργία όψεων και συστάδων πάνω σε αυτές. Οι όψεις αφορούν τα πεδία 'ημερομηνία', 'συγγραφείς', 'δημοσίευση σε', 'τύπος δημοσίευσης', 'γενικοί όροι' και 'θεματική ενότητα'. Οι τέσσερις πρώτες όψεις δημιουργούνται με βάση πληροφορίες που αντλούμε από τη Βάση Δεδομένων του DBLP, ενώ οι δύο τελευταίες προκύπτουν από πληροφορίες που παίρνουμε απευθείας από το κείμενο των δημοσιεύσεων. Κάθε όψη αποτελεί στην ουσία ένα κριτήριο το οποίο ο χρήστης θέτει στα αποτελέσματα και ανά πάσα στιγμή μπορούν να προστεθούν, να αφαιρεθούν και να τροποποιηθούν κριτήρια. Η επιλογή της τιμής μιας όψης επηρεάζει τόσο τα εμφανιζόμενα αποτελέσματα όσο και τις τιμές στις υπόλοιπες όψεις. Η δημιουργία των συστάδων πραγματοποιείται θεωρώντας ως κριτήριο στοιχεία που επίσης αντλούνται απ' ευθείας από την εκάστοτε δημοσίευση. Οι συστάδες εμφανίζονται στο χρήστη με τρόπο παρόμοιο με αυτό των όψεων και επηρεάζουν και αυτές τα τελικά αποτελέσματα.

Λέξεις Κλειδιά: χειρισμός αποτελεσμάτων αναζήτησης, ταξινόμηση μέσω όψεων, συσταδοποίηση

Abstract

The aim of this thesis is to develop a faceted classification and clustering system for results of web searches. The system has been built on top of Freemind, a mind-mapping editor tool. With our system, the user is able to create a map of thoughts and search for information concerning a topic of this map in the World Wide Web. Our system focuses on retrieving papers. For a resulting list of papers, facets and clusters are created. The facets concern fields such as 'date', 'authors', 'published in', 'publication type', 'general terms' and 'categories and subject descriptors'. The first four facets are created based on information retrieved from the DBLP database, whereas the last two are based on information retrieved directly after parsing and processing the content of papers (where available). In fact, each facet imposes a filtering criterion on the results. The user can add, remove or change such criteria, by changing the values of each facet. The clustering task is performed using text content from the papers. Papers are organized in groups. Papers in each group are considered to be relevant to the same topic.

Keywords: web search results manipulation, faceted classification, clustering

Πίνακας περιεχομένων

1

Εισαγωγή

1.1 Χειρισμός Αποτελεσμάτων Αναζήτησης στον Ιστό

Στις μέρες μας αρχίζει να γίνεται ευρέως αποδεκτό το γεγονός ότι το παραδοσιακό interface αναζήτησης πληροφορίας στον Ιστό, το οποίο αποτελείται από ένα query box όπου ο χρήστης καλείται να εισάγει λέξεις κλειδιά προκειμένου να ανακτήσει τα δεδομένα που χρειάζεται, αρχίζει να γίνεται ανεπαρκές. Οι λόγοι για τους οποίους ισχύει κάτι τέτοιο είναι οι εξής:

- Οι μεγάλου μεγέθους συλλογές δεδομένων όπως οι online κατάλογοι προϊόντων, οι ψηφιακές βιβλιοθήκες και οι συλλογές εικόνων και φωτογραφιών, γίνονται όλο και πιο δημοφιλείς στο Διαδίκτυο. Η παραδοσιακή αναζήτηση, σε μια τέτοια μεγάλη συλλογή, θα επιστρέφει μεν στο χρήστη τα αποτελέσματα που εκείνος αναζήτησε, σε πολλές περιπτώσεις όμως, αυτά θα είναι τόσο πολλά, που ίσως δεν καταφέρει να τα χειριστεί με τον αποτελεσματικότερο δυνατό τρόπο. Ας θεωρήσουμε την περίπτωση ενός χρήστη του Διαδικτύου ο οποίος επισκέπτεται ένα ηλεκτρονικό κατάστημα προκειμένου να αγοράσει μια καινούρια τηλεόραση. Το κατάστημα διαθέτει μόνο την παραδοσιακή αναζήτηση κι έτσι ο χρήστης πληκτρολογεί τη λέξη κλειδί της αναζήτησής του στο αντίστοιχο query box. Τα αποτελέσματα που έρχονται σε αυτόν, ως απάντηση θα είναι στην καλύτερη περίπτωση μερικές εκατοντάδες, εάν λάβουμε υπόψη τις διαφορετικές μάρκες τηλεοράσεων που υπάρχουν, τα διαφορετικά μεγέθη οθόνης και διαφορετική ανάλυση αυτής, το αν η οθόνη θα είναι επίπεδη ή όχι, το

επίπεδο τιμών κλπ. Είναι προφανές ότι ο χρήστης θα χρειαστεί πολλές ώρες για να βρει την τηλεόραση που τον ενδιαφέρει.

- Πολλές φορές, κάποιος χρήστης μπορεί να χρειάζεται να κάνει μια πιο εξεζητημένη αναζήτηση στο διαδίκτυο. Για παράδειγμα, κάποιος φοιτητής μπορεί να χρειάζεται να βρει ένα paper το οποίο έχει γραφτεί πριν δέκα χρόνια. Ο φοιτητής δε γνωρίζει τον ακριβή τίτλο παρά μόνο ότι το paper αναφέρεται στο faceted classification. Ο συγγραφέας του οποίου το όνομα γνωρίζει, είναι καθηγητής σε γνωστό πανεπιστήμιο και δημοσιεύει αρκετά συχνά papers που αναφέρονται στο συγκεκριμένο θέμα. Η παραδοσιακή λοιπόν αναζήτηση στην οποία λέξεις κλειδιά θα είναι ο όρος faceted classification και το όνομα του συγγραφέα, θα επιστρέψει πολλά πιθανά αποτελέσματα. Το ζητούμενο αποτέλεσμα όμως, λόγω του ότι είναι τόσο παλιό και μάλλον δεν αναζητείται συχνά, δε θα είναι στις πρώτες σελίδες αποτελεσμάτων κι ο χρήστης θα πρέπει να ψάξει πάρα πολύ για να το βρει.

Από τα παραπάνω παραδείγματα, γίνεται φανερό το πόσο σημαντική είναι **η οργάνωση και ομαδοποίηση των αποτελεσμάτων της αναζήτησης**. Στις μέρες μας, οι διαπροσωπείες που υλοποιούν την ιδέα της οργάνωσης αποτελεσμάτων γίνονται όλο και πιο δημοφιλείς αφού βοηθούν στην αποδοτικότερη εύρεση εκείνων ακριβώς των αποτελεσμάτων που ο χρήστης αναζητά.

Δύο από τις τεχνικές που χρησιμοποιούνται τόσο για την αποτελεσματική ομαδοποίηση των αποτελεσμάτων αναζήτησης όσο και για την αποδοτικότερη εξερεύνηση (navigation) του Ιστού είναι η **ομαδοποίηση όψεων** (**faceted classification)** και η **συσταδοποίηση** (**clustering)**. Οι περισσότεροι άνθρωποι τείνουν να αντιλαμβάνονται την ομαδοποίηση (ή κατηγοριοποίηση) δεδομένων και πληροφοριών ως τη δημιουργία μιας ιεραρχίας. Η ιεραρχία είναι από τη φύση της μια top-down δομή αποτελούμενη από κάποιες γενικές κατηγορίες οι οποίες γίνονται όλο και πιο λεπτομερείς καθώς προχωράμε προς τα κάτω, μέχρι να φτάσουμε στο αντικείμενο που αναζητούμε. Στην ιεραρχία αυτού του τύπου, κάθε αντικείμενο μπορεί να ανήκει σε μία μόνο κατηγορία και υπάρχει μόνο ένα μονοπάτι προκειμένου να φτάσουμε στο αντικείμενο.

Σε αντίθεση με την κλασσική ιεραρχία, η τεχνική της ομαδοποίησης μέσω όψεων αποτελεί ένα bottom-up σχήμα, όπου κάθε αντικείμενο είναι συνυφασμένο με ένα πεπερασμένο σύνολο από **χαρακτηριστικά** (**attributes)** και **τιμές (values)** και η ομαδοποίηση των δεδομένων εξαρτάται κάθε φορά από τον τρόπο που ένας χρήστης θα επιλέξει να έχει πρόσβαση σε αυτά. Δηλαδή, ο χρήστης επιλέγει προοδευτικά τα κριτήρια με τα οποία μεγάλες συλλογές δεδομένων θα φιλτραριστούν μέχρι να φτάσει σε ένα "εύκολα διαχειρίσιμο" σύνολο αποτελεσμάτων, αντί να κινείται σε μια ήδη καθορισμένη ιεραρχία αντικειμένων.

Η συσταδοποίηση αφορά την κατηγοριοποίηση μιας συλλογής αντικειμένων σύμφωνα με κάποιο **μέτρο ομοιότητας (similarity measure)** και στη δημιουργία συστάδων, δηλαδή ομάδων ομοειδών αντικειμένων από την αρχική συλλογή. Η διαδικασία δημιουργίας των συστάδων γίνεται αυτόματα με χρήση ειδικών εργαλείων λογισμικού και αλγορίθμους συσταδοποίησης, ενώ το μέτρο ομοιότητας εξαρτάται κάθε φορά από το είδος της συλλογής. Για παράδειγμα, στη συσταδοποίηση μιας συλλογής αρχείων κειμένου, κάθε συστάδα θα αποτελείται από κείμενα στα οποία οι βασικές λέξεις και φράσεις αναφέρονται στην ίδια ή παρόμοιες θεματικές περιοχές.

Η βασική διαφορά των δύο ταξινομήσεων είναι ότι στην ομαδοποίηση όψεων ο χρήστης θέτει τα κριτήρια της αναζήτησης καθορίζοντας την ταξινόμηση των τελικών αποτελεσμάτων, ενώ στη συσταδοποίηση οι ομάδες καθορίζονται αποκλειστικά από το σύστημα με βάση τους αλγόριθμους που εφαρμόζονται.

1.2 Αντικείμενο διπλωματικής

Με την παρούσα διπλωματική εργασία, στοχεύουμε στην υλοποίηση ενός συστήματος το οποίο θα ξεκινάει με μια κλασσική, μέσω λέξεων κλειδιών, αναζήτηση του χρήστη και στη συνέχεια θα δίνει τη δυνατότητα (α) ομαδοποίησης όψεων και (β) συσταδοποίησης αποτελεσμάτων.

Το σύστημα μας, αναπτύχθηκε πάνω στο open source εργαλείο Freemind, το οποίο και είναι ένας συντάκτης **χαρτών σκέψεων** (**mindmaps**). Ένας χάρτης σκέψεων είναι στην ουσία ένα διάγραμμα, το οποίο χρησιμοποιείται για την οργάνωση γνώσεων και πληροφοριών και αποτελείται από κόμβους καθένας από τους οποίους αντιπροσωπεύει μια ιδέα ή ένα αντικείμενο. Οι κόμβοι οργανώνονται σε μια ιεραρχία και μπορεί να περιέχουν κείμενο ή κάποιον ηλεκτρονικό σύνδεσμο. Η διαδικασία κατασκευής ενός χάρτη μνήμης καλείται χαρτογράφηση (mind–mapping) και είναι ιδιαίτερα χρήσιμη στην ερευνητική διαδικασία, την ανάλυση προβλημάτων, την σύνθεση λύσεων και τη λήψη αποφάσεων. Η διπλωματική είναι επέκταση μιας προηγούμενης διπλωματικής εργασίας με τίτλο “Εργαλείο Συλλογής και Οργάνωσης Γνώσης με Μηχανισμούς Μετα-Αναζήτησης στον Ιστό” (Αργύρης Κόλλιας, Μάρτιος 2009).

Το βασικό σενάριο χρήσης του συστήματος, είναι ότι χρήστης δημιουργεί ένα χάρτη σκέψης, και αναζητά πληροφορίες για κάποιον όρο από αυτόν το χάρτη σκέψης στον Ιστό, ή εκτελεί απευθείας την κλασσική αναζήτηση μέσω λέξεων κλειδιών. Για την αναζήτησή του δίνεται η επιλογή τα αποτελέσματα να είναι είτε ιστοσελίδες, είτε επιστημονικές δημοσιεύσεις.

Στα πλαίσια της διπλωματικής εργασίας, αναπτύχθηκαν δύο επιπλέον υποσυστήματα: υποσύστημα ομαδοποίησης όψεων και υποσύστημα συσταδοποίησης.

Υποσύστημα ομαδοποίησης όψεων

Στην περίπτωση που τα αποτελέσματα έχουν επιλεγεί να είναι δημοσιεύσεις, απομονώνουμε από αυτά στοιχεία για την ημερομηνία δημοσίευσης, τους συγγραφείς, το αν η δημοσίευση έγινε σε επιστημονική εφημερίδα/περιοδικό ή σε επιστημονικό συνέδριο και σε ποιο συνέδριο/ εφημερίδα/περιοδικό. Με τα στοιχεία αυτά εξάγουμε τις όψεις και τις αντίστοιχες τιμές τους. Για την εξαγωγή των τιμών όψεων χρησιμοποιήσαμε στοιχεία από τη βάση δεδομένων του DBLP (Digital Bibliography & Library Project). Το DBLP αποτελεί έναν ιστότοπο που περιλαμβάνει οργανωμένη βιβλιογραφία της Επιστήμης των Υπολογιστών.

Ας θεωρήσουμε ένα παράδειγμα χρήσης προκειμένου να γίνουν περισσότερο κατανοητές οι λειτουργίες του συστήματος. Έστω ότι αναζητούμε με τον κλασσικό τρόπο πληροφορίες οι οποίες αφορούν το θέμα faceted classification. Το σύστημα εξάγει τις τιμές όψεων. Έστω ότι επιλέγουμε την τιμή '2005' από την όψη "ημερομηνία". Τότε τα αποτελέσματα που εμφανίζονται στο χρήστη, περιορίζονται σε μόνο όσα έχουν εκδοθεί το 2005. Επίσης, οι υπόλοιπες όψεις περιορίζουν τις τιμές τους σε όσες είναι συμβατές με το 2005. Για παράδειγμα, στην όψη "συγγραφείς" παραμένουν μόνοι οι συγγραφείς που έχουν δημοσίευση το 2005. Ας υποθέσουμε ότι ο χρήστης συνεχίζει την επεξεργασία επιλέγοντας την τιμή 'S. R. Ranganathan' από την όψη "συγγραφείς". Τώρα, τα αποτελέσματα περιορίζονται ακόμα περισσότερο σε όσα έχουν δημοσιευτεί το 2005 **και** ένας από τους συγγραφείς τους είναι ο S. R. Ranganathan.

Παρατηρούμε δηλαδή ότι η τιμή μιας όψης αποτελεί ένα κριτήριο περιορισμού των τελικών αποτελεσμάτων και των τιμών των όψεων, και κάθε φορά πρέπει να ικανοποιείται το λογικό γινόμενο (AND) όλων των κριτηρίων.

Ο χρήστης εκτός από το να προσθέτει, μπορεί και να αφαιρεί κριτήρια, διευρύνοντας έτσι τη συλλογή αποτελεσμάτων. Επίσης μπορεί και να τροποποιεί κριτήρια που έχει θέσει προηγουμένως.

Υποσύστημα συσταδοποίησης

Εκτός από τη δυνατότητα της ομαδοποίησης όψεων, στους χρήστες παρέχεται και η δυνατότητα της συσταδοποίησης, δηλαδή της θεματικής οργάνωσης των αποτελεσμάτων της αναζήτησης. Η συσταδοποίηση γίνεται χρησιμοποιώντας περιεχόμενο κειμένου από τις δημοσιεύσεις. Συγκεκριμένα, χρησιμοποιούμε το Google για την εξεύρεση του κειμένου της δημοσίευσης (pdf) και από εκεί απομονώνουμε το Abstract και τους γενικούς όρους χαρακτηρισμού της δημοσίευσης, π.χ. General Terms, Category (όπου αυτοί υπάρχουν), από το υπόλοιπο κείμενο. Για την υλοποίηση της συσταδοποίησης, χρησιμοποιήσαμε το εργαλείο CLUTO.

1.2.1 Συνεισφορά

Από όλα τα παραπάνω, γίνεται φανερό, πως η συνεισφορά της παρούσας διπλωματικής εργασίας συνοψίζεται στα παρακάτω:

- σχεδίαση και υλοποίηση υποσυστήματος ομαδοποίησης αποτελεσμάτων αναζήτησης με χρήση όψεων,
- σχεδίαση και υλοποίηση υποσυστήματος συσταδοποίησης αποτελεσμάτων αναζήτησης,
- ολοκλήρωση και ενσωμάτωση των υποσυστημάτων σε ένα πρότυπο εργαλείο συλλογής και οργάνωσης γνώσης με μηχανισμούς μετα-αναζήτησης στον Ιστό, που είναι υπό ανάπτυξη στο Ινστιτούτο Πληροφοριακών Συστημάτων και Προσομοίωσης, Ερευνητικό Κέντρο "Αθηνά".

1.3 Οργάνωση κειμένου

Η διπλωματική εργασία αποτελείται από οχτώ κεφάλαια, το περιεχόμενο των οποίων περιγράφεται πολύ συνοπτικά στην παρούσα ενότητα. Το κεφάλαιο 1 αποτελεί μια γενική εισαγωγή με την οποία προσπαθούμε να περιγράψουμε το πρόβλημα που προκύπτει από την κλασσική αναζήτηση μέσω λέξεων κλειδιών στον Ιστό και την προσπάθεια για αντιμετώπιση του προβλήματος αυτού με κατάλληλο χειρισμό των αποτελεσμάτων μιας τέτοιας αναζήτησης. Στο κεφάλαιο 2 γίνεται προσπάθεια να προσεγγίσουμε θεωρητικά τις έννοιες της ταξινόμησης μέσω όψεων και της συσταδοποίησης, μέσω των οποίων θα χειριστούμε και τα αποτελέσματα των αναζητήσεων. Επίσης γίνεται αναφορά στις βασικές αρχές που πρέπει να πληροί το interface μιας τέτοιας εφαρμογής, προκειμένου αυτή να είναι εύχρηστη και να παρέχει ανά πάσα στιγμή τις απαιτούμενες πληροφορίες προς το χρήστη. Το κεφάλαιο 3 αφορά το συνολικό σύστημα. Αναφερόμαστε στα επιμέρους υποσυστήματα που το αποτελούν, στο σχεδιασμό αυτών και στις λειτουργίες που καθένα τους επιτελεί. Το κεφάλαιο 4 αναφέρεται στην αρχιτεκτονική του συστήματος. Γίνεται μια αναλυτική περιγραφή των κλάσεων που δημιουργήθηκαν ή τροποποιήθηκαν. Επίσης, γίνεται και αναφορά στη βάση δεδομένων με την οποία αλληλεπιδρά η εφαρμογή. Το κεφάλαιο 5 περιγράφει τις σημαντικότερες λεπτομέρειες της υλοποίησης και συγκεκριμένα τα θέματα εκείνα τα οποία χρειάστηκαν κάποιον ειδικό χειρισμό. Επίσης, παραθέτουμε οδηγίες σχετικές με την εξ' αρχής εγκατάσταση του συστήματος σε έναν υπολογιστή. Στο κεφάλαιο 6 περιγράφεται αναλυτικά ένα σενάριο χρήσης του συστήματος και αποτελεί ουσιαστικά επίδειξη όλων των λειτουργιών του συστήματος. Παραθέτονται σε κάθε περίπτωση τα αντίστοιχα screenshots προκειμένου να γίνουν περισσότερο κατανοητά όσα περιγράφονται. Στο κεφάλαιο 7 συνοψίζουμε τα αποτελέσματα της διπλωματικής και περιγράφουμε τα

συμπεράσματα που προέκυψαν. Επιπλέον, γίνεται αναφορά σε μελλοντικές επεκτάσεις που μπορούν να γίνουν στο σύστημα για να επεκταθεί. Τέλος, το κεφάλαιο 8 περιέχει όλες τις αναφορές στη βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή του εν λόγω κειμένου.

2

Θεωρητικό Υπόβαθρο και Σχετικές Εργασίες

Στο συγκεκριμένο κεφάλαιο περιγράφουμε τις θεωρητικές γνώσεις οι οποίες είναι απαραίτητες για την κατανόηση τόσο των λειτουργιών του συστήματος, όσο και των δυσκολιών που χρειάστηκαν να αντιμετωπιστούν κατά την υλοποίηση, και οι οποίες περιγράφονται σε επόμενα κεφάλαια. Ακόμα, γίνεται αναφορά σε σχετικές επιστημονικές εργασίες που ασχολούνται με τα θέματα που μας ενδιαφέρουν.

2.1 Faceted Classification

2.1.1 Ορισμός

Τα συστήματα που χρησιμοποιούν αναζήτηση μέσω όψεων (faceted classification) επιτρέπουν την ταξινόμηση των αντικειμένων με πολλαπλούς τρόπους σε σχέση με την κλασσική ταξινόμηση, δίνοντας τη δυνατότητα στους χρήστες να καθορίζουν τις κατηγορίες ταξινόμησης και τον τρόπο πλοήγησης ανάμεσα στις κατηγορίες αυτές. Για παράδειγμα, σε έναν οδηγό εστιατορίων που χρησιμοποιεί κλασσικές μεθόδους ταξινόμησης, τα εστιατόρια πιθανόν να έχουν κατηγοριοποιηθεί είτε με βάση την περιοχή στην οποία βρίσκονται είτε με βάση το είδος, τις τιμές ή τις κριτικές που έχει λάβει. Χρησιμοποιώντας faceted classification,

δίνεται στο χρήστη η δυνατότητα να ταξινομήσει τα εστιατόρια αρχικά ανά τιμή, στη συνέχεια ανά περιοχή και τέλος ανά είδος, ενώ κάποιος άλλος χρήστης θα τα ταξινομούσε αρχικά με βάση το είδος και στη συνέχεια με βάση τις κριτικές που έχουν λάβει. Με άλλα λόγια, τα συστήματα faceted classification παρέχουν στο χρήστη ένα σύνολο κατηγοριών (facets) τις οποίες ο χρήστης μπορεί να συνδυάσει και να αξιοποιήσει σύμφωνα με τις ανάγκες του.

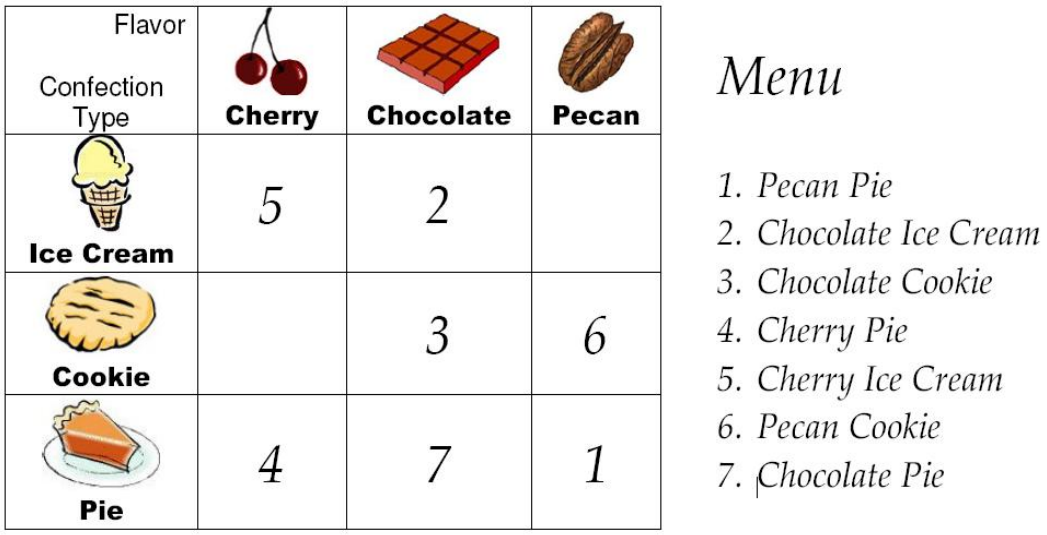
Κάθε ***όψη (facet)*** αποτελεί μια ομάδα από ***τιμές (values)*** οι οποίες αναφέρονται σε κάποιο συγκεκριμένο τρόπο ταξινόμησης ανάλογα με τα αντικείμενα που πρόκειται να ταξινομηθούν. Τα values μπορεί να αναφέρονται και ως headings ή labels, αλλά στην παρούσα διπλωματική εργασία υιοθετούμε τον όρο value. Στο παραπάνω παράδειγμα των εστιατορίων τα facets που μπορούν να δημιουργηθούν είναι η περιοχή, το είδος του εστιατορίου, η τιμή και οι κριτικές. Τα values για το facet κριτικές θα μπορούσαν να είναι: 'πολύ καλή', 'καλή', 'μέτρια' και 'κακή'. Τα ***αντικείμενα (resources)*** μπορεί να είναι έγγραφα, πρόσωπα ή οποιαδήποτε αγαθά τα οποία αναζητούμε με την ταξινόμηση. Σε κάθε αντικείμενο αποδίδονται κάποια values και έτσι φτάνουμε στο αποτέλεσμα της αναζήτησης. Στο παραπάνω παράδειγμα τα αντικείμενα είναι τα εστιατόρια.

Οι χρήστες μπορούν να φτάσουν στο επιθυμητό αποτέλεσμα ανεξάρτητα από τη σειρά πλοήγησης ανάμεσα στα facets. Για παράδειγμα, έστω ότι κάποιος χρήστης ενδιαφέρεται για εστιατόρια σε μια συγκεκριμένη περιοχή τα οποία έχουν μέχρι κάποια ορισμένη τιμή. Η αναζήτηση μπορεί να γίνει είτε επιλέγοντας αρχικά το value με την περιοχή ενδιαφέροντος από το facet περιοχών, και στη συνέχεια το value με την τιμή που είναι διατεθειμένος να πληρώσει, είτε το ανάποδο. Η ελευθερία στην πλοήγηση/αναζήτηση δεν επηρεάζει τα αποτελέσματα της ταξινόμησης.

Στα facets όπου τα values είναι πολλά σε αριθμό, μπορούμε για διευκόλυνσή μας, να τα ταξινομήσουμε σε στατικές ιεραρχίες. Για παράδειγμα, έστω κάποιο facet που αναφέρεται στην τοποθεσία. Αυτό μπορεί να ταξινομηθεί με βάση την πόλη, τις γειτονιές και τέλος τους δρόμους.

2.1.2 Δημιουργία facets

Ένα από τα ζητήματα που προκύπτει είναι τα κριτήρια με βάση τα οποία θα δημιουργήσουμε τα facets. Με μια πρώτη ματιά, η δημιουργία των facets φαίνεται εύκολη και η προσπάθεια δημιουργίας τους με βάση τα 'φυσικά χαρακτηριστικά', όπως έχουμε ήδη δει παραπάνω, φαίνεται να δουλεύει. Στην πραγματικότητα, τα πράγματα δεν είναι τόσο απλά. Ας θεωρήσουμε το παράδειγμα ενός ζαχαροπλαστείου και επίσης ότι δημιουργούμε τα facets με βάση τα φυσικά χαρακτηριστικά των γλυκών. Θα δημιουργήσουμε δύο facets: 'είδος γλυκού' και 'γεύση' όπως στο παρακάτω σχήμα.

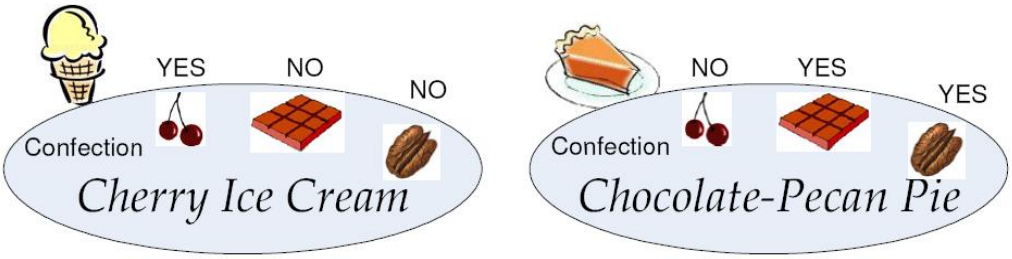


Σχήμα 2.1 Δημιουργία facets

Το πρώτο θα έχει τα values: 'παγωτό', 'μπισκότο' και 'τάρτα' και το δεύτερο: 'κεράσι', 'σοκολάτα' και 'καρύδι'. Όλα τα γλυκά που αποτελούν το menu μπορούν να περιγραφούν με αυτή την ταξινόμηση. Το ίδιο όμως δεν ισχύει για οποιοδήποτε γλυκό. Αν για παράδειγμα θελήσουμε να προσθέσουμε στο menu παγωτά με δύο γεύσεις, αυτά δε μπορούν να περιγραφούν με βάση τα υπάρχοντα facets.

Μέχρι τώρα, αν και δεν έχει αναφερθεί ξεκάθαρα, έχουμε υποθέσει ότι το κάθε αντικείμενο μπορεί να λάβει μόνο ένα value από κάθε facet. Χωρίς την υπόθεση αυτή θα μιλούσαμε για tagging και clustering (ενότητα 2.2) και όχι για faceted classification. Είναι προφανές ότι η δημιουργία facets με βάση τα φυσικά χαρακτηριστικά δεν είναι η κατάλληλη. Ο περιορισμός ο οποίος ακολουθούμε για το σχεδιασμό των facets είναι το ***να περιέχουν χαρακτηριστικά τα οποία είναι μεταξύ τους αμοιβαία αποκλειόμενα*** έτσι ώστε η απονομή ενός value σε κάποιο αντικείμενο να αποκλείει την απονομή οποιουδήποτε άλλου value από το ίδιο facet. Η απονομή value από οποιοδήποτε άλλο facet είναι δυνατή.

Σύμφωνα με αυτά, τα facets που πρέπει να δημιουργήσουμε για το παραπάνω παράδειγμα είναι τα: 'είδος γλυκού', 'σοκολάτα', 'κεράσι' και 'καρύδι'. Δηλαδή έχουμε τέσσερα facets, όσες είναι και οι ερωτήσεις που μπορούμε να κάνουμε για ένα γλυκό και οι οποίες είναι ανεξάρτητες μεταξύ τους. Τα facets φαίνονται στο παρακάτω σχήμα:



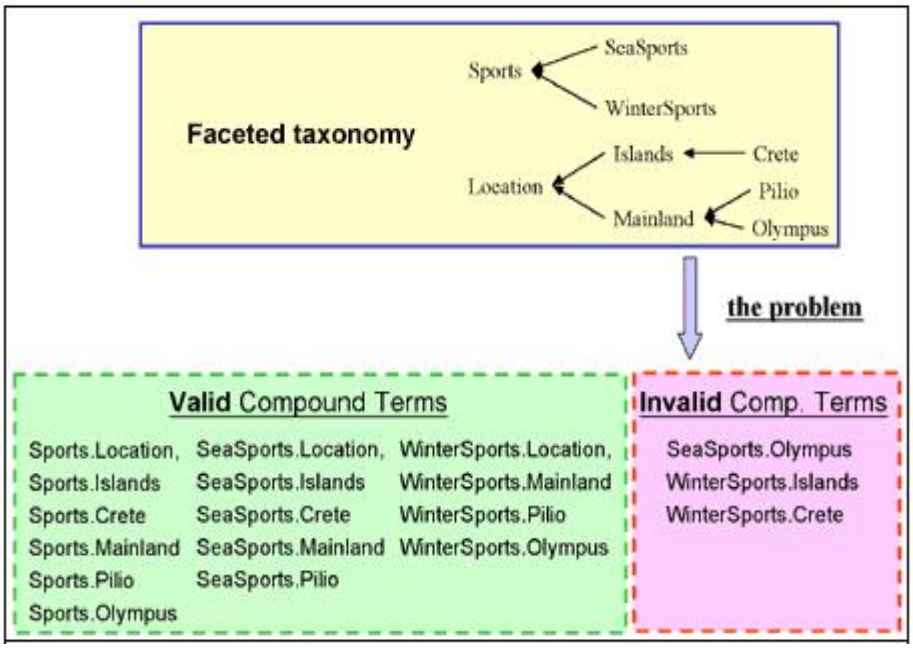
Σχήμα 2.2 Σωστή κατασκευή facets

Τρία από αυτά έχουν δυαδικές τιμές. Οποιοδήποτε γλυκό μπορεί τώρα να ταξινομηθεί. Στην πραγματικότητα, τώρα είναι που τα facets απεικονίζουν σωστότερα το φυσικό κόσμο. Οι γεύσεις μπορούν από τη φύση τους να συνδυαστούν οπότε ως μη αμοιβαία αποκλειόμενες αποτελούν ξεχωριστά facets. Αντίθετα, τα διαφορετικά values για το είδος γλυκού είναι αμοιβαία αποκλειόμενα μεταξύ τους γι αυτό και αποτελούν ενιαίο facet.

2.1.3 Ειδικά Θέματα Σχεδιασμού

Εκτός από τη δημιουργία του σωστού αριθμού facets, ένα άλλο θέμα στο οποίο χρειάζεται ιδιαίτερη προσοχή κατά το σχεδιασμό ενός συστήματος faceted classification, είναι το ενδεχόμενο να οδηγείται ο χρήστης σε ***κενό σύνολο αποτελεσμάτων (null result set)*** καθώς θέτει τα κριτήρια της αναζήτησης.

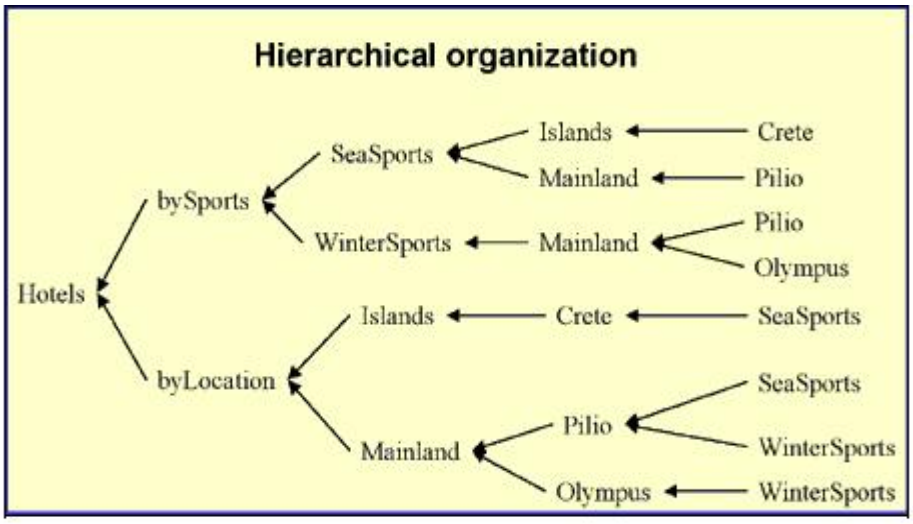
Από τον ορισμό του faceted classification, η επιλογή values από δύο διαφορετικά facets, σημαίνει το λογικό γινόμενο (λογικό AND) ανάμεσα στις δύο διαφορετικές ταξινομήσεις που προκύπτουν από τα facets. Ας θεωρήσουμε το παράδειγμα ενός τουριστικού οδηγού όπως αυτός που φαίνεται στο παρακάτω σχήμα:



Σχήμα 2.3 Παράδειγμα οργάνωσης ταξιδιωτικού οδηγού με χρήση faceted classification

Εάν κάποιος χρήστης επιλέξει θαλάσσια σπορ (SeaSports) από το πρώτο facet και Όλυμπος (Olympus) από το δεύτερο, τότε, δεδομένου ότι στον Όλυμπο δεν υπάρχει θάλασσα, ο χρήστης θα οδηγηθεί σε κενό σύνολο αποτελεσμάτων, γεγονός μη αποδεκτό για ένα επιτυχημένο σύστημα. Το παραπάνω πρόβλημα, αποτελεί σημαντικό περιοριστικό παράγοντα στην οργάνωση συστημάτων με faceted classification. Αντί αυτού, προτιμούνταν,

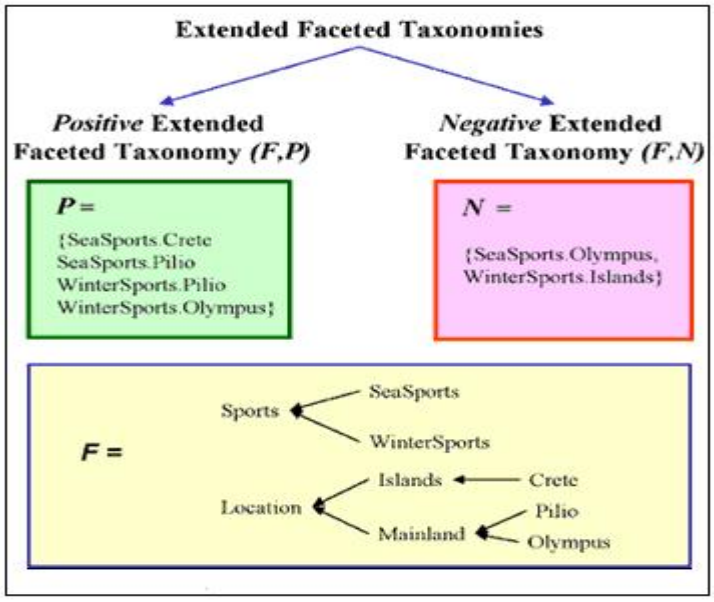
τουλάχιστον μέχρι πρόσφατα, δομές όπως αυτή του *σχήματος 2.4* με εμφανή τα προβλήματα του τεράστιου μεγέθους για μεγάλα συστήματα και της πολύπλοκης δομής. Από τα



Σχήμα 2.4 *Παράδειγμα οργάνωσης ταξιδιωτικού οδηγού χωρίς faceted classification*

παραπάνω γίνεται φανερό το πόσο σημαντική είναι η αναφορά στους έγκυρους και μη έγκυρους συνδυασμούς facet values όπως στο δεύτερο κομμάτι του *σχήματος 2.3*.

Μια προτεινόμενη λύση για το παραπάνω πρόβλημα είναι η δημιουργία δύο συνόλων P και N, τα οποία επεκτείνουν το σύνολο F των facets *(σχήμα 2.5)*. Το σύνολο P αποτελείται από τους έγκυρους συνδυασμούς, όπως αυτοί έχουν οριστεί παραπάνω, και το σύνολο N από τους μη έγκυρους. Έτσι ο χρήστης, γνωρίζοντας τα δύο παραπάνω σύνολα, αποφεύγει να θέσει ως κριτήριο κάποιον συνδυασμό facet values που θα τον οδηγήσει σε κενό σύνολο αποτελεσμάτων.



Σχήμα 2.5 Επέκταση του συνόλου των facets

Μια δεύτερη λύση είναι οι έγκυροι και μη έγκυροι συνδυασμοί να διαφαίνονται στο χρήστη μέσω του interface της εφαρμογής. Η λύση αυτή έχει επιλεγεί στην υλοποίηση της παρούσας διπλωματικής εργασίας και αναλύεται περαιτέρω στην ενότητα 2.3.2.

2.1.4 Πλεονεκτήματα

Η υλοποίηση ιεραρχιών faceted classification παρουσιάζει πολλά πλεονεκτήματα σε σχέση με την παραδοσιακή ταξινόμηση. Τα κυριότερα είναι τα παρακάτω:

- Εστιάζει στα σημαντικά και απολύτως απαραίτητα χαρακτηριστικά των αντικειμένων κάτι που είναι πολύ σημαντικό για εφαρμογές στις οποίες τα δεδομένα αλλάζουν γρήγορα με το χρόνο και η ανανέωση κρίνεται απαραίτητη.
- Παρουσιάζει αποδοτικότητα στο χρόνο εκτέλεσης αφού τα στοιχεία που δέχονται για επεξεργασία είναι μόνο τα facets και τα αντικείμενα προς ταξινόμηση. Σε άλλου τύπου ταξινομήσεις (tagging) τα values οργανώνονται σε πίνακες και προκειμένου να φτάσουμε στο επιθυμητό αποτέλεσμα είναι απαραίτητα τα joins μεταξύ πινάκων κάτι που είναι πολύ χρονοβόρο.
- Εννοιολογική καθαρότητα. Ας θεωρήσουμε την κλασσική ταξινόμηση η οποία παρέχει 100 όρους ως προς τους οποίους μπορούν να ταξινομηθούν τα αντικείμενα, και μια ταξινόμηση όψεων με 10 facets καθένα από τα οποία έχει 10 values. Στο δεύτερο σχήμα ο χρήστης αντιλαμβάνεται ευκολότερα τις δυνατότητες που του παρέχει το σύστημα και μπορεί να τις χρησιμοποιήσει σε μεγαλύτερο βαθμό σε σχέση με το πρώτο σχήμα.
- Οικονομία ως προς την ποσότητα μνήμης που χρειάζεται για αποθήκευση του συστήματος. Στο παραπάνω παράδειγμα, με μόλις 100 αποθηκευμένους όρους (10 facets καθένα από τα οποία έχει 10 values) μπορούμε να πετύχουμε 10 δισεκατομμύρια διαφορετικούς τρόπους ταξινόμησης. Στο κλασσικό σχήμα, θα χρειαζόταν λοιπόν να δεσμεύσουμε μνήμη 100 εκατομμύρια φορές μεγαλύτερη προκειμένου να πετύχουμε τις ίδιες επιλογές ταξινόμησης.
- Γενική ευελιξία του συστήματος αφού ανάλογα με τις ανάγκες της εφαρμογής μπορούμε να προσθέσουμε ή να αφαιρέσουμε, χωρίς να επηρεάζεται το υπόλοιπο σύστημα.
- Η πληροφορίες που είναι αποθηκευμένες στο σύστημα καθίστανται ιδιαίτερα χρήσιμες λόγω του ότι παρέχονται πολλά διαφορετικά μονοπάτια προσπέλασης ανάλογα με τις ανάγκες του εκάστοτε χρήστη

2.1.5 Μειονεκτήματα

Δύο είναι τα βασικά μειονεκτήματα της ταξινόμησης μέσω όψεων:

- Δεν υπάρχει κάποια επίσημη μέθοδος για την επιλογή και τη δημιουργία των facets. Έτσι, προς το παρόν, ο σχεδιαστής του συστήματος πρέπει να καθορίσει τα facets εξ' αρχής και να επεμβαίνει στο σύστημα κάθε φορά που χρειάζεται να προστεθεί ή να αφαιρεθεί κάποιο από αυτά.
- Η όλη ιδέα φαίνεται πολύ αφηρημένη και δύσκολη στο χειρισμό κυρίως σε μη έμπειρους χρήστες, όπως έχει προκύψει από σχετικές έρευνες.

Παρά τους παραπάνω περιορισμούς, τα πλεονεκτήματα που παρέχονται, ειδικά στις online εφαρμογές, είναι πολύ περισσότερα, γι αυτό και η ταξινόμηση μέσω όψεων αποτελεί σημαντικό αντικείμενο έρευνας.

2.2 Clustering

2.2.1 Ορισμός

Ο όρος ***clustering (συσταδοποίηση)*** αναφέρεται στην κατηγοριοποίηση μιας συλλογής αντικειμένων σύμφωνα με κάποιο ***μέτρο ομοιότητας (measure of similarity)*** και η δημιουργία clusters, δηλαδή ομάδων ομοειδών αντικειμένων από την αρχική συλλογή. Η διαδικασία δημιουργίας των clusters γίνεται αυτόματα με χρήση ειδικών εργαλείων λογισμικού και το μέτρο ομοιότητας εξαρτάται κάθε φορά από το είδος της συλλογής. Για παράδειγμα, στο clustering μιας συλλογής αρχείων κειμένου, κάθε cluster θα αποτελείται από κείμενα στα οποία οι βασικές λέξεις και φράσεις αναφέρονται στην ίδια ή παρόμοιες θεματικές περιοχές.

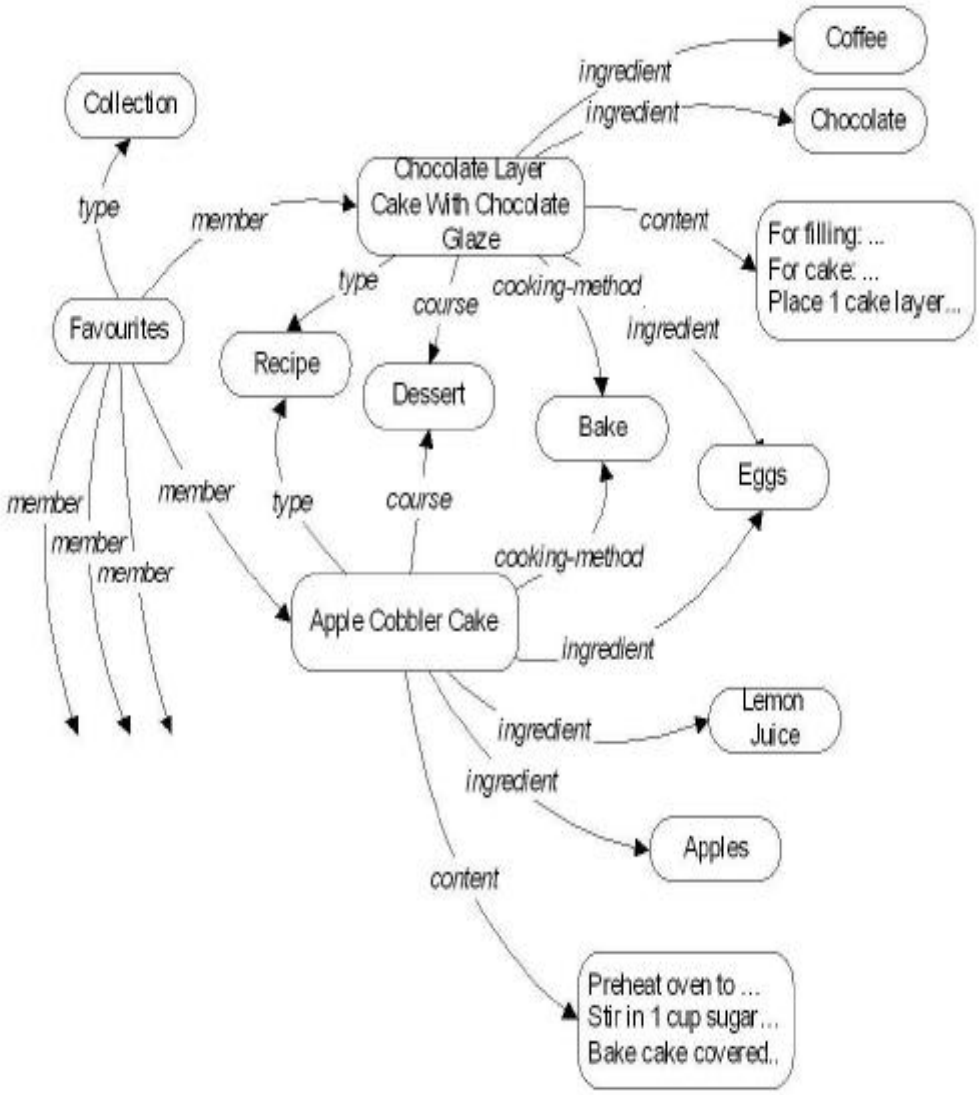
2.2.2 Δημιουργία Clusters

Υπάρχουν πολλά διαφορετικά μοντέλα και αλγόριθμοι ενσωματωμένοι σε διάφορα εργαλεία λογισμικού για τη δημιουργία clusters δεδομένης μιας συλλογής αντικειμένων. Η παρούσα διπλωματική εργασία βασίστηκε στο μοντέλο ***Vector Space Model*** για την υλοποίηση του clustering. Το μοντέλο αυτό είναι το καταλληλότερο για δημιουργία clusters από μια αρχική συλλογή αρχείων κειμένου και περιγράφεται στην υπόλοιπη ενότητα.

2.2.2.1 Βασικές Αρχές

Το Vector Space Model βασίζεται στη δημιουργία διανυσμάτων για κάθε ένα από τα κείμενα της προς ομαδοποίηση συλλογής. Κάθε διάνυσμα έχει τόσα πεδία/συνιστώσες όσες οι διαφορετικές λέξεις στο αντίστοιχο κείμενο, ενώ η τιμή κάθε πεδίου προκύπτει από το πλήθος εμφανίσεων της αντίστοιχης λέξης στο συγκεκριμένο κείμενο. Μια βελτίωση που πραγματοποιείται κατευθείαν με την εφαρμογή του μοντέλου σε μια συλλογή κειμένων, είναι η απομάκρυνση συχνά εμφανιζόμενων λέξεων από τα διανύσματα, οι οποίες δεν προσδίδουν καμία επιπλέον σημασία στο κείμενο. Οι λέξεις αυτές ονομάζονται ***stop-words*** και συνήθως είναι άρθρα, σύνδεσμοι, αριθμητικά και μεταβατικές λέξεις.

Ας εφαρμόσουμε το Vector Space Model σε μια σελίδα με συνταγές, μία για κέικ μήλου και μία για κέικ σοκολάτας. Η διαδικασία φαίνεται στα παρακάτω σχήματα:



Σχήμα 2.6 Γράφος για δημιουργία διανυσμάτων

Αρχικά, όπως γίνεται φανερό από το *σχήμα 2.6,* οι λέξεις κάθε παραγράφου διαβάζονται και με χρήση ειδικών εργαλείων parsing (συνήθως το Lucene), δημιουργούνται ζευγάρια ιδιότητας – τιμής (attribute – value pairs). Για παράδειγμα βλέπουμε τα ζευγάρια για τα συστατικά κάθε συνταγής()ingredients. Το κέικ μήλου αποτελείται από μήλα, χυμό λεμονιού και αυγά, ενώ το κέικ σοκολάτας αποτελείται από καφέ, σοκολάτα και αυγά. Καθένα από αυτά τα ζευγάρια θα αποτελέσει στη συνέχεια ξεχωριστή συνιστώσα όπως φαίνεται στο *σχήμα 2.7*.

Document ID	type: Recipe	course: Main dish	course: Dessert	course: Salad	cooking-method: Bake	cooking-method: Roast	ingredient: Coffee	ingredient: Eggs	ingredient: Lemon Juice	ingredient: Apples	ingredient: Turkey	title: apple	title: vinegar	title: cobbler	title: cake	title: chocolate	title: cider	content: filling	content: cake	content: layer	content: ...	...
Choc-Layer-Cake	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	2	0	1	9	3	...	...
Apple-Cobb-Cake	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0	2	6	0	...	...
Fennel-Apple-Salad	1	0	0	1	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	...	...
Cider-Basted-Turkey	1	1	0	0	0	1	0	0	0	1	1	1	1	0	1	0	1	5	0	0	...	...
Apple-Crunch-Pie	1	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	...	...

Σχήμα 2.7 Αναπαράσταση γράφου με το Vector Space Model

Στην πρώτη γραμμή του σχήματος 2.7 βλέπουμε τα ζευγάρια των οποίων τη δημιουργία προαναφέραμε. Στις επόμενες γραμμές, στην πρώτη στήλη έχουμε τα αντικείμενα προς ταξινόμηση (περισσότερες συνταγές από τις μόνο δύο που απεικονίζονται στο σχήμα 2.6), και στις υπόλοιπες στήλες έχουμε το πλήθος των φορών που κάθε ένα από τα προαναφερθέντα ζευγάρια συναντάται στο κάθε αντικείμενο. Έτσι, τελικά έχουμε για κάθε αντικείμενο που πρόκειται να ταξινομηθεί, το πλήθος των σημαντικών όρων που το αποτελούν.

2.2.2.2 Κανονικοποίηση

Το επόμενο βήμα που ακολουθούμε κατά την εφαρμογή του Vector Space Model, είναι η κανονικοποίηση προκειμένου ο τελικός καθορισμός της ομοιότητας μεταξύ αντικειμένων να είναι όσο γίνεται πιο αντικειμενικός. Αυτό επιτυγχάνεται μετατρέποντας τα διανύσματα σε ίσου μεγέθους για όλα τα κείμενα προς ταξινόμηση. Ο αριθμός συνιστωσών ισούται τώρα με των αριθμό διαφορετικών λέξεων που εμφανίζονται σε όλα τα κείμενα κι έτσι όλα τα διανύσματα έχουν την ίδια σημασία. Χωρίς το βήμα αυτό, διανύσματα που εμφανίζουν

μεγαλύτερη ποικιλία λέξεων θα ήταν πιο ισχυρά από άλλα στα οποία το πλήθος διαφορετικών όρων είναι μικρό, κι έτσι διανύσματα με πολλούς όρους θα έμοιαζαν περισσότερο όμοια μεταξύ τους χωρίς αυτό να αποδίδει την πραγματική ομοιότητα. Τα καινούρια βάρη κάθε όρου καθορίζονται από τους παρακάτω τύπους:

$$\text{term-weight} = \log(\text{freq} + 1.0) \times \log \frac{\text{num-docs}}{\text{num-docs-with-term}}$$

$$\text{normalized-weight} = \frac{\text{term-weight}}{\sqrt{\sum_{\forall \text{term}} \text{term-weight}^2}}$$

2.2.2.3 Καθορισμός ομοιότητας μεταξύ διανυσμάτων

Αφού ολοκληρωθεί η κανονικοποίηση, υπολογίζουμε την ομοιότητα μεταξύ δύο αντικειμένων ως το εσωτερικό γινόμενο των διανυσμάτων των αντικειμένων αυτών. Το εσωτερικό γινόμενο επιλέγεται ως κριτήριο ομοιότητας επειδή, από τον τρόπο με τον οποίο κατασκευάσαμε τα διανύσματα, δύο κείμενα με πολλούς κοινούς όρους θα έχουν μεγάλο εσωτερικό γινόμενο. Γενικεύοντας, για να ελέγξουμε την ομοιότητα ενός κειμένου με μια ομάδα κειμένων, υπολογίζουμε το εσωτερικό γινόμενο του διανύσματος του κειμένου με το "μέσο όρο" διανυσμάτων της ομάδας. Για μεγαλύτερη απόδοση, αντί για το μέσο όρο μπορούν να χρησιμοποιηθούν περισσότερο πολύπλοκοι τύποι και αλγόριθμοι. Αυτό καθορίζεται κάθε φορά από το σχεδιαστή του συστήματος και εξαρτάται από τις ιδιαιτερότητες των εκάστοτε αντικειμένων.

2.2.3 Πλεονεκτήματα

Τα πλεονεκτήματα του clustering είναι τα εξής:

- Πλήρως αυτοματοποιημένη δημιουργία των clusters. Η διαδικασία είναι απλή και γρήγορη και μπορεί να εφαρμοστεί σε οποιαδήποτε συλλογή αρχείων κειμένου.
- Ανάδειξη θεματικών περιοχών ομοιότητας τις οποίες ο σχεδιαστής πιθανώς να μην σκέφτονταν σε περίπτωση που τις καθόριζε εξ' αρχής με το χέρι. Για παράδειγμα, σε μια συλλογή κειμένων αναφερόμενα στην Νέα Ορλεάνη, μπορούμε εύκολα να σκεφτούμε τις θεματικές περιοχές των 'αξιοθέατων', 'πανεπιστημίων' και 'ξενοδοχείων' της πόλης, ή της 'πολιτεία' στην οποία ανήκει. Δε θα σκεφτόμασταν όμως εύκολα τη θεματική περιοχή 'τυφώνας' η οποία προκύπτει λόγω της εμφάνισης του τυφώνα Katrina στη συγκεκριμένη περιοχή.

2.2.4 Μειονεκτήματα

Τα μειονεκτήματα του clustering είναι τα εξής:

- Η δυσκολία να αποδοθούν ετικέτες στα clusters. Τα αυτοματοποιημένα προγράμματα λογισμικού ναι μεν κατηγοριοποιούν τα αντικείμενα με μεγάλη απόδοση, υστερούν όμως στο να ονομάσουν τις κατηγορίες που προκύπτουν. Έχουν γίνει προσπάθειες επίλυσης του προβλήματος αυτού, χωρίς όμως οι ετικέτες να είναι πάντα εκατό τοις εκατό αντιπροσωπευτικές. Αυτό δυσκολεύει τους χρήστες αφού δεν είναι εμφανής σε αυτούς η λογική της κατηγοριοποίησης.
- Ο χρήστης δεν έχει τη δυνατότητα πλοήγησης στην ιεραρχία ούτε μπορεί να θέσει κριτήρια στην αναζήτησή του όπως στην περίπτωση του faceted classification. Οι ομάδες είναι καθορισμένες εξ' αρχής και η αναζήτηση περιορίζεται σε καθεμία από αυτές ξεχωριστά.
- Ο αριθμός των clusters που είτε καθορίζεται αυτόματα από το σύστημα είτε καθορίζεται από το χρήστη (ανάλογα με το λογισμικό που χρησιμοποιείται). Και οι δύο περιπτώσεις παρουσιάζουν αδυναμίες. Στην περίπτωση καθορισμού του αριθμού clusters από το χρήστη, υπάρχει η περίπτωση μη βέλτιστης συνοχής ανάμεσα στα αντικείμενα κάθε cluster, λόγω του ότι ο χρήστης μπορεί να επιλέξει περισσότερα ή λιγότερα clusters σε σχέση με τα ιδανικά. Από την άλλη, στην περίπτωση που το σύστημα αποφασίσει μόνο του για τον αριθμό, ο αριθμός αυτός μπορεί να είναι υπερβολικά μεγάλος, για μεγάλες συλλογές, δημιουργώντας τόσο προβλήματα μνήμης και αποθήκευσης όσο και μπέρδεμα στο χρήστη για τον οποίο δεν είναι εύκολο να εξοικειωθεί με μεγάλο αριθμό κατηγοριών.

2.3 Magnet

Το Magnet είναι ένα σύστημα που υποστηρίζει ταυτόχρονα και τις δύο παραπάνω ταξινομήσεις δεδομένων. Τις ταξινομήσεις αυτές τις εφαρμόζει στα αποτελέσματα κλασσικής keyword αναζήτησης, και για το λόγο αυτό, κρίθηκε σκόπιμο να αναλύσουμε τις βασικές αρχές του έργου αυτού και κυρίως της διαπροσωπείας του (interface).

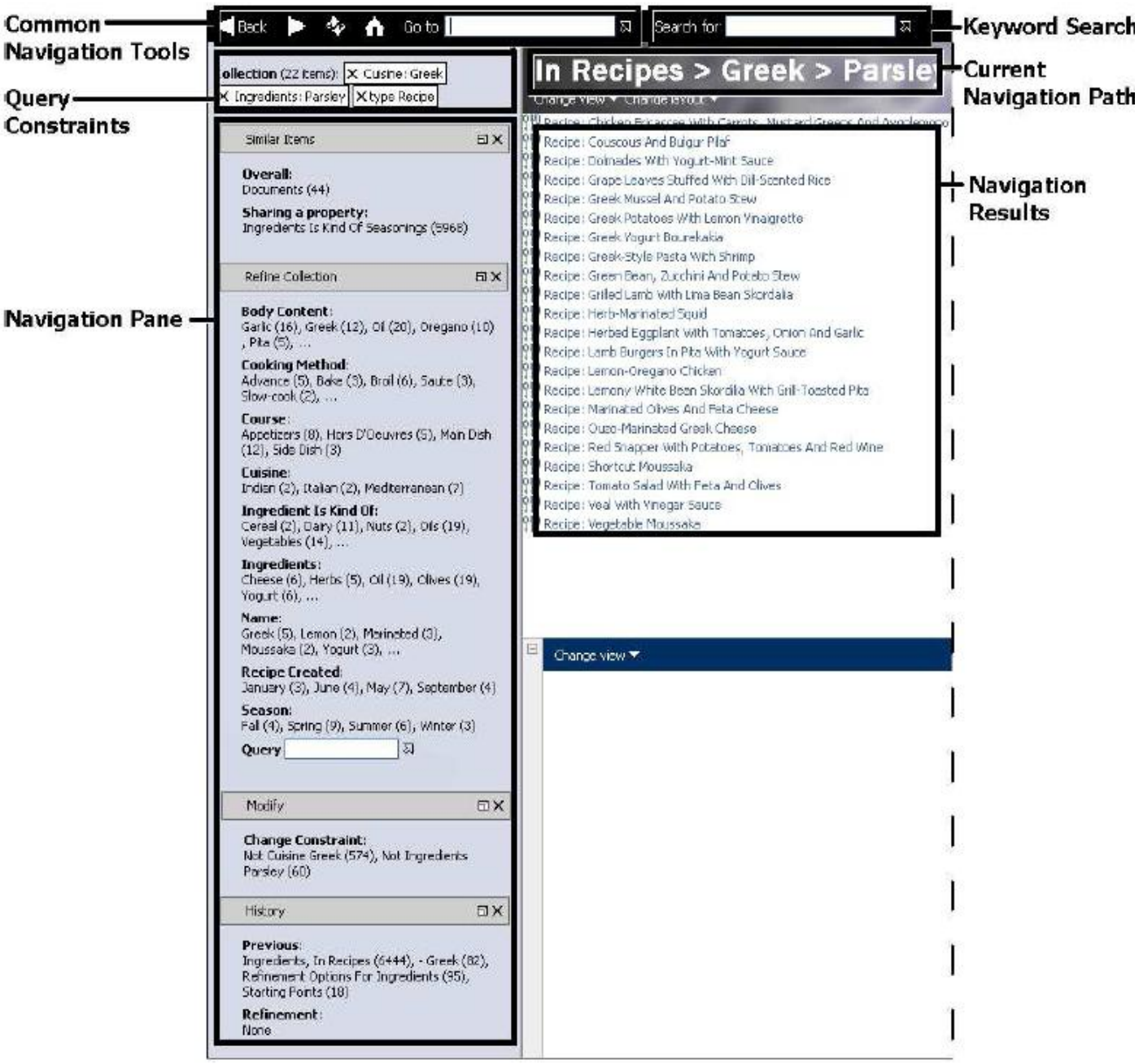
2.3.1 Βασική Ιδέα

Η βασική ιδέα είναι ότι το interface πρέπει να καθοδηγεί το χρήστη πληροφορώντας τον για τις δυνατότητες που του παρέχονται, και σε κάθε βήμα να είναι ορατά σε αυτόν τα πιθανά επόμενα βήματα. Οι διαφορετικές επιλογές που έχει ο χρήστης εμφανίζονται ανά ομάδες κάτω από τον αντίστοιχο Advisor, η ονομασία του οποίου σηματοδοτεί τον τρόπο με τον οποίο ο χρήστης θα επιδράσει στα αποτελέσματα. Έτσι, η αναζήτηση γίνεται πιο εύκολη για

άπειρους χρήστες, ενώ ταυτόχρονα δίνεται η ευκαιρία στους πιο έμπειρους εξ' αυτών να αξιοποιήσουν στο μέγιστο τις δυνατότητες του συστήματος.

2.3.2 Περιγραφή Interface

Οι δυνατότητες που παρέχει το Magnet γίνονται περισσότερο κατανοητές επεξηγώντας τα επιμέρους τμήματα του interface του το οποίο φαίνεται στο *σχήμα 2.8* παρακάτω.



Σχήμα 2.8 Magnet Basic Interface

Στην κορυφή, βλέπουμε τη δυνατότητα για αναζήτηση μέσω λέξεων κλειδιά. Τα αποτελέσματα της αναζήτησης αυτής εμφανίζονται στο δεξί μισό του interface (Navigation Results), ενώ στο αριστερό μισό παρουσιάζονται τα facets στα οποία μπορούν να ταξινομηθούν τα συγκεκριμένα αποτελέσματα, κάτω από τον Advisor "Refine Collection". Στην περίπτωση που το πλήθος των αποτελεσμάτων είναι πάρα πολύ μεγάλο, το interface τροποποιείται παροδικά παίρνοντας τη μορφή του *σχήματος 2.9*.

Και στα δύο σχήματα, δίπλα από κάθε facet value, βλέπουμε τον αριθμό αποτελεσμάτων που το αποτελούν, κι έτσι εξασφαλίζεται ότι ο χρήστης δε θα οδηγηθεί ποτέ σε κενό σύνολο αποτελεσμάτων, γεγονός που αποτελεί πρόβλημα στο faceted classification. Επιλέγοντας ο χρήστης κάποιο από τα facet values, κάνει πιο συγκεκριμένα τα αποτελέσματα και μπορεί να οδηγηθεί πολύ εύκολα σε αυτά ακριβώς που θέλει.

Στην αριστερή στήλη του σχήματος 2.8, εκτός από την παρουσίαση των facets και πάνω από αυτά, παρουσιάζονται και τα αποτελέσματα του clustering. Στον Advisor "Similar Items", δίνεται η δυνατότητα στο χρήστη, να ασχοληθεί με τα αντικείμενα εκείνα της συλλογής, τα οποία παρουσιάζουν ομοιότητα ως προς κάποιο χαρακτηριστικό με αυτά που ήδη έχει ανακτήσει ο χρήστης με τις επιλογές του.

Σχήμα 2.9 Magnet Interface για μεγάλες συλλογές

Στο χρήστη δίνονται επιπλέον οι δυνατότητες να καταργήσει κάποιον από τους περιορισμούς που έχει θέσει (πάνω αριστερό τμήμα του interface, Query Constraints) καθώς και να πάρει ανά πάσα στιγμή το συμπλήρωμα των μέχρι εκείνη τη στιγμή επιλογών του (κάτω αριστερό τμήμα του interface, Modify Advisor – Change Constraint).

Τέλος, παρατηρούμε ότι το σύστημα κρατάει το "ιστορικό" επιλογών του χρήστη (πάνω δεξιό τμήμα του interface, Current Navigation Path).

3

Ανάλυση Απαιτήσεων Συστήματος

Στο κεφάλαιο αυτό περιγράφονται συνοπτικά τα διάφορα υποσυστήματα που αποτελούν το συνολικό σύστημα της διπλωματικής εργασίας, οι επιμέρους λειτουργίες που καθένα επιτελεί και ο τρόπος που αυτά συνδέονται και αλληλεπιδρούν τόσο μεταξύ τους όσο και με το ενιαίο σύστημα. Κατάλληλα διαγράμματα παρατίθενται όπου είναι απαραίτητο, ώστε να γίνουν κατανοητές οι συσχετίσεις αυτές.

3.1 Αρχιτεκτονική

Το σύστημά μας, στήθηκε πάνω στο εργαλείο ανάπτυξης mindmaps που καλείται FreeMind και στις επεκτάσεις που ενσωματώθηκαν σε αυτό από σχετική διπλωματική εργασία με τίτλο "Εργαλείο Συλλογής και Οργάνωσης Γνώσης με Μηχανισμούς Μετα-Αναζήτησης στον Ιστό". Το παραπάνω εργαλείο είναι μια open source εφαρμογή χαρτογράφησης σκέψεων (mind-mapping tool), με τον κώδικά της να είναι διαθέσιμος στην τοποθεσία http://sourceforge.net/projects/freemind/. Με την προαναφερθείσα διπλωματική εργασία ενσωματώθηκαν στην εφαρμογή Freemind οι εξής λειτουργίες:

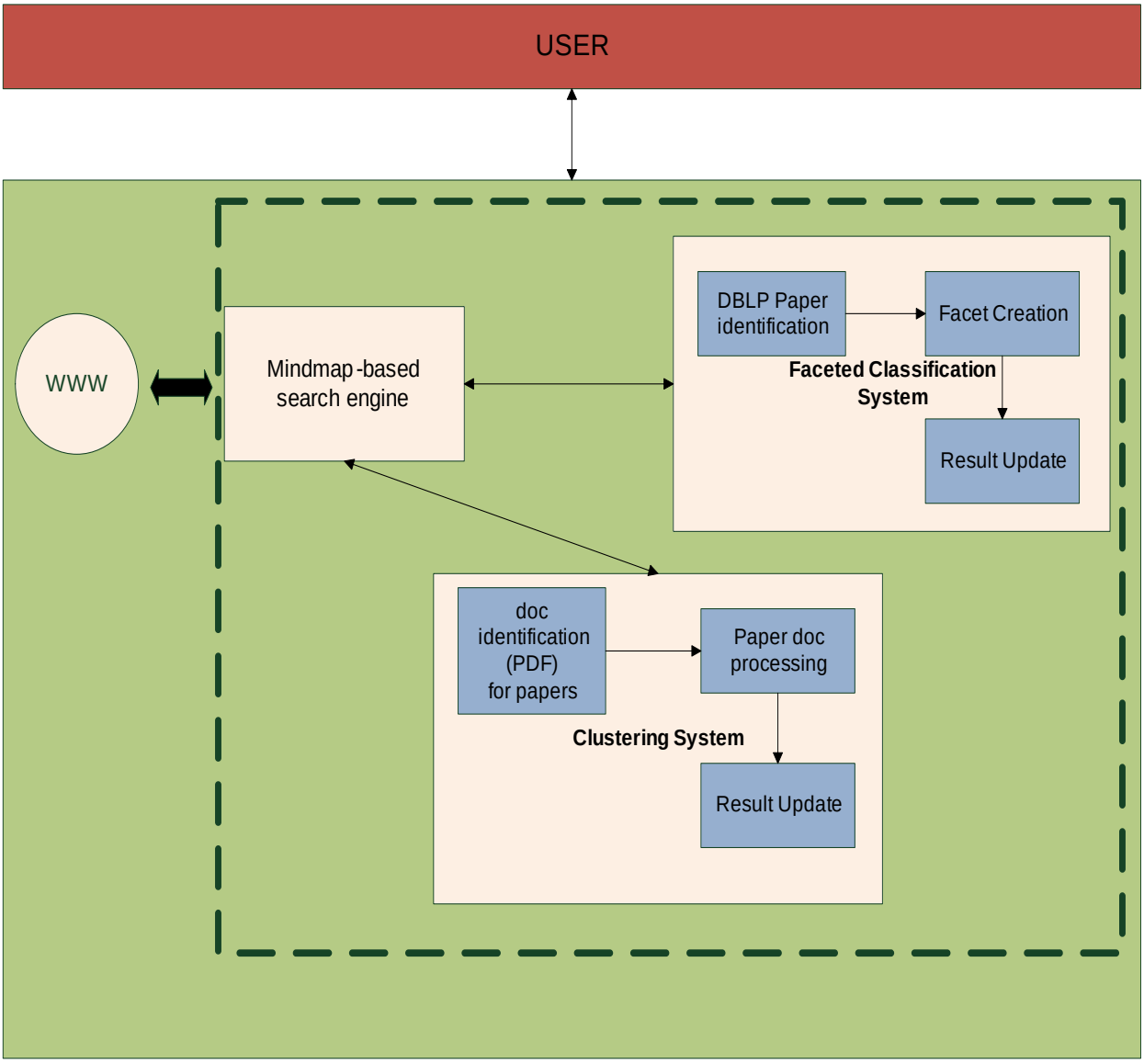
i. Αναζήτηση στον Ιστό, κάνοντας χρήση λέξεων-κλειδιών που υπάρχουν σε κόμβους του mindmap.
ii. Κατάλληλη ταξινόμηση (ranking) των αποτελεσμάτων για μεγαλύτερη ακρίβεια.
iii. Εμπλουτισμός του χάρτη ιδεών με πληροφορίες που ανακτήθηκαν από το Διαδίκτυο.

iv. Περιήγηση στον Παγκόσμιο Ιστό.

v. Ενσωμάτωση πρόσθετων μηχανών αναζήτησης.

Στο συνολικό σύστημα που προέκυψε, ενσωματώθηκαν υποσυστήματα για Faceted Classification και Clustering. Τα υποσυστήματα αυτά παίρνουν τα αποτελέσματα web search αναζήτησης η οποία γίνεται μέσω λέξεων κλειδιά, και στα αποτελέσματα αυτής εφαρμόζουν αντίστοιχα τις δύο παραπάνω ταξινομήσεις. Τα αποτελέσματα των ταξινομήσεων γεμίζουν με τις τιμές τους ειδικά σχεδιασμένα combo-boxes, κι έτσι ο χρήστης έχει τη δυνατότητα πλοήγησης στα αρχικά αποτελέσματα και εντοπισμού εκείνων που ικανοποιούν καλύτερα τις ανάγκες του.

Παρατίθεται το block διάγραμμα του συνολικού συστήματος με τα υποσυστήματα που το αποτελούν και τις βασικές τους λειτουργίες.



Σχήμα 3.1 Block Διάγραμμα Συνολικού Συστήματος

3.2 Περιγραφή Λειτουργιών

Στην ενότητα αυτή περιγράφονται οι λειτουργίες τις οποίες είναι υπεύθυνα να επιτελούν τα επιμέρους υποσυστήματα του συστήματος. Η λειτουργία του συνολικού συστήματος ξεκινάει με τη δημιουργία κάποιου mindmap από το χρήστη, και την αναζήτηση κάποιου όρου από το mindmap στον Ιστό, όπου και εκτελείται προηγμένη αναζήτηση. Τα αποτελέσματα που εμφανίζονται στο χρήστη αποτελούν είσοδο στα υποσυστήματα της παρούσας διπλωματικής εργασίας, στην περίπτωση που ο χρήστης έχει επιλέξει να εξειδικεύσει την αναζήτησή του σε επιστημονικές δημοσιεύσεις (papers).

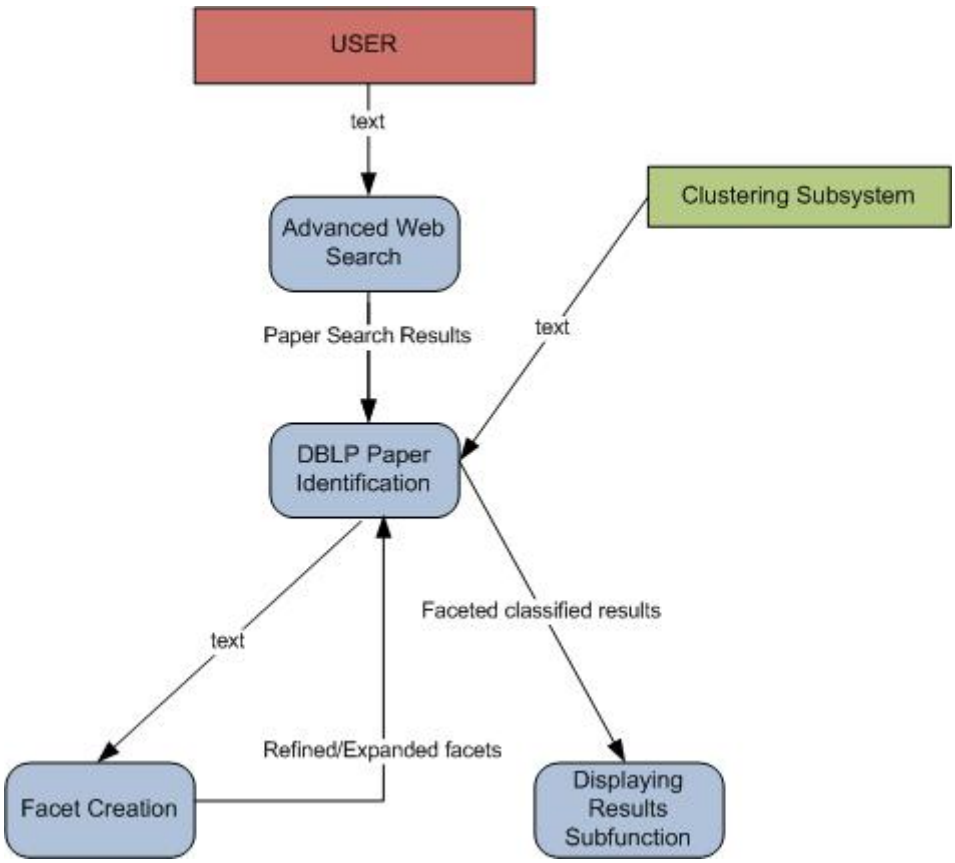
3.2.1 Υποσύστημα Faceted Classification

Το υποσύστημα αυτό είναι γενικά υπεύθυνο για την ταξινόμηση μέσω όψεων των προαναφερθέντων αποτελεσμάτων.

- Αρχικά, τα papers φιλτράρονται προκειμένου να εντοπιστούν όσα από αυτά είναι καταχωρημένα στη βάση δεδομένων του DBLP (Digital Bibliography & Library Project). Το DBLP είναι μια ψηφιακή βιβλιοθήκη η οποία παρέχει πρόσβαση σε επιστημονικές δημοσιεύσεις, μεταπτυχιακές και διδακτορικές διατριβές, και γενικότερα βιβλιογραφία σχετική με την Επιστήμη των Υπολογιστών. Κανείς μπορεί να επισκεφθεί την συγκεκριμένη βάση δεδομένων ακολουθώντας την σελίδα: http://www.informatik.uni-trier.de/~ley/db/. Ο λόγος που χρειάζεται αυτός ο διαχωρισμός των papers που ανήκουν και δεν ανήκουν στο DBLP, είναι διότι το DBLP αποθηκεύει κάθε paper μαζί με πολλά στοιχεία που το χαρακτηρίζουν. Τα στοιχεία αυτά αφορούν τους συγγραφείς, την ημερομηνία συγγραφής, το επιστημονικό περιοδικό ή εφημερίδα στο οποίο έγινε η δημοσίευση ή το αντίστοιχο συνέδριο, τον μοναδικό ISBN αριθμό σε περίπτωση που έχουμε βιβλίο, τους εκδότες, τις σελίδες, το πανεπιστήμιο σε περίπτωση που πρόκειται για μεταπτυχιακή εργασία ή διδακτορική διατριβή κλπ.
- Στη συνέχεια, για τα πεδία ‘συγγραφέας’, ‘ημερομηνία’, για το αν η δημοσίευση έγινε σε επιστημονική εφημερίδα ή συνέδριο καθώς και για τα ονόματα της εφημερίδας ή συνεδρίου αντίστοιχα, δημιουργούμε facets. Τα facet values για κάθε facets προκύπτουν από τις τιμές καθενός από τα ευρισκόμενα στο DBLP papers, για κάθε πεδίο από τα παραπάνω. Στα αυτά τα πεδία των papers έχουμε πρόσβαση λόγω της προηγμένης αναζήτησης που έχει προηγηθεί.
- Με αυτά τα facet values γεμίζουμε ειδικά σχεδιασμένα combo-boxes τα οποία έχουν προστεθεί στο γενικό interface ης εφαρμογής. Δίπλα από κάθε facet value εμφανίζεται ο αριθμός papers που χαρακτηρίζεται από το συγκεκριμένο facet value,

κι έτσι εξασφαλίζουμε ότι ποτέ ο χρήστης δε θα καταλήξει σε κενό σύνολο αποτελεσμάτων.

- Επιπλέον facets δημιουργούνται για τα πεδία ‘λέξεις κλειδιά’ (keywords) και ‘θεματική ενότητα’ (Categories & Subject Descriptors). Οι τιμές των πεδίων αυτών δεν είναι αποθηκευμένες στο DBLP όπως οι τιμές των υπόλοιπων facets αλλά φτάνουν στο υποσύστημα Faceted Classification μέσω του υποσυστήματος Clustering με μια διαδικασία η οποία θα περιγραφεί στην ενότητα 3.2.2.
- Επειδή η επιλογή ενός facet value από κάποιο facet, αποτελεί κριτήριο το οποίο περιορίζει τα εμφανιζόμενα αποτελέσματα, κάθε φορά που ο χρήστης επιλέγει ένα facet value από κάποιο facet, τα facet values των υπόλοιπων facets ανανεώνονται ανάλογα. Για παράδειγμα, έστω ότι αρχικά ο χρήστης επιλέγει το facet ‘ημερομηνία’ (date) και το facet value 2008. Τα εμφανιζόμενα αποτελέσματα μετά από αυτή την επιλογή, θα περιλαμβάνουν μόνο εκείνα τα papers τα οποία δημοσιεύτηκαν το 2008. Αν στη συνέχεια επιλέξει το facet ‘δημοσίευση σε’ (publishedIn) και από εκεί το facet value συνέδρια (conferences), τα αποτελέσματα θα μειωθούν περαιτέρω σε όσα από τα papers που δημοσιεύτηκαν το 2008 δημοσιεύτηκαν σε συνέδριο.



Σχήμα 3.2 Διάγραμμα Ροής για το υποσύστημα Faceted Classification

- Ο χρήστης μπορεί ανά πάσα στιγμή να ακυρώσει κάποιο από τα κριτήρια που έθεσε προηγουμένως επιλέγοντας το facet value “All()” από το αντίστοιχο facet. Με την επιλογή αυτή λαμβάνονται υπ’ όψιν όλα τα facet values του facet κι έτσι ουσιαστικά το κριτήριο ακυρώνεται.

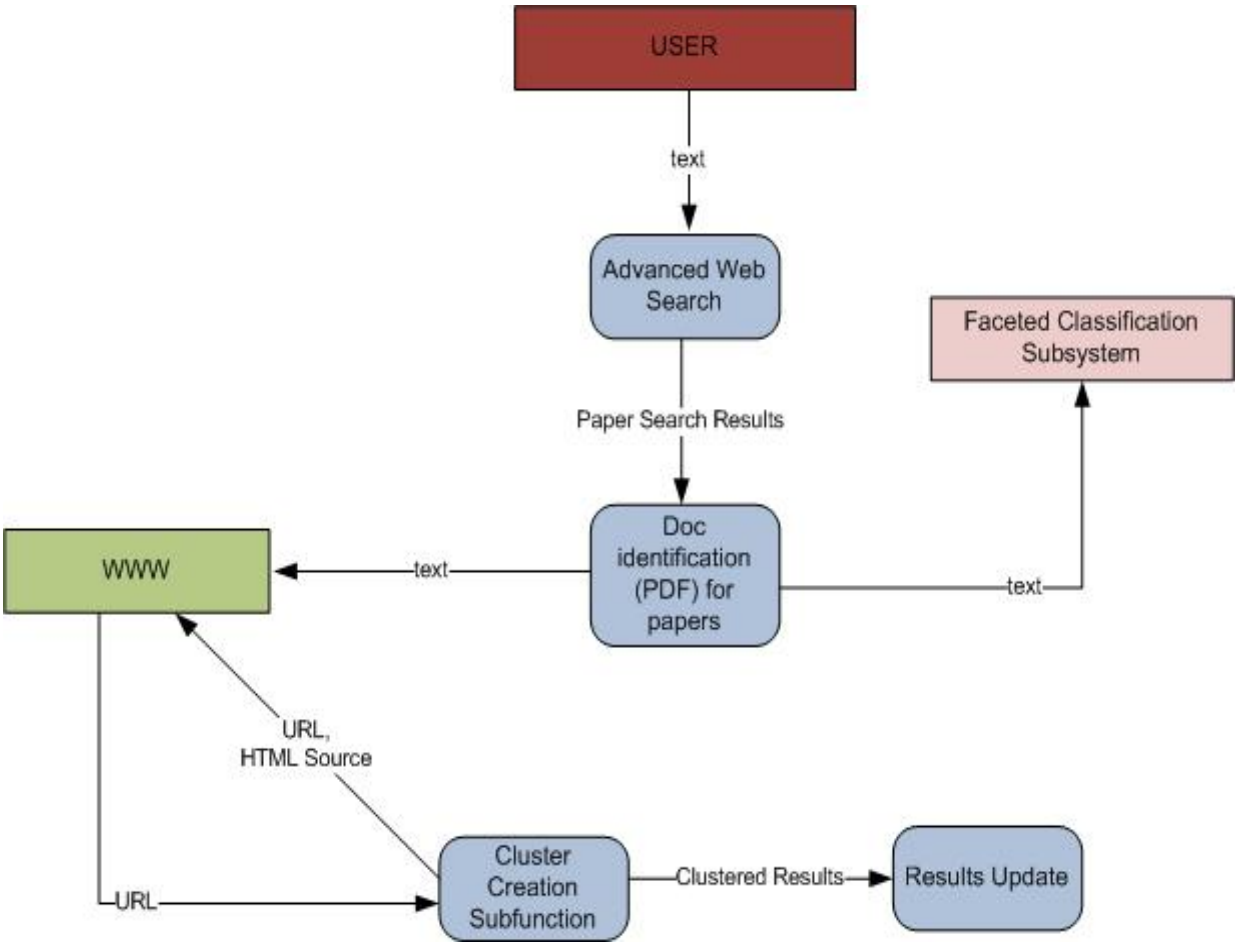
Τα ανωτέρω, φαίνονται και διαγραμματικά στο αντίστοιχο Διάγραμμα Ροής Δεδομένων (Data Flow Diagram) που απεικονίζεται στο σχήμα 3.2.

3.2.2 Υποσύστημα Clustering

Το υποσύστημα αυτό, είναι υπεύθυνο για τη δημιουργία clusters ανάμεσα στα αποτελέσματα. Πιο αναλυτικά:

- Αρχικά, αναζητούμε στο Google τα papers που βρίσκονται στη βάση δεδομένων του DBLP, προκειμένου να ανακτήσουμε το URL στο αντίστοιχο PDF. Το PDF αυτό το ανοίγουμε ως HTML σελίδα, με την επιλογή “View as HTML” που δίνει η μηχανή αναζήτησης Google, και αποθηκεύουμε το source της σελίδας αυτής.
- Από το παραπάνω source, απομονώνουμε τα κομμάτια του κειμένου με τίτλους: “Abstract”, “Categories and Subject Descriptors”, “General Terms” και “Keywords”.
- Τα κομμάτια “Categories and Subject Descriptors” και “General Terms” θα αποτελέσουν ξεχωριστά facets μέσω του υποσυστήματος Faceted Classification.
- Το κομμάτι “Abstract” μαζί με τον εκάστοτε τίτλο, τα “Categories and Subject Descriptors”, “Keywords” και “General Terms” θα χρησιμοποιηθούν για τη δημιουργία clusters με εφαρμογή του αλγορίθμου Vector Space Model που αναλύθηκε στην ενότητα 2.2.2 και χρήση του CLUTO το οποίο έχει ενσωματωθεί στην εφαρμογή. Το CLUTO είναι ένα freeware εργαλείο δημιουργίας clusters ο δικτυακός τόπος του οποίου είναι ο http://glaros.dtc.umn.edu/gkhome/views/cluto/ .
- Τα clusters που θα δημιουργηθούν θα γεμίσουν ένα ξεχωριστό facet μέσω του οποίου ο χρήστης θα μπορεί να ταξινομεί τα αποτελέσματα βάσει αυτών.
- Κάθε επιλογή του χρήστη αποτελεί και εδώ κριτήριο το οποίο περιορίζει τα εμφανιζόμενα αποτελέσματα, επομένως με κάθε επιλογή, έχουμε ανανέωση των facets και του συνόλου αποτελεσμάτων.

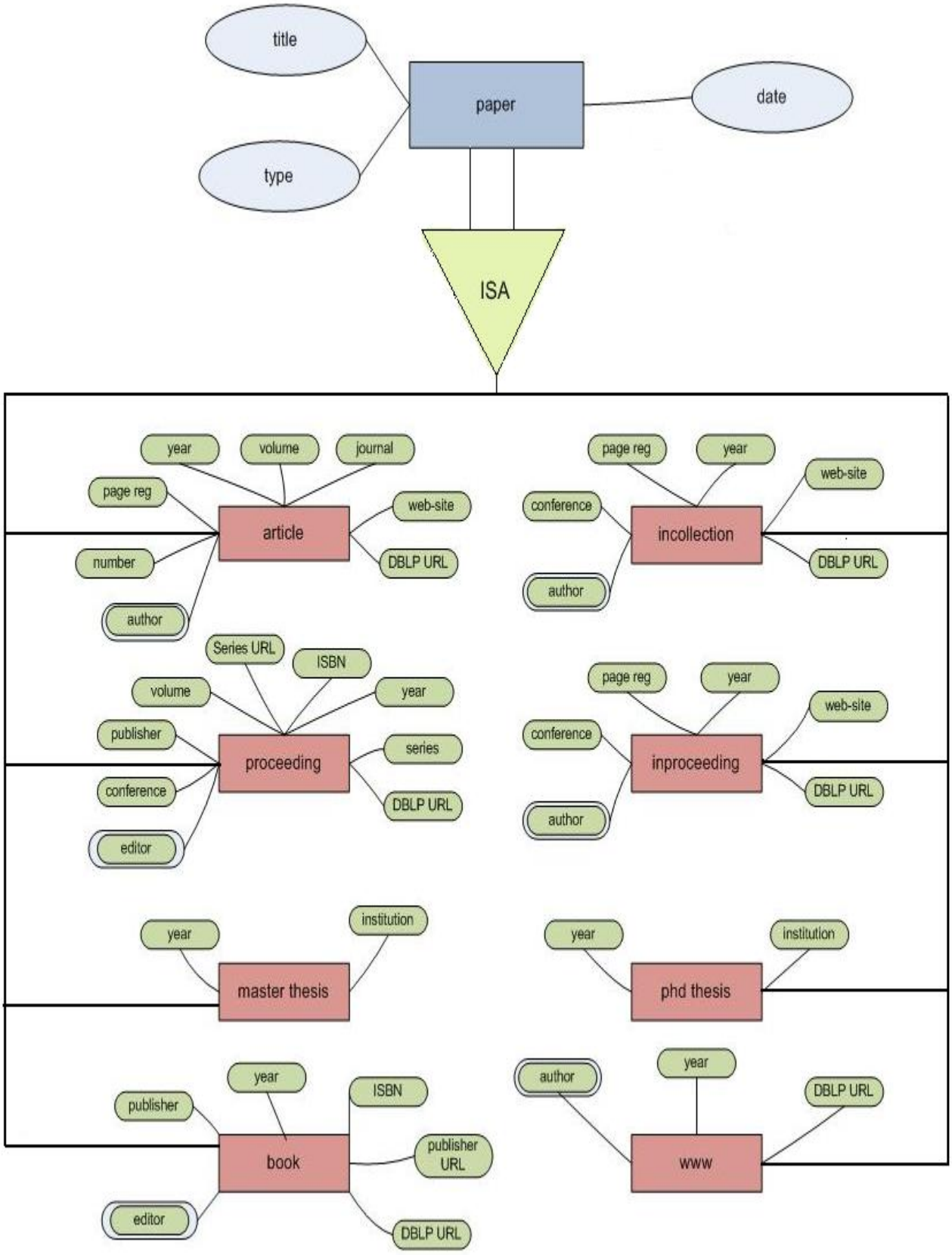
Τα ανωτέρω, φαίνονται και διαγραμματικά στο αντίστοιχο Διάγραμμα Ροής Δεδομένων (Data Flow Diagram) που απεικονίζεται στο σχήμα 3.3 παρακάτω:



Σχήμα 3.3 Διάγραμμα Ροής για το υποσύστημα Clustering

3.3 Μοντέλο Οντοτήτων Συσχετίσεων

Στην παρούσα ενότητα παρουσιάζουμε το Διάγραμμα Οντοτήτων – Συσχετίσεων (E-R model) της βάσης δεδομένων με την οποία αλληλεπιδρά το σύστημά μας:



Σχήμα 3.4 Μοντέλο Οντοτήτων – Συσχετίσεων της Βάσης Δεδομένων

4

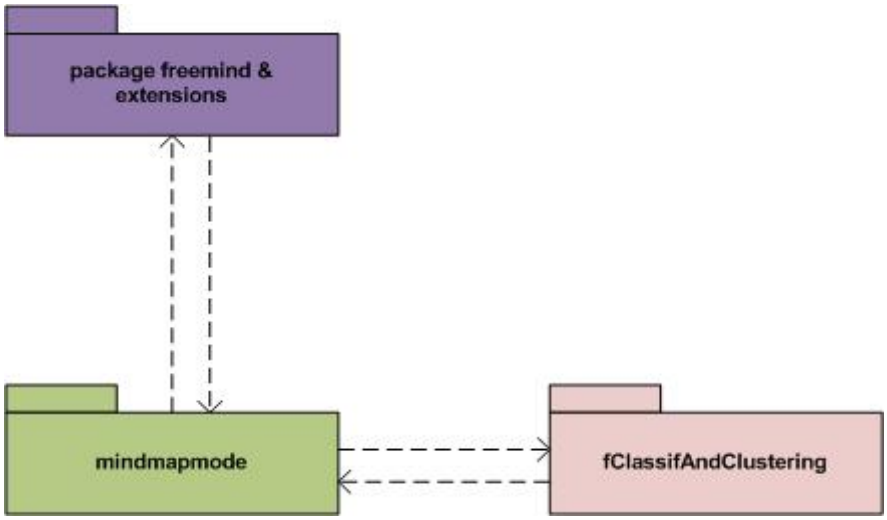
Σχεδίαση Συστήματος

Στο κεφάλαιο αυτό θα ασχοληθούμε με την παρουσίαση της αρχιτεκτονικής του συστήματος. Εστιάζουμε στην περιγραφή των κλάσεων που δημιουργήθηκαν (οι οποίες αποτελούν ουσιαστικά την υλοποίηση των επεκτάσεων της εφαρμογής) και παραθέτουμε τα αντίστοιχα block διαγράμματα. Επίσης, δίνονται και κάποιες λεπτομέρειες σχετικά με την Βάση Δεδομένων, η οποία συμπεριλαμβάνεται στο σύστημά μας.

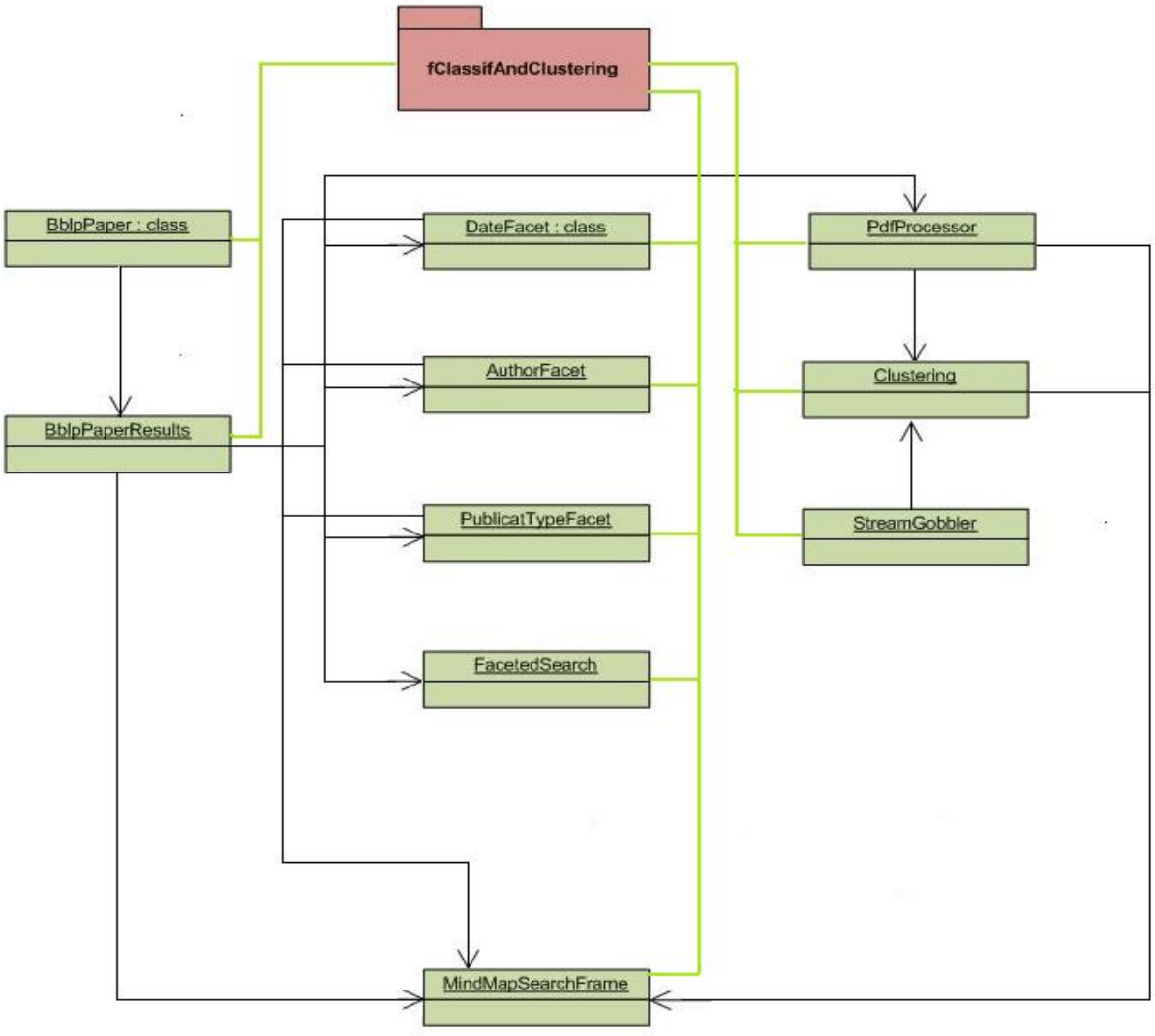
4.1 Αρχιτεκτονική

Όπως ήδη έχει αναφερθεί σε προηγούμενο κεφάλαιο, η ανάπτυξη την εφαρμογής βασίστηκε στο open source εργαλείο FreeMind και σε επεκτάσεις που ενσωματώθηκαν σε αυτό με τη διπλωματική εργασία "Εργαλείο Συλλογής και Οργάνωσης Γνώσης με Μηχανισμούς Μετα-Αναζήτησης στον Ιστό". Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η αντικειμενοστρεφής γλώσσα Java. Για τις ανάγκες τις εφαρμογής, δημιουργήθηκε ένα καινούριο πακέτο κλάσεων, το *FClassifAndClustering*, το οποίο περιλαμβάνει όλες τις καινούριες κλάσεις που δημιουργήθηκαν και το οποίο αλληλεπιδρά με ήδη υπάρχοντα πακέτα. Εκτός από τις κλάσεις που δημιουργήθηκαν εξ' αρχής, τροποποιήθηκαν και ήδη υπάρχουσες κλάσεις. Παρακάτω στην ίδια ενότητα, ακολουθεί περιγραφή όλων αυτών των κλάσεων.

Τα block διαγράμματα των πακέτων κλάσεων της εφαρμογής και των κλάσεων του νέου πακέτου παρατίθενται στη συνέχεια:



Σχήμα 4.1 Γενικό Block Διάγραμμα των πακέτων κλάσεων του συστήματος



Σχήμα 4.2 Block Διάγραμμα των κλάσεων του πακέτου FClassifAndClustering

4.1.1 Κλάσεις Επεξεργασίας Papers

Οι κλάσεις αυτές είναι οι DblpPaper και DblpPaperResults και είναι υπεύθυνες για τον εντοπισμό των papers που ανήκουν στο DBLP καθώς επίσης και για να φέρουν την πληροφορία που αφορά κάθε paper σε μορφή πίνακα ώστε να διευκολυνθεί η μετέπειτα επεξεργασία τους.

4.1.2 Κλάσεις Δημιουργίας και Λειτουργίας Facets

Οι κλάσεις αυτές είναι οι DateFacet, AuthorFacet, PublicationTypeFacet, GeneralTermsFacet, CategoryFacet και FacetedSearch. Οι τρεις πρώτες παίρνουν πληροφορίες που αφορούν τα papers του DBLP και δημιουργούν τα αντίστοιχα facets. Οι δύο επόμενες αντλούν τις απαιτούμενες πληροφορίες κατευθείαν από τα PDFs αλληλεπιδρώντας με τις κλάσεις της ενότητας 4.1.3 και δημιουργούν τα αντίστοιχα facets. Όλες μαζί είναι υπεύθυνες για το "γέμισμα" των ComboBoxes μέσω των οποίων εμφανίζονται τα facet values. Η κλάση FacetedSearch αφορά τη συνεργασία των facets έτσι ώστε όταν επιλέγεται η τιμή ενός να αλλάζουν κατάλληλα οι τιμές των υπόλοιπων.

4.1.3 Κλάσεις για ειδικό χειρισμό PDF

Στην κατηγορία αυτή έχουμε μόνο μια κλάση, την PdfProcessor η οποία αφορά τον ειδικό χειρισμό των PDFs. Προκειμένου να αυξήσουμε τις δυνατότητες της εφαρμογής και τις επιλογές ταξινόμησης που παρέχονται στο χρήστη, δεν αρκούμαστε μόνο στις πληροφορίες που αντλούμε από τη βάση δεδομένων του DBLP, αλλά γίνεται προσπάθεια άντλησης πληροφοριών από τα ίδια τα PDFs των papers που μας αφορούν.

4.1.4 Κλάσεις Δημιουργίας Clusters

Εκτός από τη δημιουργία facets, παρέχεται στο χρήστη και η επιλογή του clustering. Τρεις κλάσεις ασχολούνται με τη διαδικασία αυτή: η FileIO που αφορά τον χειρισμό των βοηθητικών txt αρχείων, η StreamGobbler που αφορά στον χειρισμό των εκτελέσιμων προγραμμάτων που βρίσκονται εξωτερικά της Java και η Clustering που είναι η κύρια κλάση. Με κατάλληλους αλγορίθμους και μεθόδους, τα papers ταξινομούνται με βάση τη σχετικότητα του περιεχομένου τους σε ομάδες, τον αριθμό των οποίων καθορίζει ο χρήστης.

4.1.5 Διεπαφή με το χρήστη

Για να φτιάξουμε τη διεπαφή με το χρήστη, επεμβήκαμε στην ήδη υπάρχουσα κλάση MindMapSearchFrame και την επεκτείναμε. Η κλάση αυτή υλοποιεί το βασικό interface της εφαρμογής και αρχικά αφορούσε μόνο την αναζήτηση όρων από το χάρτη σκέψεων ή λέξεων

κλειδιών στον Ιστό. Με τις επεκτάσεις που προστέθηκαν, η διεπαφή προσφέρει επιπλέον στο χρήστη επιλογές ταξινόμησης των αποτελεσμάτων τόσο μέσω Faceted Classification όσο και μέσω Clustering.

4.2 Περιγραφή Κλάσεων

Στην ενότητα αυτή περιγράφονται συνοπτικά οι λειτουργίες που επιτελεί κάθε κλάση του πακέτου *FClassifAndClustering* της εφαρμογής. Κάθε κλάση περιλαμβάνει μια μέθοδο constructor στην οποία δε θα γίνεται αναφορά, αφού η ύπαρξή της εξυπακούεται, εκτός αν σε αυτή γίνονται ειδικές αρχικοποιήσεις. Για τις υπόλοιπες κλάσεις, δεν παραθέτουμε κώδικα μαζί με την περιγραφή.

4.2.1 Κλάση MindMapSearchFrame

Η κλάση MindMapSearchFrame είναι υπεύθυνη για τη διεπαφή χρήστη – συστήματος δίνοντας στον χρήστη τόσο επιλογές αναζήτησης όσο και επιλογές ταξινόμησης των αποτελεσμάτων. Οι προσθήκες που έγιναν στην κλάση για την εν λόγω εφαρμογή, αφορούν το διαχωρισμό των papers από τα υπόλοιπα αποτελέσματα της αναζήτησης, την κλήση σε κατάλληλο σημείο όλων των κλάσεων που περιγράφονται στη συνέχεια, και τη δημιουργία ComboBoxes τα οποία γεμίζουν με τα facet values των διαφορετικών facets και του clustering. Επίσης, προστέθηκε στη διεπαφή ένα TextField στο οποίο ο χρήστης καθορίζει τον αριθμό των clusters που θέλει να δημιουργηθούν.

4.2.2 Κλάση DblpPaper

Η κλάση αυτή περιέχει μια μοναδική μέθοδο, την createDblpPaper. Η μέθοδος αυτή δέχεται ως είσοδο ένα paper του οποίου τα πεδία έχουν αποθηκευτεί σε string χωριζόμενα από τη λέξη “splitchar” και μετατρέπει το string σε πίνακα μεγέθους 18, όσα δηλαδή είναι και τα fields του paper που παρέχει το DBLP.

4.2.3 Κλάση DblpPaperResults

Η κλάση αυτή περιέχει μια μοναδική μέθοδο, την insert. Η μέθοδος αυτή παίρνει σα παράμετρο εισόδου έναν πίνακα από strings, όπου κάθε string είναι κι ένα αποτέλεσμα της συνολικής δικτυακής αναζήτησης, κι ένα ArrayList από ακεραίους οι οποίοι είναι δείκτες στον πίνακα από strings στις θέσεις εκείνες όπου τα αποτελέσματα είναι papers. Καλώντας τη συνάρτηση createDblpPaper της κλάσης DblpPaper, επιστρέφει ένα ArrayList από πίνακες strings καθένας από τον οποίο αναπαριστά ένα paper.

4.2.4 Κλάση DateFacet

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία του facet ‘ημερομηνία’. Περιέχει τέσσερις μεθόδους οι οποίες περιγράφονται αναλυτικά παρακάτω:

i) ***Μέθοδος createDateFacet***: η μέθοδος αυτή δημιουργεί μια δομή Hashtable με τα papers του DBLP. Κάθε εγγραφή της δομής αυτής, είναι ένα ζευγάρι από ένα string το οποίο αντιστοιχεί κάθε φορά σε μια από τις διαφορετικές ημερομηνίες που υπάρχουν στα papers, κι ένα ArrayList από ακεραίους οι οποίο αντιστοιχούν στις θέσεις του πίνακα με τα συνολικά papers, στις οποίες βρίσκονται τα papers με τη συγκεκριμένη ημερομηνία.

ii) ***Μέθοδος dateHashtableSort***: η μέθοδος αυτή παίρνει μια δομή Hashtable και την ταξινομεί σε αύξουσα σειρά. Η ταξινόμηση γίνεται με τη βοήθεια μιας δομής Vector, η οποία και επιστρέφεται από τη μέθοδο. Η μέθοδος αυτή χρησιμοποιείται σε συνδυασμό με την προηγούμενη προκειμένου να έχουμε ταξινομημένες σε μια δομή τις διαφορετικές ημερομηνίες που αποτελούν το facet, δηλαδή τα facet values.

iii) ***Μέθοδος displayDateHashValues***: η μέθοδος αυτή είναι υπεύθυνη για την εμφάνιση των τιμών του facet ‘ημερομηνία’ στο αντίστοιχο ComboBox.

iv) ***Μέθοδος displayInitialDateFacetValues:*** η μέθοδος αυτή είναι υπεύθυνη για την εμφάνιση των τιμών του facet ‘ημερομηνία’ στο αντίστοιχο ComboBox, καλώντας την προηγούμενη συνάρτηση. Επιπλέον εμφανίζει στο ComboBox την επιλογή All(), η οποία είναι για να ληφθούν υπ’ όψιν όλα τα facet values αυτού του facet, και την επιλογή other() η οποία αναφέρεται σε όσα αποτελέσματα του web search δεν είναι papers.

4.2.5 Κλάση AuthorFacet

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία του facet ‘συγγραφέας’. Περιέχει τέσσερις μεθόδους οι οποίες είναι παρόμοιες με τις μεθόδους της κλάσης DateFacet, και περιγράφονται παρακάτω:

i) Μέθοδος createAuthorFacet: η μέθοδος αυτή είναι αντίστοιχη της createDateFacet της κλάσης DateFacet, και δημιουργεί την αντίστοιχη δομή Hashtable με τα papers του DBLP όπως περιγράφθηκε προηγουμένως.

ii) Μέθοδος authorHashtableSort: η μέθοδος αυτή είναι αντίστοιχη της dateHashtableSort της κλάσης DateFacet, και ταξινομεί μια δομή Hashtable όπως περιγράφθηκε προηγουμένως.

*iii) **Μέθοδος displayAuthorHashValues:*** η μέθοδος αυτή είναι αντίστοιχη της displayDateHashValues της κλάσης DateFacet, και εμφανίζει τα facet values στο αντίστοιχο ComboBox όπως περιγράφθηκε προηγουμένως.

*iv) **Μέθοδος displayInitialAuthorFacetValues:*** η μέθοδος αυτή είναι αντίστοιχη της displayInitialDateFacetValues της κλάσης DateFacet, και εμφανίζει τις επιλογές που περιγράφθηκε προηγουμένως στο αντίστοιχο ComboBox.

4.2.6 Κλάση PublicationTypeFacet

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία των facets 'δημοσίευση σε' και 'τύπος δημοσίευσης'. Το facet 'τύπος δημοσίευσης' ξεχωρίζει τα papers σε αυτά που δημοσιεύτηκαν σε συνέδρια και σε εκείνα που δημοσιεύτηκαν σε επιστημονικές εφημερίδες ή περιοδικά. Το facet 'δημοσίευση σε' αναφέρεται στα ονόματα των συνεδρίων ή εφημερίδων/περιοδικών στα οποία έγιναν οι δημοσιεύσεις. Περιέχει έξι μεθόδους οι οποίες είναι παρόμοιες με τις μεθόδους των παραπάνω κλάσεων, και περιγράφονται στη συνέχεια:

*i) **Μέθοδος createPublicationFacet:*** η μέθοδος αυτή είναι αντίστοιχη της createDateFacet της κλάσης DateFacet, και δημιουργεί την αντίστοιχη δομή Hashtable με τιμές τα ονόματα των συνεδρίων και των επιστημονικών εφημερίδων/περιοδικών των papers του DBLP όπως περιγράφθηκε προηγουμένως. Επιπλέον, σε σχέση με τις προηγούμενες αντίστοιχες μεθόδους, δημιουργεί και δύο ArrayLists<Integer>. Το ένα περιέχει δείκτες στα papers τα οποία έχουν δημοσιευτεί σε συνέδρια, και το άλλο σε αυτά που έχουν δημοσιευτεί σε εφημερίδα/ περιοδικό. Με βάση τις πληροφορίες αυτών των ArrayLists γεμίζει το facet 'τύπος δημοσίευσης' (publicationType).

*ii) **Μέθοδος publicationHashtableSort:*** η μέθοδος αυτή είναι αντίστοιχη της dateHashtableSort της κλάσης DateFacet, και ταξινομεί το Hashtable όπως αναφέρθηκε παραπάνω.

*iii) **Μέθοδος displayPublishedInHashValues:*** η μέθοδος αυτή είναι αντίστοιχη της displayDateHashValues της κλάσης DateFacet, και εμφανίζει τα facet values στο αντίστοιχο ComboBox όπως περιγράφθηκε προηγουμένως.

*iv) **Μέθοδος displayInitialPublishedInFacetValues:*** η μέθοδος αυτή είναι αντίστοιχη της displayInitialDateFacetValues της κλάσης DateFacet, και εμφανίζει τις επιλογές που περιγράφθηκε προηγουμένως στο αντίστοιχο ComboBox.

*v) **Μέθοδος displayPublicationTypeHashValues:*** η μέθοδος αυτή εμφανίζει τα βασικά facet values του facet 'τύπος δημοσίευσης' στο αντίστοιχο ComboBox, τα οποία

είναι ο αριθμός των papers που δημοσιεύονται σε συνέδριο και ο αριθμός αυτών που δημοσιεύονται σε επιστημονικά έντυπα.

vi) ***Μέθοδος displayPublicationTypeFacetValues:*** η μέθοδος αυτή εμφανίζει τα βασικά facet values όπως η προηγούμενη, μαζί με τις επιλογές "All()" και "other()" οι οποίες έχουν επεξηγηθεί πιο πάνω.

4.2.7 Κλάση GeneralTermsFacet

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία του facet 'γενικοί όροι'. Περιέχει τέσσερις μεθόδους οι οποίες είναι αντίστοιχες με τις μεθόδους της κλάσης DateFacet και για το λόγο αυτό η περιγραφή τους παραλείπεται.

4.2.8 Κλάση CategoryFacet

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία του facet 'θεματική ενότητα'. Περιέχει τέσσερις μεθόδους οι οποίες είναι αντίστοιχες με τις μεθόδους της κλάσης DateFacet και για το λόγο αυτό η περιγραφή τους παραλείπεται.

4.2.9 Κλάση FacetedSearch

Η κλάση αυτή είναι υπεύθυνη για την αλληλεπίδραση των facets μεταξύ τους. Όπως έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο, η επιλογή ενός facet value ενός facet από το χρήστη, περιορίζει τα αποτελέσματα σε όσα μόνο πληρούν την επιλεχθείσα συνθήκη. Όσο ο χρήστης επιλέγει facet values, τα facet values των υπόλοιπων facets πρέπει να ανανεώνονται έτσι ώστε ο χρήστης να έχει προς επιλογή μόνο έγκυρες τιμές και ως εκ τούτου να μην είναι δυνατόν να οδηγηθεί σε κενό σύνολο αποτελεσμάτων. Η κλάση αποτελείται από δύο μεθόδους οι οποίες περιγράφονται στη συνέχεια.

i) ***Μέθοδος initialize:*** η μέθοδος αυτή είναι υπεύθυνη για την αρχικοποίηση των διαφόρων μεταβλητών κλάσης και είναι βοηθητική ώστε να εισαχθούν προς επεξεργασία στην κλάση, δομές όπως Hashtables και ArrayLists οι οποίες αφορούν τη δημιουργία των facets και παράγονται από τις κλάσεις που έχουμε ως τώρα αναφέρει.

ii) ***Μέθοδος createResults:*** η μέθοδος αυτή είναι η κύρια μέθοδος της παρούσας κλάσης και είναι υπεύθυνη για την αλληλεπίδραση των facets μεταξύ τους. Η κλάση αυτή ξεχωρίζει τρεις διαφορετικές περιπτώσεις επιλογών του χρήστη και χειρίζεται διαφορετικά την καθεμία από αυτές:

1η περίπτωση: ο χρήστης επιλέγει ένα facet value για πρώτη φορά. Στην περίπτωση αυτή, εκμεταλλευόμαστε τις δομές Hashtables που έχουμε δημιουργήσει σε προηγούμενες κλάσεις και οι οποίες έχουν εισαχθεί στην παρούσα. Έτσι ψάχνοντας στην κατάλληλη δομή Hashtable έχουμε απευθείας τη λίστα των νέων αποτελεσμάτων. Για παράδειγμα, ας υποθέσουμε ότι το πρώτο κριτήριο του χρήστη είναι το facet value 2005 από το facet 'ημερομηνία'. Ψάχνοντας την τιμή 2005 στο Hashtable με τις ημερομηνίες, θα βρούμε όλα τα papers που έχουν ημερομηνία συγγραφής 2005.

2η περίπτωση: ο χρήστης εξακολουθεί να προσθέτει κριτήρια. Στην περίπτωση αυτή, θα ξεκινήσουμε από τον πιο πρόσφατο πίνακα αποτελεσμάτων που έχουμε, και θα ελέγξουμε αν καθένα από τα στοιχεία αυτού ικανοποιεί το κριτήριο που μόλις έθεσε ο χρήστης. Τα papers που το ικανοποιούν, αποτελούν τη νέα λίστα αποτελεσμάτων. Συνεχίζοντας το παραπάνω παράδειγμα, ας υποθέσουμε ότι ο χρήστης προσθέτει το κριτήριο author = 'S. R. Ranganathan'. Για τα στοιχεία της λίστας των papers που έχουν ημερομηνία δημοσίευσης 2005, ελέγχω σε πόσα κάποιος από τους συγγραφείς είναι ο S. R. Ranganathan και αυτά δημιουργούν την καινούρια λίστα αποτελεσμάτων.

3η περίπτωση: ο χρήστης αλλάζει τα μέχρι τώρα κριτήρια που έχει θέσει είτε επιλέγοντας "All()" από κάποιο facet, οπότε αφαιρεί ένα από τα κριτήρια, είτε αλλάζει το facet value από κάποιο facet. Στην περίπτωση αυτή, ξεκινάμε από την αρχική λίστα των συνολικών αποτελεσμάτων και για κάθε paper ελέγχουμε αν πληροί όλα τα κριτήρια που έχουν συνολικά τεθεί. Ας γυρίσουμε στο παραπάνω παράδειγμα και ας υποθέσουμε ότι ο χρήστης είχε προσθέσει ένα επιπλέον κριτήριο, το publishedIn = 'conference' και στη συνέχεια άλλαξε την ημερομηνία από 2005 σε 2009. Για να υπολογίσουμε το σωστό σύνολο από papers, θα ξεκινήσουμε από τα συνολικά ελέγχοντας ποια από αυτά πληρούν τα κριτήρια: date = '2009', author = 'S. R. Ranganathan' και publishedIn = 'conference'. Αντίστοιχα λειτουργεί η συνάρτηση αν αφαιρεθεί τελείως κάποιο από τα προηγούμενα κριτήρια.

4.2.10 Κλάση PdfProcessor

Η κλάση αυτή είναι υπεύθυνη για τον ειδικό χειρισμό των PDFs. Η μηχανή αναζήτησης Google έχει την ιδιότητα να κρατάει τα PDFs και σε HTML μορφή και να δίνει τη δυνατότητα στο χρήστη να ανοίξει όποια από τις δύο μορφές τον εξυπηρετεί. Την ιδιότητα

αυτή, την εκμεταλλευόμαστε, ώστε εκτός από τις πληροφορίες που παίρνουμε από τη βάση δεδομένων του DBLP για τα papers, και με τις οποίες φτιάχνουμε τα ήδη αναφερθέντα facets, να μπορέσουμε να εξάγουμε και επιπλέον πληροφορίες από τον HTML κώδικα του PDF για όσα από αυτά βρεθούν στον Ιστό σε αυτή τη μορφή. Οι επιπλέον αυτές πληροφορίες θα βελτιώσουν και θα επεκτείνουν το παρόν σύστημα ταξινόμησης. Κάποιες από αυτές θα χρησιμοποιηθούν για τη δημιουργία επιπλέον facets και άλλες για τη διαδικασία του Clustering στην υλοποίηση της οποίας θα αναφερθούμε αναλυτικά παρακάτω. Η κλάση αποτελείται από τρεις μεθόδους:

i) ***Μέθοδος googleReplies:*** Η μέθοδος αυτή είναι υπεύθυνη για την εύρεση του URL που οδηγεί στην HTML σελίδα του PDF. Αρχικά, θεωρώντας ως λέξεις κλειδιά τον τίτλο και τους συγγραφείς αναζητούμε στο Google το εκάστοτε paper. Η αναζήτηση γίνεται φτιάχνοντας το κατάλληλο URL με τρόπο ο οποίος περιγράφεται αναλυτικά στο κεφάλαιο 5. Στη συνέχεια, αποθηκεύουμε σε μια μεταβλητή string, το source της σελίδας αποτελεσμάτων του Google. Γνωρίζοντας ότι βάση της αναζήτησης το PDF που ζητάμε, εάν υπάρχει, επιστρέφεται πρώτο από το Google και χρησιμοποιώντας τον parser Jericho, καταφέρνουμε να απομονώσουμε το URL της HTML σελίδας του PDF. Το URL αυτό επιστρέφεται από τη συνάρτηση.

ii) ***Μέθοδος htmlInformationRetrieval:*** Η μέθοδος αυτή είναι υπεύθυνη για την απομόνωση των επιθυμητών πληροφοριών από το εκάστοτε PDF. Ξεκινώντας από το URL το οποίο αντιστοιχεί στο HTML source του PDF, θα καταλήξουμε στο κομμάτι του source που περιέχει την πληροφορία που μας ενδιαφέρει. Καθώς διατρέχουμε το source με τις κατάλληλες συναρτήσεις που παρέχει η Java, εκτελούμε ελέγχους ώστε μόνο το κομμάτι κειμένου που αναφέρεται στο "Abstract" στα "Categories and Subject Descriptors" στα "General Terms" και στα "Keywords" να αποθηκευτούν σε μεταβλητή String. Στη συνέχεια, επεξεργαζόμαστε κατάλληλα τη μεταβλητή αυτή και τελικά καθένα από τα προαναφερθέντα πεδία του PDF γεμίζει μια θέση στον πίνακα από string με όνομα returnString ο οποίος έχει μέγεθος τέσσερα και επιστρέφεται τελικά από τη συνάρτηση.

iii) ***Μέθοδος htmlViewHandler:*** Η μεθοδός αυτή είναι η βασική μέθοδος της κλάσης. Για κάθε paper, απομονώνει τους τίτλους και τους συγγραφείς και στη συνέχεια, καλεί διαδοχικά τις παραπάνω συναρτήσεις. Αρχικά ο τίτλος και οι συγγραφείς αποτελούν είσοδο για τη μέθοδο googleReplies η οποία θα επιστρέψει, αν υπάρχει, το URL για την HTML μορφή του PDF. Αν το URL υπάρχει, θα

αποτελέσει είσοδο για τη μέθοδο htmlInformationRetrieval η οποία θα επιστρέψει τελικά τις πληροφορίες προς επεξεργασία.

4.2.11 Κλάση FileIO

Η κλάση αυτή αφορά στον χειρισμό των αρχείων .txt και είναι βοηθητική κλάση για τη διαδικασία του Clustering. Περιέχει τρεις μεθόδους:

i) ***Μέθοδος getContents:*** Δέχεται σαν όρισμα το όνομα ενός αρχείου και επιστρέφει σε μορφή string το περιεχόμενο του συγκεκριμένου αρχείου.

ii) ***Μέθοδος setContents:*** Δέχεται σαν όρισμα το όνομα ενός αρχείου και ένα text σε μορφή string, και γράφει το text στην αρχή του συγκεκριμένου αρχείου. Σε περίπτωση που στο αρχείο υπάρχουν προηγούμενες εγγραφές, αυτές χάνονται.

iii) ***Μέθοδος append:*** Δέχεται σαν όρισμα το όνομα ενός αρχείου και ένα text σε μορφή string, και γράφει το text στο τέλος του συγκεκριμένου αρχείου. Δηλαδή, εάν το αρχείο περιέχει προηγούμενες εγγραφές, το text προστίθεται μετά από αυτές.

4.2.12 Κλάση StreamGobbler

Η κλάση αυτή αφορά την εμφάνιση μηνυμάτων κατά την εκτέλεση εξωτερικών προγραμμάτων κι έχει μόνο μια μέθοδο, τη run.

Μέθοδος run: Είναι υπεύθυνη για την εκτύπωση μηνυμάτων που αφορούν την πρόοδο εκτέλεσης των εξωτερικών προγραμμάτων και το είδος του λάθους σε περίπτωση που προκύψει κάποιο. Η κλάση αυτή είναι απαραίτητη κατά την κλήση των εκτελέσιμων προγραμμάτων του CLUTO στην διαδικασία της συσταδοποίησης.

4.2.13 Κλάση Clustering

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία των clusters χρησιμοποιώντας το CLUTO το οποίο είναι εργαλείο υλοποίησης clustering. Η ενσωμάτωση του CLUTO στην εφαρμογή και η αναλυτική περιγραφή του θα γίνουν στην ενότητα 5.1.2. Στην παρούσα ενότητα, θα αναφερθούμε πολύ συνοπτικά στις μεθόδους createVSMatrix, vclusterExec και ClusterHandler της κλάσης.

i) ***Μέθοδος createVSMatrix:*** Η μέθοδος αυτή είναι υπεύθυνη για τη δημιουργία κατάλληλου πίνακα διανυσμάτων σύμφωνα με το μοντέλο Vector Space Model το οποίο έχει περιγραφεί σε προηγούμενη ενότητα.

ii) ***Μέθοδος vclusterExec:*** Η μέθοδος αυτή είναι υπεύθυνη για την εφαρμογή κατάλληλου αλγορίθμου σύγκρισης στα διανύσματα που δημιουργήθηκαν με το μοντέλο Vector Space Model, έτσι ώστε να παραχθούν τα επιθυμητά clusters.

iii) Μέθοδος ClusterHandler: Η μέθοδος αυτή χειρίζεται κατάλληλα τις πληροφορίες που δίνει τελικά το CLUTO για τα clusters και φροντίζει για το "γέμισμα" του αντίστοιχου ComboBox με τις τιμές αυτών των clusters.

4.3 Βάση Δεδομένων

Στην συγκεκριμένη ενότητα ερμηνεύουμε το E-R διάγραμμα της βάσης δεδομένων, το οποίο δόθηκε στο προηγούμενο κεφάλαιο. Επειδή η παρούσα διπλωματική εργασία επικεντρώνεται στην διαχείριση papers, αξιοποιήσαμε την βάση επιστημονικών δημοσιεύσεων του DBLP.

Τα papers, τα οποία αποτελούν τις οντότητες της παρούσας βάσης, υπάγονται σε ακριβώς μία από τις ακόλουθες κατηγορίες (ολική ειδίκευση):

i. proceedings
ii. article
iii. book
iv. www
v. phdthesis
vi. masterthesis
vii. inproceedings
viii. incollection

Οι ιδιότητες αυτών των επιστημονικών δημοσιεύσεων διακρίνονται στις παρακάτω:

- title (προσδιοριστική ιδιότητα – για κάθε είδους paper)
- type (μερικώς προσδιοριστική ιδιότητα – για κάθε είδους paper)
- author / editor (ιδιότητα πολλαπλών τιμών – για papers τύπου proceedings, article, book, www, phdthesis, masterthesis, inproceedings, incollection)
- date_added (ιδιότητα που παράγεται αυτοματοποιημένα – για κάθε είδους paper)
- year (απλή ιδιότητα – για papers τύπου proceedings, article, book, www, phdthesis, masterthesis, inproceedings, incollection)
- publisher (απλή ιδιότητα – για papers τύπου proceedings, book)
- conference (απλή ιδιότητα – για papers τύπου proceedings, inproceedings, incollection)

- institution (απλή ιδιότητα – για papers τύπου phdthesis, masterthesis)
- series (απλή ιδιότητα – για papers τύπου proceedings)
- journal (απλή ιδιότητα – για papers τύπου article)
- ISBN (απλή ιδιότητα – για papers τύπου proceedings, book)
- volume (απλή ιδιότητα – για papers τύπου proceedings, article)
- page_region (απλή ιδιότητα – για papers τύπου article, inproceedings, incollection)
- number (απλή ιδιότητα – για papers τύπου article)
- web_site (απλή ιδιότητα – για papers τύπου article, inproceedings, incollection)
- DBLP_URL (απλή ιδιότητα – για papers τύπου proceedings, article, book, www, inproceedings, incollection)
- publisher_URL (απλή ιδιότητα – για papers τύπου book)
- series_URL (απλή ιδιότητα – για papers τύπου proceedings)

5

Ειδικά Ζητήματα Υλοποίησης

Το κεφάλαιο αυτό αφορά τις σημαντικότερες λεπτομέρειες της υλοποίησης και συγκεκριμένα τα θέματα εκείνα τα οποία χρειάστηκαν κάποιον ειδικό χειρισμό. Επίσης, παραθέτουμε οδηγίες σχετικές με την εγκατάσταση του συστήματος σε έναν υπολογιστή εξ' αρχής.

5.1 Λεπτομέρειες υλοποίησης

Στην παρούσα ενότητα γίνεται αναφορά στα θέματα της υλοποίησης τα οποία χρειάστηκαν ειδική αντιμετώπιση. Τα προβλήματα και οι λύσεις που προτιμήθηκαν θα παρουσιαστούν με τη σειρά που τα συναντάμε στην υλοποίηση, από τη στιγμή που παίρνουμε τα αποτελέσματα του web-search μέχρι να εμφανιστούν στο χρήστη όλα τα facets και οι ομάδες του clustering.

5.1.1 Ανάκτηση πληροφορίας από PDF

Ένα από τα προβλήματα που προέκυψαν ήταν η ανάκτηση πληροφορίας από ένα PDF αρχείο. Η επίτευξη του παραπάνω κρίθηκε αναγκαία προκειμένου να αυξηθούν οι δυνατότητες ταξινόμησης που παρέχονται στο χρήστη κι έτσι η εφαρμογή να καταστεί περισσότερο αποδοτική και χρήσιμη. Πιο συγκεκριμένα, επειδή η εφαρμογή επικεντρώνεται στο χειρισμό των αποτελεσμάτων της αναζήτησης που είναι papers, κρίθηκε χρήσιμο, να αντλήσουμε από κάθε paper πληροφορίες όπως τα κομμάτια "Abstract", "Categories & Subject Descriptors" "Keywords" και "General Terms", όπου αυτά υπάρχουν, καθένα από τα οποία έχει μια ιδιαίτερη σημασία.

Το “Abstract” αποτελεί μια περίληψη του εκάστοτε paper, και σε συνδυασμό με τον αντίστοιχο τίτλο, αποτελούν το κριτήριο για τη δημιουργία των clusters. Το να προσπαθούσαμε να δημιουργήσουμε τα clusters από ολόκληρο το κείμενο, θα ήταν αρκετά πιο χρονοβόρο και σπάταλο όσον αφορά τη μνήμη του συστήματος χωρίς ιδιαίτερη βελτίωση στην ποιότητα των αποτελεσμάτων.

Το κομμάτι “Categories & Subject Descriptors” κατατάσσει το κάθε paper σε μια ACM κατηγορία. Με τον τρόπο αυτό, έχουμε τη θεματική ενότητα κάθε κειμένου, το οποίο αποτελεί πολύ σημαντική πληροφορία αφού αποτελεί συγχρόνως την πιο σύντομη και έγκυρη περιγραφή του κειμένου. Πολλά papers μπορεί να έχουν τίτλους που δεν είναι αρκετά επεξηγηματικοί, ή ακόμα που να μην παρέχουν καμία πληροφορία σε κάποιον άπειρο προς το αντικείμενο χρήστη. Ας θεωρήσουμε το παράδειγμα ενός paper το οποίο έχει τίτλο “Clustering”. Ο τίτλος αυτός είναι πολύ γενικός και δε δίνει σχεδόν καμία πληροφορία για το περιεχόμενο του κειμένου. Το κείμενο θα μπορούσε να αναφέρεται σε αλγορίθμους υλοποίησης Clustering, σε εργαλεία με τα οποία μπορούμε να το υλοποιήσουμε, στις γενικές αρχές στις οποίες βασίζεται ο συγκεκριμένος τρόπος ταξινόμησης ή οτιδήποτε άλλο σχετικό. Ο χρήστης πιθανό να ενδιαφέρονταν για μία μόνο από αυτές τις θεματικές ενότητες, κι έτσι δε θα μπορούσε να καταλάβει αν το κείμενο ανήκει ή όχι στην περιοχή ενδιαφέροντός του χωρίς την αναφορά στην ACM κατηγορία του. Ακόμα, ας θεωρήσουμε το παράδειγμα ενός φοιτητή ο οποίος μόλις έχει αρχίσει να ασχολείται με την επιστήμη των υπολογιστών κι αναζητά πληροφορίες για ταξινομήσεις δεδομένων. Ο παραπάνω είναι πολύ πιθανόν να του είναι τελείως άγνωστος κι έτσι μόνο με αναφορά στην θεματική ενότητα θα μπορέσει να καταλάβει αν το κείμενο αυτό θα τον βοηθήσει ή όχι στην έρευνά του.

Τα “Keywords” είναι οι λέξεις κλειδιά κάθε paper και είναι πολύ πιθανό να αποτελούν κριτήριο κατά την αναζήτηση.

Τέλος, τα “General Terms”, αποτελούν επίσης πολύ σημαντικά κριτήρια ταξινόμησης και είναι ιδιαίτερα χρήσιμα σε όσους αναζητούν πληροφορίες με βάση κάποιο γενικό αλλά συγκεκριμένο όρο.

Τα βήματα με τα οποία υλοποιήθηκε η ανάκτηση δεδομένων από ένα PDF, και τα προβλήματα που συναντήσαμε, περιγράφονται αναλυτικά στη συνέχεια μαζί με τις αντίστοιχες μεθόδους επίλυσης.

5.1.1.1 Αναζήτηση PDF στο Google

Αρχικά πρέπει να αναζητήσουμε το PDF στο Google με σκοπό να ανακτήσουμε το URL προς την HTML σελίδα του PDF από τα αποτελέσματα της αναζήτησης. Ως λέξεις κλειδιά, θα θεωρήσουμε τον τίτλο του εκάστοτε PDF και τα ονόματα από δύο το πολύ συγγραφείς. Με αυτά τα keywords, το PDF που ζητάμε, εάν υπάρχει, θα είναι το πρώτο επιστρεφόμενο

αποτέλεσμα. Το πρόβλημα που προκύπτει είναι το πώς θα κατασκευάσουμε το κατάλληλο URL για την αναζήτησή μας.

Πρωτού αναφερθούμε στα βήματα τα οποία πρέπει να ακολουθήσει κανείς για την κατασκευή ενός URL, είναι σκόπιμο να αναφερθούμε συνοπτικά στην ίδια τη διαδικασία της αναζήτησης η οποία ουσιαστικά αποτελεί μια αλληλεπίδραση του χρήστη με τον Web Server της αντίστοιχης μηχανής. Ο πιο συνηθισμένος τρόπος αλληλεπίδρασης ανάμεσα σε ένα χρήστη και έναν Web Server, είναι μέσω ενός προγράμματος CGI. Το CGI είναι συντομογραφία του Common Gateway Interface, κι ένα τέτοιο πρόγραμμα έχει σχεδιαστεί για να δέχεται και να στέλνει δεδομένα σε Web Servers. Παράδειγμα χρήσης προγραμμάτων CGI είναι οι σελίδες του διαδικτύου στις οποίες περιέχονται φόρμες. Η επεξεργασία των στοιχείων που υποβάλλονται από χρήστες στα πεδία της φόρμας γίνεται με τέτοια προγράμματα. Στην δική μας περίπτωση της αναζήτησης, το CGI πρόγραμμα του Google πρέπει να κληθεί μέσα από το URL προκειμένου να πραγματοποιηθεί με επιτυχία η αλληλεπίδραση.

Το να κατασκευάσουμε ένα URL αποτελείται γενικά από τα εξής βήματα:

1) Καθορισμός του CGI path δηλαδή του μονοπατιού μέσω του οποίου θα κληθεί το αντίστοιχο CGI πρόγραμμα του Google, και προσθήκη αυτού στο βασικό URL της αναζήτησης. Το πρόγραμμα αυτό ονομάζεται search, και το βασικό URL της αναζήτησης είναι το http://www.google.com/. Μετά από το βήμα αυτό, το URL έχει πάρει τη μορφή http://www.google.com/search.
2) Προσθέτουμε ένα "?" στο μέχρι τώρα URL. Το "?" διαχωρίζει το CGI path από τις παραμέτρους οι οποίες αφορούν την αναζήτηση.
3) Μετά το "?", ακολουθεί ο καθορισμός των παραμέτρων που προαναφέρθηκαν και αφορούν την αναζήτηση.

Κάθε παράμετρος είναι ένα ζευγάρι key – value, όπου στη θέση του key έχουμε το όνομα της παραμέτρου και στη θέση του value την αντίστοιχη τιμή. Μεταξύ μιας παραμέτρου και της αντίστοιχης τιμής παρεμβάλλεται το "=". Δύο διαδοχικές παράμετροι χωρίζονται μεταξύ τους με το "&".

Οι παράμετροι που καθορίζονται οπωσδήποτε σε κάθε αναζήτηση είναι οι:

- ✓ ***q=(value):*** Το q προέρχεται από το query και η αντίστοιχη τιμή της παραμέτρου είναι η εκάστοτε λέξη κλειδί. Αν η αναζήτηση γίνεται με περισσότερες από μία λέξεις κλειδιά, τότε αυτές διαχωρίζονται μεταξύ τους με ειδικούς χαρακτήρες όπως το "+", το οποίο είναι και το πιο συνηθισμένο. Αυτό συμβαίνει διότι το κενό διάστημα δεν είναι επιτρεπτός χαρακτήρας για ένα URL. Για παράδειγμα, έστω ότι αναζητούμε

πληροφορίες με λέξεις κλειδιά το faceted classification. Η παράμετρος θα πάρει την τιμή q=faceted+classification.

- ✓ ***btnG=Google+Search:*** Είναι μια παράμετρος την οποία το Google θεωρεί απαραίτητη να καθοριστεί προκειμένου να πραγματοποιηθεί αναζήτηση και για αυτό προστίθεται πάντα στο URL.
- ✓ ***hl=en:*** Το hl προέρχεται από το handle language και η τιμή της παραμέτρου υποδεικνύει τη γλώσσα στην οποία επιστρέφονται τα αποτελέσματα. Για να καθοριστεί η παράμετρος αυτή, ελέγχουμε τις λέξεις κλειδιά. Αν αυτές αποτελούνται από αγγλικούς χαρακτήρες, τότε τα αποτελέσματα που επιστρέφονται είναι στα αγγλικά, ενώ αν αποτελούνται από ελληνικούς χαρακτήρες, τότε παίρνουμε αποτελέσματα στα ελληνικά.

Αυτές οι τρεις παράμετροι είναι απαραίτητες για τη βασική αναζήτηση στο Google. Υπάρχουν πολλές ακόμα κατηγορίες παραμέτρων οι οποίες αφορούν πιο εξεζητημένες αναζητήσεις. Μια τέτοια προαιρετική παράμετρος είναι η ***as_filetype = (filetype extension)*** με την οποία δίνεται προτεραιότητα σε αποτελέσματα που έχουν το συγκεκριμένο τύπο αρχείων. Στην περίπτωση της δικής μας αναζήτησης, αυτό είναι απαραίτητο διότι ψάχνουμε συγκεκριμένα για PDF αρχεία.

Λαμβάνοντας υπόψη όλα τα παραπάνω, μπορέσαμε να κατασκευάσουμε το κατάλληλο URL για την αναζήτηση των PDFs. Ο κώδικας που επιτελεί τα παραπάνω παρατίθεται στη συνέχεια:

```
keyword = keyword.replace(" ", "+");
String lang = "en";
If ((keyword.codePointAt (0) >= 880) && (keyword.codePointAt (0) <= 974)) lang = "el";
URL url = new URL (http://www.google.com/search?num= + String.valueof(resnum) +
"&hl=" + lang + "&q=" + keyword + "&as_filetype=pdf&btnG=Google+Search")
```

Ο κώδικας αυτός:

i) Τροποποιεί κατάλληλα τις λέξεις κλειδιά

ii) Καθορίζει τη γλώσσα με έλεγχο στα keywords

iii) "Χτίζει" το URL λαμβάνοντας υπόψη όλες τις παραμέτρους που αναφέραμε.

5.1.1.2 Ανάκτηση Πληροφορίας από HTML σελίδα

Έχοντας δημιουργήσει το URL αναζήτησης PDF στο προηγούμενο βήμα, προχωράμε τώρα στην ανάκτηση της επιθυμητής πληροφορίας από τη σελίδα αποτελεσμάτων του Google. Η

πληροφορία που χρειαζόμαστε είναι το URL που οδηγεί στην HTML μορφή του PDF. Το πρόβλημα που προκύπτει είναι η απομόνωση του επιθυμητού κομματιού κειμένου, όχι από το υπόλοιπο source, αλλά από τα HTML tags ανάμεσα στα οποία αυτό βρίσκεται.

Για να το καταφέρουμε αυτό, χρησιμοποιούμε τον Jericho HTML Parser. Ο Jericho είναι ένας open source parser ο οποίος επιτρέπει την ανάλυση και το χειρισμό κομματιών HTML κειμένου με πολύ αποδοτικό τρόπο. Η επίσημη ιστοσελίδα στην οποία μπορεί να βρει κανείς αναλυτικές πληροφορίες για τον parser αυτόν είναι η http://jericho.htmlparser.net/docs/index.html. Προκειμένου να γίνει κατανοητός ο τρόπος λειτουργίας του Jericho και οι λεπτομέρειες της υλοποίησης μας, παραθέτουμε ακολούθως μερικά πολύ βασικά θεωρητικά στοιχεία.

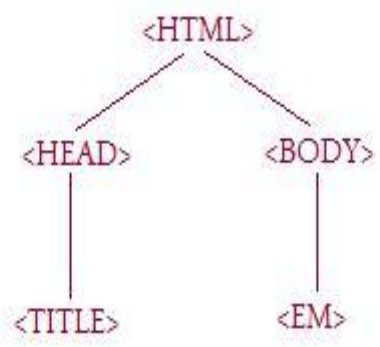
Η γλώσσα HTML (HyperText Markup Language) είναι μια από τις πιο διαδεδομένες *περιγραφικές γλώσσες (markup language)*. Αυτό σημαίνει ότι δεν είναι γλώσσα προγραμματισμού αλλά ένας ειδικός τρόπος γραφής κειμένου. Οι browsers αναγνωρίζουν αυτόν τον τρόπο γραφής και εκτελούν τις "οδηγίες" ή "εντολές" που βρίσκονται σε αυτόν. Η HTML χρησιμοποιεί ειδικές ετικέτες (tags) για να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι οι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας και βρίσκονται μεταξύ < και >.

Λόγω της δομής της HTML, οποιοδήποτε HTML document αποτελείται ουσιαστικά από ξεχωριστά στοιχεία (elements) τα οποία μπορούν να αναπαρασταθούν σε μια δενδρική δομή. Κάποιο στοιχείο αποτελεί τη ρίζα του δένδρου ενώ κάθε στοιχείο μπορεί να περιέχει είτε άλλα στοιχεία, είτε απλώς και μόνο κείμενο. Στο επόμενα σχήμα βλέπουμε ένα παράδειγμα HTML document:

```
<html lang="en-US">
  <head>
    <title>
      Blank Document!
    </title>
  </head>
  <body bgcolor="#d010ff">
    I've got
    <em>
      something to saaaaay
    </em>
    !
  </body>
</html>
```

Σχήμα 5.1 Παράδειγμα HTML document

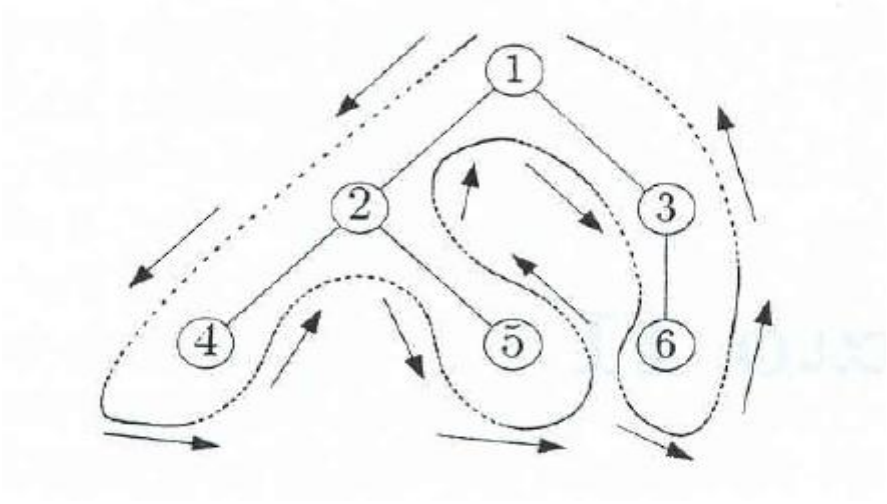
Στο παράδειγμα αυτό τα elements που αποτελούν το συγκεκριμένο document είναι τα: <html>, <head>, <title>, <body> και . Παρατηρούμε ότι κάποια elements περιέχουν άλλα elements, ενώ κάποια άλλα περιέχουν απλώς κείμενο. Το αντίστοιχο δένδρο για την παραπάνω HTML δομή είναι αυτό που φαίνεται στο *σχήμα 5.2*.



Σχήμα 5.2 Παράδειγμα Δενδρικής Δομής HTML document

Το γεγονός αυτό, δηλαδή την αναπαράσταση ενός HTML document σε μια δενδρική δομή, εκμεταλλεύεται ο Jericho. Για κάθε HTML σελίδα δημιουργεί ένα εικονικό δένδρο, η διάσχιση του οποίου γίνεται με τον γνωστό από τη θεωρία των γράφων "βάθος-πρώτα" αλγόριθμο (DFS: Depth First Search algorithm). Η διάσχιση αυτή ονομάζεται επίσης και προδιατεταγμένη (pre-order).

Για να καταλάβουμε τον τρόπο με τον οποίο πραγματοποιείται η διάσχιση αυτή, ας θεωρήσουμε μια διαδρομή γύρω από ένα δένδρο η οποία ξεκινάει από τη ρίζα και προχωράει με φορά αντίθετη από αυτή των δεικτών του ρολογιού, μένοντας όσο το δυνατόν πλησιέστερα στο δένδρο, όπως φαίνεται στο *σχήμα 5.3*.



Σχήμα 5.3 Προδιατεταγμένη Διάσχιση Γράφου

Η προδιατεταγμένη (ή κατά βάθος) διάσχιση παράγεται όταν κατά τη διάρκεια της διαδρομής καταγράψουμε κάθε κόμβο την πρώτη φορά που τον συναντάμε. Η εξήγηση αυτή είναι περισσότερο πρακτική κι όχι τόσο επιστημονική, ωστόσο επαρκεί για το σκοπό που τη χρειαζόμαστε στο εν λόγω κείμενο.

Κατανοώντας λοιπόν τα παραπάνω, και μελετώντας το source από μια σελίδα αποτελεσμάτων του Google, καταλήξαμε σε κάποια κριτήρια τα οποία θέσαμε καθώς διασχίζαμε με τον Jericho τα elements της δενδρικής δομής, κι έτσι καταφέραμε να απομονώσουμε το επιθυμητό URL μέσω του οποίου θα ανακτήσουμε το source του PDF.

5.1.1.3 Ανάκτηση πληροφορίας από HTML source με χρήση Regular Expressions

Το τελευταίο βήμα της ειδικής επεξεργασίας ενός PDF είναι η ανάκτηση της επιθυμητής πληροφορίας μέσα από το source του PDF. Αυτό γίνεται με χρήση ***κανονικών εκφράσεων (Regular Expressions, συντομότερα RegExes)***. Μια κανονική έκφραση είναι μια ειδική συμβολοακολουθία η οποία αποτελεί ένα μοτίβο που εξυπηρετεί στην αναζήτηση εντός κειμένου και στην αναγνώριση ορισμένων τμημάτων κειμένου.

Κατά την υλοποίηση, χρησιμοποιήσαμε τις κανονικές εκφράσεις για να ταυτοποιήσουμε κάποια συγκεκριμένα σημεία των κειμένων των PDFs ανάμεσα στα οποία βρίσκονται οι πληροφορίες που μας ενδιαφέρουν. Πιο συγκεκριμένα, ταυτοποιήσαμε τις κεφαλίδες “ABSTRACT” και “INTRODUCTION” κάθε κειμένου ανάμεσα στις οποίες βρίσκεται το κείμενο “Abstract”, τα “Categories and Subject Descriptors”, τα “Keywords” και τα “General Terms”.

5.1.2 Υλοποίηση Clustering με χρήση του CLUTO

Ένα δεύτερο πρόβλημα που συναντήσαμε, είναι ο τρόπος υλοποίησης του clustering καθώς υπάρχουν πολλοί διαφορετικοί αλγόριθμοι, βιβλιοθήκες και εργαλεία που εξυπηρετούν το συγκεκριμένο σκοπό. Καταλληλότερη επιλογή θεωρήσαμε το clustering να υλοποιηθεί με χρήση του εργαλείου CLUTO. Το CLUTO είναι ένα δημοφιλές freeware εργαλείο το οποίο χρησιμοποιείται τόσο για την υλοποίηση clustering με αποδοτικό τρόπο σε μικρές και μεγάλες ομάδες δεδομένων, όσο και για την ανάλυση των χαρακτηριστικών των clusters που προκύπτουν. Η αντίστοιχη ιστοσελίδα στην οποία μπορεί κανείς να βρει πληροφορίες για το εργαλείο αυτό, αλλά και να το κατεβάσει είναι η: http://glaros.dtc.umn.edu/gkhome/views/cluto .

5.1.2.1 Το πρόγραμμα vcluster γενικά

Το CLUTO αποτελείται από δύο αυτόνομα προγράμματα το vcluster και το scluster και από μια αυτόνομη βιβλιοθήκη. Η υλοποίηση του clustering μπορεί να γίνει είτε χρησιμοποιώντας κάποιο από τα προαναφερθέντα προγράμματα με κατάλληλη επιλογή παραμέτρων, είτε με απευθείας υλοποίηση διαφόρων αλγορίθμων clustering χρησιμοποιώντας τις συναρτήσεις της βιβλιοθήκης.

Στη συγκεκριμένη εφαρμογή, το clustering υλοποιήθηκε με το πρόγραμμα vcluster. Η κύρια διαφορά μεταξύ των δύο προγραμμάτων είναι ότι στο vcluster η είσοδος είναι μια πολυδιάστατη αναπαράσταση των αντικειμένων (το *v* προέρχεται από το *vector* δηλαδή διάνυσμα) ενώ η είσοδος του scluster είναι ένας πίνακας (ή γράφος) ομοιότητας (το *s* προέρχεται από το *similarity* δηλαδή ομοιότητα). Πέραν αυτής της διαφοράς τα δύο προγράμματα λειτουργούν με παρόμοιο τρόπο. Ο λόγος που το vcluster κρίθηκε καταλληλότερο, είναι η μορφή των δεδομένων εισόδου. Στη συνέχεια θα ασχοληθούμε μόνο με αυτό.

Η κλήση του vcluster δημιουργεί έναν προκαθορισμένο αριθμό clusters από μια συλλογή δεδομένων εισόδου, έστω ***k***, ενώ κάθε αντικείμενο εισόδου αποτελεί ένα πολυδιάστατο διάνυσμα. Η κλήση του vcluster γίνεται με την εντολή:

vcluster [optional parameters] *MatrixFile NClusters*

Το *NClusters* είναι ο προκαθορισμένος αριθμός από clusters ο οποίος καθορίζεται κατά την κλήση. Το *MatrixFile* είναι ένα αρχείο στο οποίο είναι αποθηκευμένα τα ***n*** αντικείμενα από τα οποία θα δημιουργηθούν τα clusters. Κάθε αντικείμενο θεωρείται ένας *n x m* πίνακας, όπου κάθε γραμμή αντιστοιχεί σε ένα αντικείμενο και κάθε στήλη σε μια διάσταση του κάθε αντικειμένου. Η ακριβής μορφή του εν λόγω πίνακα θα περιγραφεί αναλυτικά στην ενότητα 5.1.2.3.

Όταν το vcluster εκτελεστεί με επιτυχία, στα αποτελέσματα εμφανίζονται και κάποια στατιστικά στοιχεία τα οποία αφορούν την ποιότητα των δημιουργημένων clusters και το χρόνο που χρειάστηκε για τη δημιουργία τους. Τα στατιστικά αυτά αποθηκεύονται μαζί με τη λύση σε ένα αρχείο το οποίο έχει όνομα *MatrixFile.clustering.NClusters*. Η ακριβής μορφή του αρχείου αποτελεσμάτων περιγράφεται με λεπτομέρειες στην ενότητα 5.1.2.4.

Η συμπεριφορά του vcluster καθορίζεται και από μερικές διαφορετικές προαιρετικές παραμέτρους ([optional parameters] κατά την κλήση του προγράμματος). Οι μεταβλητές αυτές διαχωρίζονται σε τρεις γενικές ομάδες. Η πρώτη ομάδα αφορά λεπτομέρειες της υλοποίησης του clustering όπως ο αλγόριθμος ο οποίος χρησιμοποιείται, η δεύτερη ομάδα αφορά τον τύπο της ανάλυσης στα δημιουργημένα clusters και η τρίτη ομάδα αφορά την οπτικοποίηση των αποτελεσμάτων δηλαδή, το αν τα τελικά αποτελέσματα θα τα δούμε μόνο

από το αρχείο αποτελεσμάτων, ή επιπλέον θα δημιουργηθούν απεικονίσεις με διαφορετικούς τρόπους στις οποίες θα φαίνονται τόσο τα τελικά clusters που δημιουργούνται όσο και η ποιότητα καθενός από αυτά.

Στο σχήμα 5.4 βλέπουμε ένα παράδειγμα στο οποίο χρησιμοποιήθηκε το vcluster για να ταξινομήσει έναν πίνακα δεδομένων σε δέκα clusters. Από το σχήμα, βλέπουμε ότι το vcluster αρχικά τυπώνει πληροφορίες σχετικές με τον πίνακα, όπως το όνομά του, τον αριθμός γραμμών (*#Rows*), τον αριθμό στηλών (*#Columns*) και τον αριθμό μη μηδενικών στοιχείων (*#NonZeros*). Στη συνέχεια, τυπώνει πληροφορίες για τις τιμές διαφόρων παραμέτρων οι οποίες καθορίστηκαν κατά την υλοποίηση του clustering και τον αριθμό των clusters που δημιουργήθηκαν. Οι παράμετροι αυτές είναι οι Optional Parameters που αναφέρθηκαν προηγουμένως και οι οποίες δεν είναι απαραίτητο ότι καθορίστηκαν κατά την κλήση του συστήματος. Για κάθε παράμετρο, υπάρχει μια προκαθορισμένη (default) τιμή στο σύστημα, η οποία χρησιμοποιείται σε περίπτωση που δεν καθοριστεί κάποια διαφορετική τιμή για την παράμετρο αυτή από τη γραμμή εντολών.

Πιο κάτω, στην απεικόνιση της λύσης, εμφανίζονται πληροφορίες σχετικές με την ποιότητα των συνολικών αποτελεσμάτων αλλά και την ποιότητα κάθε cluster ξεχωριστά. Η σημασία των στατιστικών αυτών στοιχείων περιγράφεται πιο αναλυτικά στην ενότητα 5.1.2.2.

Στο τέλος του πίνακα, αναγράφεται ο χρόνος που χρειάστηκαν για να ολοκληρωθούν οι διάφορες φάσεις του προγράμματος. Για το συγκεκριμένο παράδειγμα, χρειάστηκαν 0.950 δευτερόλεπτα για να διαβαστεί το αρχείο εισόδου και να γραφτεί η λύση, 9.060 δευτερόλεπτα για την ολοκλήρωση του υπολογισμού όλων των clusters και 0.240 δευτερόλεπτα για τον υπολογισμό των στατιστικών που αφορούν την ποιότητα του clustering.

```
prompt% vcluster sports.mat 10
*******************************************************************************
vcluster (CLUTO 2.1) Copyright 2001-02, Regents of the University of Minnesota

Matrix Information ------------------------------------------------------------
  Name: sports.mat, #Rows: 8580, #Columns: 126373, #NonZeros: 1107980

Options -----------------------------------------------------------------------
  CLMethod=RB, CRfun=I2, SimFun=Cosine, #Clusters: 10
  RowModel=None, ColModel=IDF, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=10, NIter=10

Solution ----------------------------------------------------------------------

------------------------------------------------------------------------
10-way clustering: [I2=2.29e+03] [8580 of 8580]
------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  |
------------------------------------------------------------------------
  0   359 +0.168 +0.050 +0.020 +0.005 |
  1   629 +0.106 +0.041 +0.022 +0.007 |
  2   795 +0.102 +0.036 +0.018 +0.006 |
  3   762 +0.099 +0.034 +0.021 +0.006 |
  4   482 +0.098 +0.045 +0.022 +0.009 |
  5   844 +0.095 +0.035 +0.023 +0.007 |
  6  1724 +0.059 +0.026 +0.022 +0.007 |
  7  1175 +0.051 +0.015 +0.021 +0.006 |
  8   853 +0.043 +0.015 +0.019 +0.006 |
  9   957 +0.032 +0.012 +0.015 +0.006 |
------------------------------------------------------------------------

Timing Information ------------------------------------------------------------
   I/O:                                  0.950 sec
   Clustering:                           9.060 sec
   Reporting:                            0.240 sec
*******************************************************************************
```

Σχήμα 5.4 Παράδειγμα υλοποίησης Clustering

5.1.2.2 Στατιστικές Πληροφορίες για τα αποτελέσματα

Το vcluster έχει τη δυνατότητα να υπολογίσει πολλά διαφορετικά στατιστικά στοιχεία τα οποία αφορούν τα τελικά αποτελέσματα, στα οποία όμως δε θα επεκταθούμε για λόγους έκτασης. Το πιο απλά από αυτά, είναι τα ποιοτικά στοιχεία που έχουν ήδη περιγραφεί και αφορούν τόσο την ποιότητα του κάθε cluster ξεχωριστά, όσο και την ποιότητα του συνολικού αποτελέσματος. Η ποιότητα του κάθε cluster εξαρτάται από το πόσο "μοιάζουν" τα αντικείμενα τα οποία το αποτελούν, ενώ η ποιότητα της ολικής ταξινόμησης εξαρτάται προφανώς από την ποιότητα των clusters μεμονωμένα. Ο υπολογισμός της ποιότητας της συνολικής λύσης, γίνεται με την εφαρμογή μιας συνάρτησης στα αποτελέσματα. Η συνάρτηση ονομάζεται "*Criterion Function*" και καθορίζεται από τις προαιρετικές παραμέτρους.

Στο παραπάνω παράδειγμα, τα σχετικά με την ποιότητα στατιστικά, εμφανίζονται μαζί με τη λύση στο κομμάτι "*Solution*" του αρχείου. Αρχικά, βλέπουμε μια αναφορά στη συνάρτηση – κριτήριο *"I2 = 2.29e + 03"*, όπου I2 είναι το όνομα μιας από τις συναρτήσεις που είναι διαθέσιμες σύμφωνα με το manual του CLUTO.

Δίπλα ακριβώς, αναφέρεται ο αριθμός των δεδομένων που κατέστη δυνατόν να ταξινομηθούν σε clusters σε σχέση με τα συνολικά. Στο παράδειγμά μας, όλα τα αντικείμενα ταξινομήθηκαν. Κατά γενικό κανόνα, το vcluster, προσπαθεί να ταξινομήσει όλα τα δεδομένα. Ωστόσο, μερικές φορές, κάποια από τα δεδομένα, είναι πιθανόν να μην μπορούν να αντιστοιχηθούν σε κάποιο από τα clusters που προκύπτουν. Οι λόγοι που συμβαίνει αυτό, επεξηγούνται στην ενότητα 5.1.2.4.

Κάτω από την αναφορά στη συνολική ποιότητα του συστήματος, έχουμε αναγραφή της λύσης μαζί με στατιστικά στοιχεία που αφορούν το κάθε cluster ξεχωριστά. Η πρώτη στήλη με τίτλο "cid" (cluster id) αντιστοιχεί στον αριθμό του cluster στο οποίο γίνεται αναφορά στην υπόλοιπη γραμμή. Η δεύτερη στήλη, αναφέρεται στο μέγεθος "Size" του κάθε cluster, δηλαδή στον αριθμό αντικειμένων από την αρχική συλλογή που το αποτελούν. Η στήλη "ISim" (internal similarities), αναφέρεται στη μέση ομοιότητα μεταξύ των αντικειμένων του κάθε cluster. Η στήλη "ISdev" (internal standard deviation), αναφέρεται στην τυπική απόκλιση των μέσων αυτών ομοιοτήτων. Η στήλη "ESim" (external similarities), αναφέρεται στη μέση ομοιότητα των αντικειμένων κάθε cluster με τα υπόλοιπα αντικείμενα και τέλος, η στήλη "ESdev" (external standard deviation), αναφέρεται στην τυπική απόκλιση των μέσων εξωτερικών ομοιοτήτων.

5.1.2.3 Πίνακας Διανυσμάτων

Το vcluster, όπως έχει ήδη αναφερθεί, δέχεται σαν είσοδο έναν πίνακα διανυσμάτων από τον οποίο στη συνέχεια, με κατάλληλο αλγόριθμο, θα υπολογιστούν τα clusters. Κάθε γραμμή του πίνακα αναπαριστά ένα μεμονωμένο αντικείμενο, ενώ οι στήλες του αντιστοιχούν στις διαστάσεις του διανύσματος. Ο πίνακας αποθηκεύεται σε ένα αρχείο, το οποίο αποτελεί βασική παράμετρο κατά την κλήση του προγράμματος.

Ο πίνακας διανυσμάτων, έστω *A*, έχει *n* γραμμές και *m* στήλες και αποθηκεύεται σε αρχείο κειμένου το οποίο περιέχει *n+1* γραμμές. Η πρώτη γραμμή περιέχει πληροφορίες σχετικές με το μέγεθος του πίνακα, ενώ οι υπόλοιπες *n* γραμμές περιέχουν πληροφορίες για κάθε αντικείμενο της συλλογής.

Η πρώτη γραμμή του πίνακα περιέχει ακριβώς τρεις ακέραιους αριθμούς. Ο πρώτος ακέραιος αντιστοιχεί στις γραμμές του πίνακα (n), ο δεύτερος στις στήλες (m), και ο τρίτος στο σύνολο των μη μηδενικών εισόδων σε αυτόν. Σημειώνουμε ότι, κατά την κατασκευή πίνακα διανυσμάτων για αρχεία κειμένου, όπως στην εφαρμογή, μόνο μη μηδενικές τιμές καταγράφονται στον πίνακα, οπότε ουσιαστικά ο τρίτος αριθμός αποτελεί το σύνολο των καταγεγραμμένων τιμών του αρχείου.

Οι υπόλοιπες n γραμμές αποθηκεύουν πληροφορίες σχετικές με τα αντικείμενα προς ταξινόμηση. Πιο συγκεκριμένα, η (i+1) γραμμή του αρχείου περιέχει πληροφορίες για το

αντικείμενο i της συλλογής. Οι μη μηδενικές είσοδοι κάθε γραμμής καθορίζονται ως ζευγάρια αριθμών. Κάθε ζευγάρι αποτελείται από τον αριθμό της στήλης και από την τιμή που έχει το συγκεκριμένο αντικείμενο στη στήλη αυτή. Οι τιμές των διαφόρων μεγεθών των αντικειμένων είναι δεκαδικοί αριθμοί και το τι αντιπροσωπεύουν κάθε φορά, εξαρτάται από τον τύπο των δεδομένων και του προβλήματος.

Σημειώνουμε ότι, οι στήλες αρχίζουν από τον αριθμό 1 κι όχι από το 0 όπως συνηθίζεται σε πολλές γλώσσες προγραμματισμού, συμπεριλαμβανομένης και της Java.

Στο σχήμα 5.5, βλέπουμε ένα παράδειγμα ενός πίνακα διανυσμάτων 7x8 και την αντίστοιχη απεικόνισή του στο CLUTO.

	1.1			-0.5			0.2
1.4	0.4		-0.4				
		1.8			2.0		3.0
1.0							
	5.5		3.0			8.0	
		1.0		-1.0	2.0		
	3.5		-1.0	4.0		2.0	8.0

Matrix Input File

```
7  8  21
2  1.1  5  -0.5  8  0.2
1  1.4  2  0.4  4  -0.4
3  1.8  6  2.0  8  3.0
1  1.0
2  5.5  4  3.0  7  8.0
3  1.0  5  -1.0  6  2.0
2  3.5  4  -1.0  5  4.0  7  2.0  8  8.0
```

Σχήμα 5.5 Παράδειγμα δημιουργίας Πίνακα Διανυσμάτων

5.1.2.4 Αρχείο Αποτελεσμάτων

Το αρχείο αποτελεσμάτων ενός πίνακα διανυσμάτων με n+1 γραμμές, αποτελείται από n γραμμές και κάθε γραμμή περιέχει έναν ακέραιο αριθμό. Η i-στή γραμμή του αρχείου αποτελεσμάτων αντιστοιχεί στο i-στο αντικείμενο της συλλογής. Οι αριθμοί που αντιστοιχούν στα clusters κυμαίνονται από 0 έως τον αριθμό των clusters μείον ένα. Εάν το CLUTO δεν καταφέρει να αναθέσει κάποιο αντικείμενο σε κάποιο cluster, τότε αναγράφεται ο αριθμός -1 στη γραμμή που αντιστοιχεί στο αντικείμενο αυτό στον πίνακα αποτελεσμάτων. Αυτό μπορεί να συμβεί διότι, το vcluster απομακρύνει όλες τις στήλες οι οποίες προκύπτουν σε λιγότερες από τρεις γραμμές πρωτού υπολογίσει τη λύση. Αυτό συμβαίνει για λόγους απόδοσης του συστήματος και δεν επηρεάζει την ποιότητα των τελικών clusters. Ωστόσο, σαν αποτέλεσμα του χειρισμού αυτού, κάποια αντικείμενα μπορεί να χάσουν τελείως όλες τους τις ιδιότητες και ως αποτέλεσμα αυτού να μην είναι δυνατή η κατάταξή τους σε κάποιο cluster.

5.1.2.5 Υλοποίηση

Αξιοποιώντας όλα τα παραπάνω, χρησιμοποιήσαμε το CLUTO για την υλοποίηση του Clustering στο σύστημά μας.

Αρχικά, αποθηκεύουμε το "Abstract" από κάθε PDF, μαζί με τον αντίστοιχο τίτλο, τα "Categories & Subject Descriptors", τα "General Terms" και τα "Keywords" σε ένα αρχείο κειμένου. Το αρχείο αυτό, έχει τόσες σειρές, όσα τα αντικείμενα προς συσταδοποίηση. Σε κάθε σειρά, γράφουμε τον αύξοντα αριθμό του αντικειμένου (θα μπορούσαμε να χρησιμοποιήσουμε οποιοδήποτε άλλο προσδιοριστικό) και στη συνέχεια το προαναφερθέν κείμενο.

Από το αρχείο αυτό, ύστερα από επεξεργασία με το πρόγραμμα doc2mat.pl, δημιουργείται το clutoInput.mat το οποίο είναι το αρχείο διανυσμάτων που περιγράφτηκε σε προηγούμενη ενότητα. Η εντολή με την οποία καλούμε το doc2mat.pl μέσα από τη Java, είναι η ακόλουθη:

perl doc2mat.pl –nlskip=1 doc2matInput.txt clutoInput.mat

Στη συνέχεια, το αρχείο clutoInput.mat, αποτελεί είσοδο για το vcluster. Η εντολή με την οποία καλούμε το vcluster, είναι η ακόλουθη:

vcluster clutoInput.mat k

Το k καθορίζεται από το χρήστη και είναι ο αριθμός των clusters που θα δημιουργηθούν. Το αρχείο αποτελεσμάτων δημιουργείται αυτόματα από το vcluster κι έχει όνομα *clutoInput.mat.clustering.k.* Από το αρχείο αυτό, εξάγουμε το μέγεθος κάθε συστάδας και τα papers ανήκουν σε καθεμία από αυτές. Με τα δεδομένα αυτά, γεμίζουμε το αντίστοιχο ComboBox στο interface.

5.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Στην ενότητα αυτή αναφέρουμε την πλατφόρμα ανάπτυξης και εκτέλεσης του συστήματος, καθώς και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν. Επιπλέον, περιγράφεται βήμα προς βήμα ο τρόπος εγκατάστασης της εφαρμογής σε έναν υπολογιστή.

5.2.1 Τεχνικά Θέματα της Υλοποίησης της Εφαρμογής

Θα ξεκινήσουμε με μια αναφορά στη γλώσσα προγραμματισμού που χρησιμοποιήσαμε και είναι η Java. Η Java είναι μια από τις πιο δημοφιλείς γλώσσες αντικειμενοστρεφούς προγραμματισμού (object – oriented programming). Ο επίσημος δικτυακός τόπος της γλώσσας αυτής είναι ο http://java.sun.com/ .

Η πλατφόρμα ανάπτυξης και εκτέλεσης που χρησιμοποιήσαμε είναι το Eclipse SDK (Software Development Kit) version 3.2.2. Ο επίσημος δικτυακός τόπος της προγραμματιστικής αυτής πλατφόρμας είναι ο http://www.eclipse.org/ .

Προκειμένου να χρησιμοποιήσουμε στο μέγιστο βαθμό τις δυνατότητες που παρέχει το Eclipse, εγκαταστήσαμε επιπλέον σε αυτό, τον Visual Editor version 1.2.1. Ο Visual Editor είναι ένα plug – in του Eclipse το οποίο προσφέρει τα κατάλληλα εργαλεία για την εύκολη κατασκευή του interface μιας εφαρμογής. Αυτό συμβαίνει τόσο διότι διευκολύνει το σχεδιασμό βασικών στοιχείων του interface όπως frames, windows, κλπ, όσο και γιατί παρέχει έτοιμο κώδικα για γραφικά όπως buttons, tooltips, radio buttons, check boxes, combo boxes κλπ.

Εκτός από το Eclipse, αξιοποιήθηκαν και άλλα δύο προγραμματιστικά εργαλεία με τα οποία αντιμετωπίσαμε τα πιο απαιτητικά κομμάτια της υλοποίησης. Τα εργαλεία αυτά είναι τα παρακάτω:

- ***Jericho HTML Parser version 2.6:*** Ο Jericho συνιστά ένα εργαλείο με το οποίο πραγματοποιούμε HTML parsing με αποδοτικό τρόπο όπου αυτό είναι απαραίτητο. Ο επίσημος δικτυακός τόπος του προγραμματιστικού αυτού εργαλείου είναι ο παρακάτω: http://jericho.htmlparser.net/docs/index.html .
- ***CLUTO version 2.1.2:*** Το CLUTO είναι ένα freeware εργαλείο υλοποίησης clustering και ανάλυσης των χαρακτηριστικών και της ποιότητας των clusters που προκύπτουν. Ο επίσημος δικτυακός τόπος του προγραμματιστικού αυτού εργαλείου είναι ο εξής: http://glaros.dtc.umn.edu/gkhome/views/cluto .

Επίσης, ήταν απαραίτητη η εγκατάσταση ενός interpreter για τη γλώσσα ***PERL***. Η PERL είναι απαραίτητη προκειμένου να λειτουργήσει το CLUTO, αφού μέσω ενός PERL script πραγματοποιείται η κωδικοποίηση αρχείων κειμένου διαδικασία απαραίτητη για την υλοποίηση του clustering. Επιλέξαμε την τελευταία έκδοση της γλώσσας η οποία είναι η ***version 5.10.0***. Ο επίσημος δικτυακός τόπος της γλώσσας αυτής είναι ο http://www.perl.org/ .

Ο υπολογιστής στον οποίο πραγματοποιήθηκε η ανάπτυξη της εφαρμογής είναι ο VAIO VGN – CS11 S/Q. Ο φορητός αυτός υπολογιστής διαθέτει τα εξής τεχνικά χαρακτηριστικά:

- ✓ ***Επεξεργαστής:*** Intel Core2 Duo P8400 @ 2.26Ghz 2.27Ghz
- ✓ ***Μνήμη RAM:*** 4096MB
- ✓ ***Τύπος Συστήματος:*** 32-bit

5.2.2 Διαδικασία Εγκατάστασης:

Εδώ θα περιγραφούν συνοπτικά όλες οι διαδικασίες και ενέργειες οι οποίες πρέπει να πραγματοποιηθούν προκειμένου να εγκαταστήσουμε εξ' αρχής σε έναν υπολογιστή την παρούσα εφαρμογή.

1) Εγκατάσταση της τελευταίας κάθε φορά έκδοσης Java, δηλαδή ενός JRE (Java Runtime Environment). Η τοποθεσία στην οποία μπορούμε να βρούμε το JRE είναι η: *http://java.sun.com/javase/downloads/index.jsp* .
2) Εγκατάσταση των αρχείων του Freemind. Αυτό γίνεται αναζητώντας το αρχείο WebFreeMind.rar στο δικτυακό τόπο *http://www.dbnet.ece.ntua.gr/dowloads/WebFreeMind.htm* και δημιουργώντας ένα καινούριο project στο Eclipse το οποίο θα περιέχει όλον τον κώδικα και τα επιπλέον αρχεία που βρίσκονται συμπιεσμένα στο WebFreeMind.rar.
3) Εγκαθιστούμε την τελευταία έκδοση ενός interpreter για τη γλώσσα PERL στον υπολογιστή μας. Μπορούμε να τη βρούμε στον ιστοτόπο: *http://www.perl.org/* όπως έχει αναφερθεί και παραπάνω.
4) Επιπλέον αναζητούμε το αρχείο cluto-2.1.2a.zip από το δικτυακό τόπο *http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download* . Στη συνέχεια, δημιουργούμε σε κάποιο σημείο του υπολογιστή μας (π.χ. στο "C:\") ένα φάκελο με το όνομα CLUTO και εκεί βάζουμε τα εκτελέσιμα doc2mat.pl και vcluster.exe τα οποία βρίσκουμε στο παραπάνω αρχείο. Στο φάκελο αυτό, θα πρέπει να αποθηκεύεται στη συνέχεια το αρχείο με τα αντικείμενα στα οποία πρέπει να εφαρμοστεί το clustering, και επίσης στο φάκελο αυτό θα δημιουργείται και το αρχείο με τα clusters, δηλαδή το αρχείο αποτελεσμάτων της διαδικασίας.

6

Έλεγχος

Το κεφάλαιο αυτό αποτελεί αξιολόγηση του συστήματος και επίσης λειτουργεί ως εγχειρίδιο χρήσης της εφαρμογής.

6.1 Μεθοδολογία ελέγχου

Η πραγματοποίηση του ελέγχου έγινε με τη βοήθεια ενός σεναρίου χρήσης, το οποίο και φροντίσαμε να χρησιμοποιεί τις περισσότερες από τις λειτουργίες του συστήματος. Το σενάριο αυτό περιγράφεται στη συνέχεια.

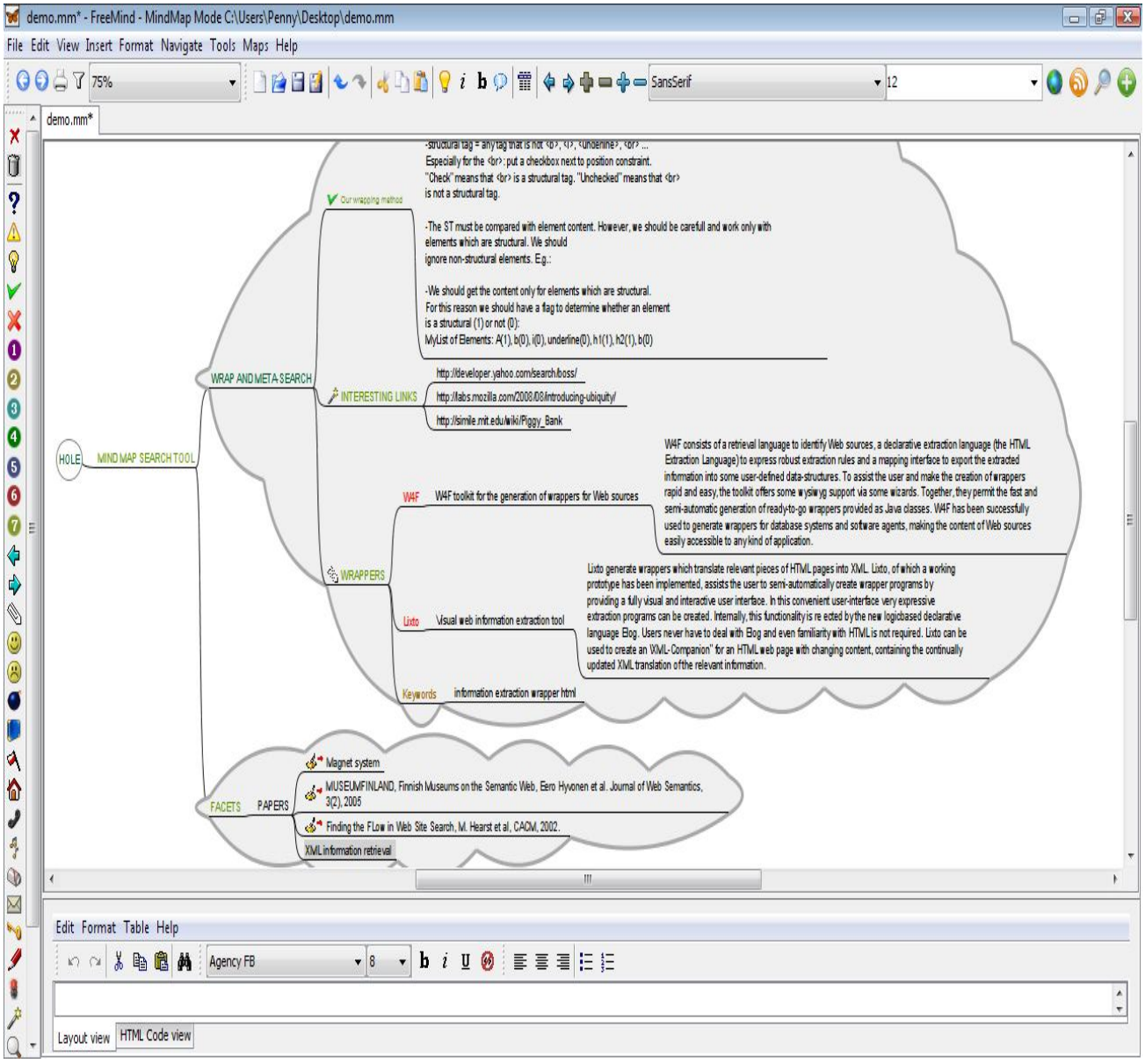
Ο χρήστης ξεκινάει δημιουργώντας ένα χάρτη σκέψης και αναζητά έναν από τους όρους του χάρτη στον Ιστό. Στη συνέχεια, αφού επιστραφούν τα αποτελέσματα και καθοριστούν οι τιμές των όψεων, ο χρήστης προσπαθεί να ταξινομήσει τα αποτελέσματα θέτοντας διάφορα κριτήρια. Κατά την προσπάθεια ταξινόμησης, προσθέτει, αφαιρεί και αλλάζει κριτήρια.

Αφού καταλήξει στη συλλογή αποτελεσμάτων που τον ενδιαφέρουν, ο χρήστης επιλέγει να πραγματοποιήσει και συσταδοποίηση με σκοπό να καθορίσει ποιες από τις συνολικές δημοσιεύσεις σχετίζονται περισσότερο με τα papers που τον ενδιαφέρουν.

6.2 *Αναλυτική παρουσίαση ελέγχου*

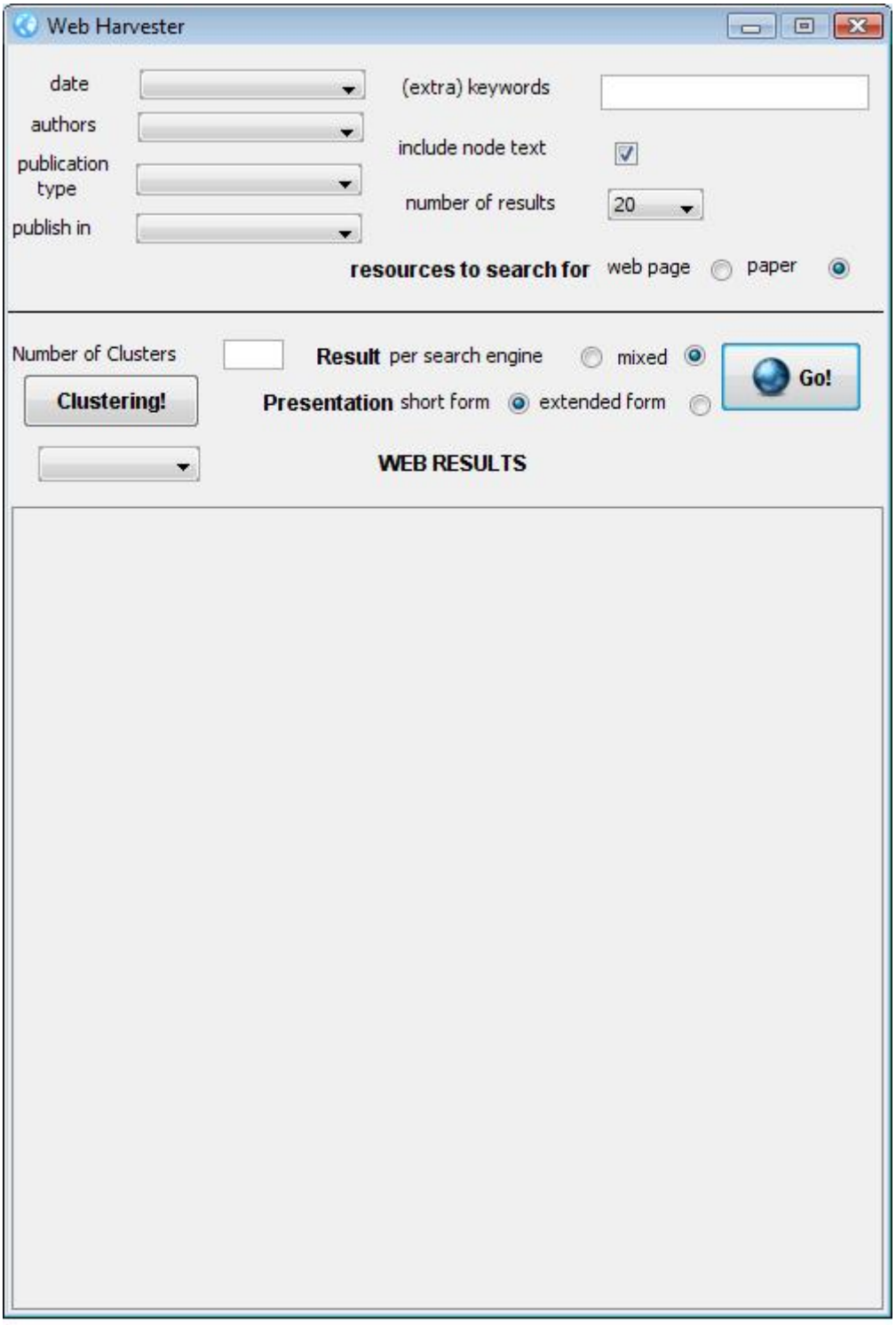
Στην ενότητα αυτή παρουσιάζουμε αναλυτικά τον έλεγχο του συστήματος σύμφωνα με το σενάριο που περιγράφηκε στην προηγούμενη ενότητα.

Αρχικά ο χρήστης δημιουργεί ένα χάρτη σκέψεων ο οποίος αφορά όρους σχετικούς με Mindmap Search Tools.



Σχήμα 6.1 Παράδειγμα Χάρτη Σκέψεων για όρους σχετικούς με Mindmap Search Tools

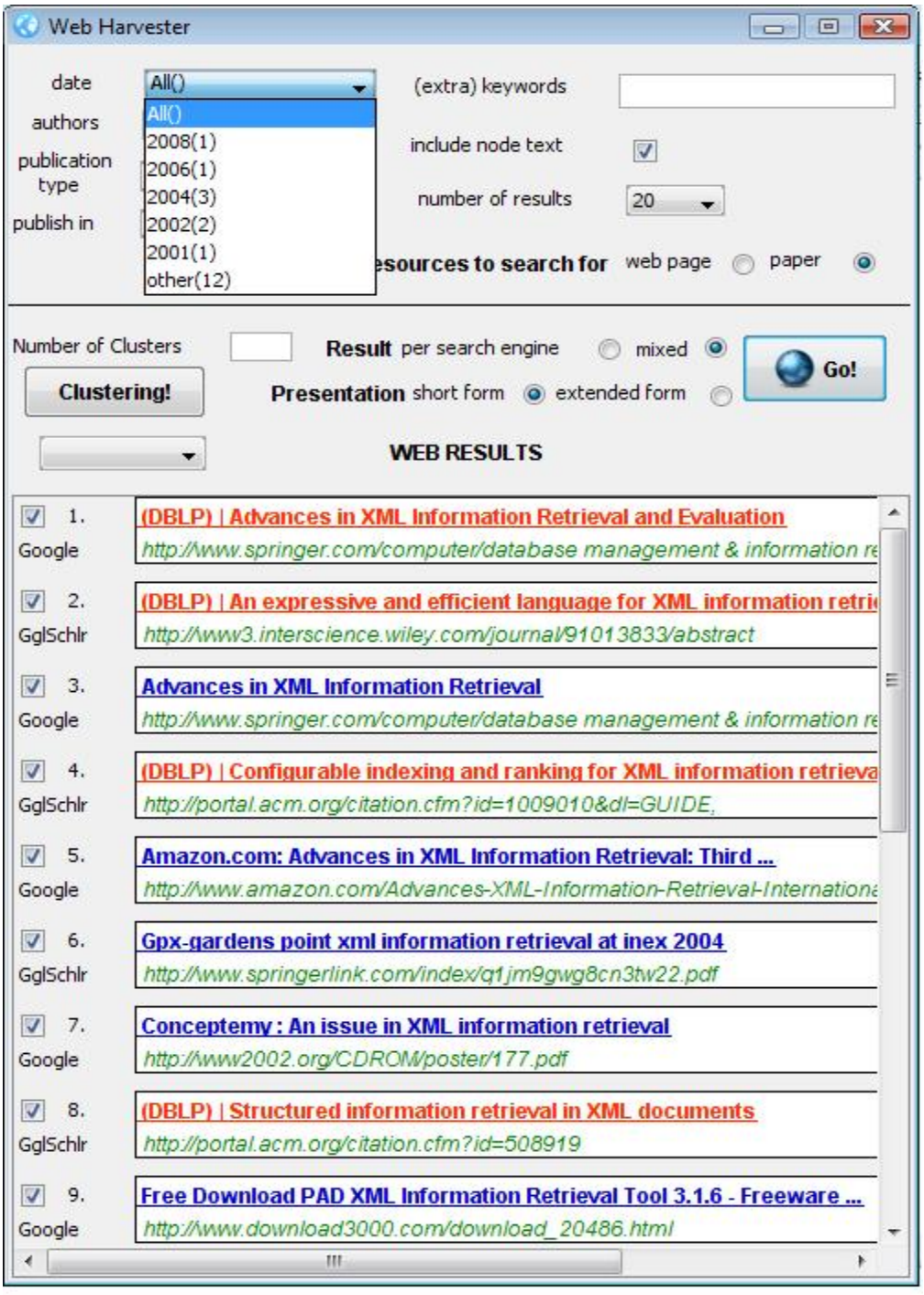
Από τους όρους τους οποίους προσθέτει στο χάρτη, επιλέγει να αναζητήσει στον Ιστό επιστημονικές δημοσιεύσεις για τον όρο "XML Information Retrieval". Ο αριθμός δημοσιεύσεων που ζητά είναι είκοσι. Στο επόμενο σχήμα βλέπουμε τις επιλογές αυτές στο interface:



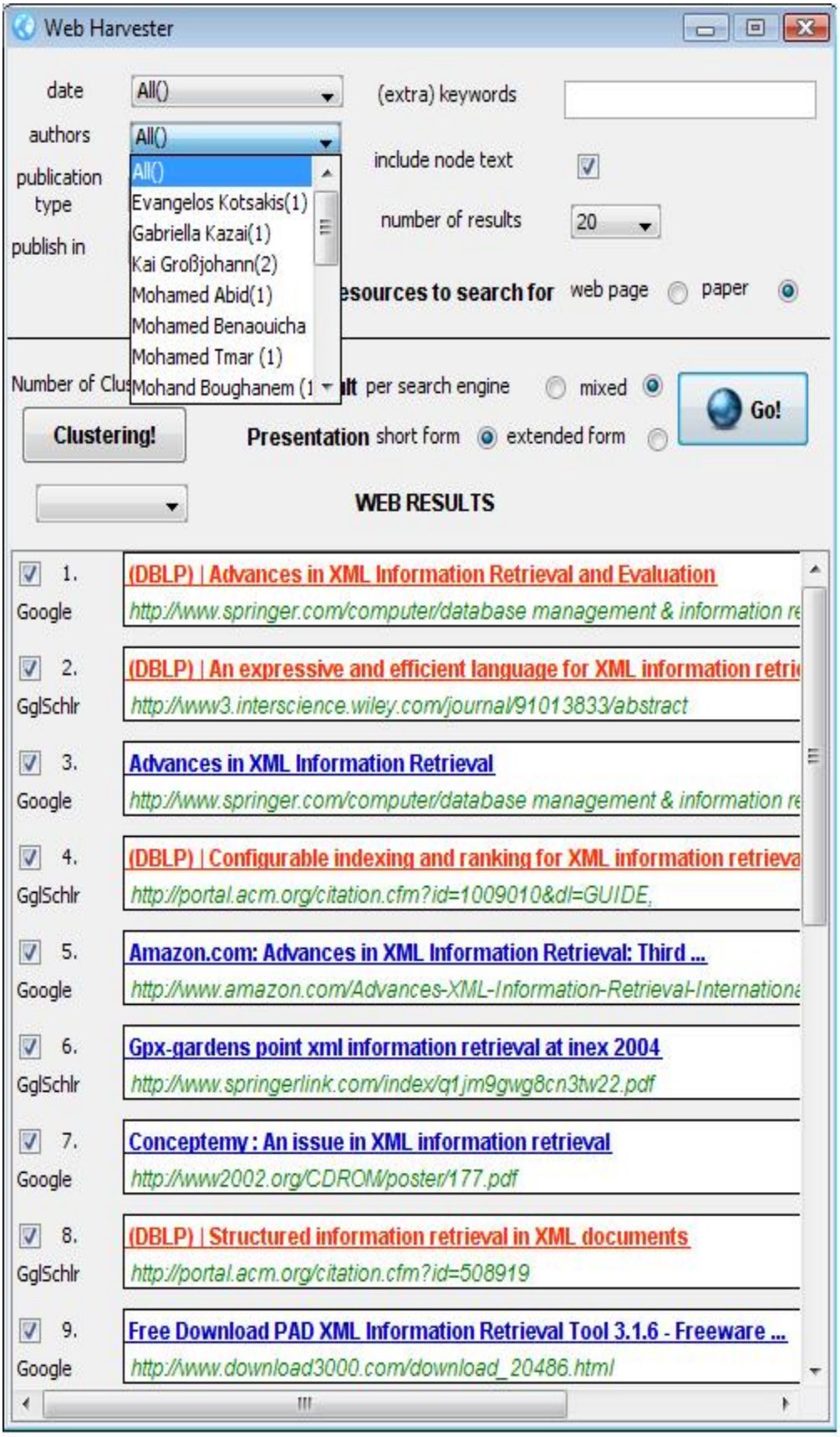
Σχήμα 6.2 Επιλογές Αναζήτησης στο interface

Στο πεδίο των επιπλέον λέξεων κλειδιών δεν εισάγαμε κάποια τιμή για λόγους ευκολίας κι έτσι επιλέξαμε το check box "include node text" προκειμένου ως λέξη κλειδί να θεωρηθεί το κείμενο του κόμβου.

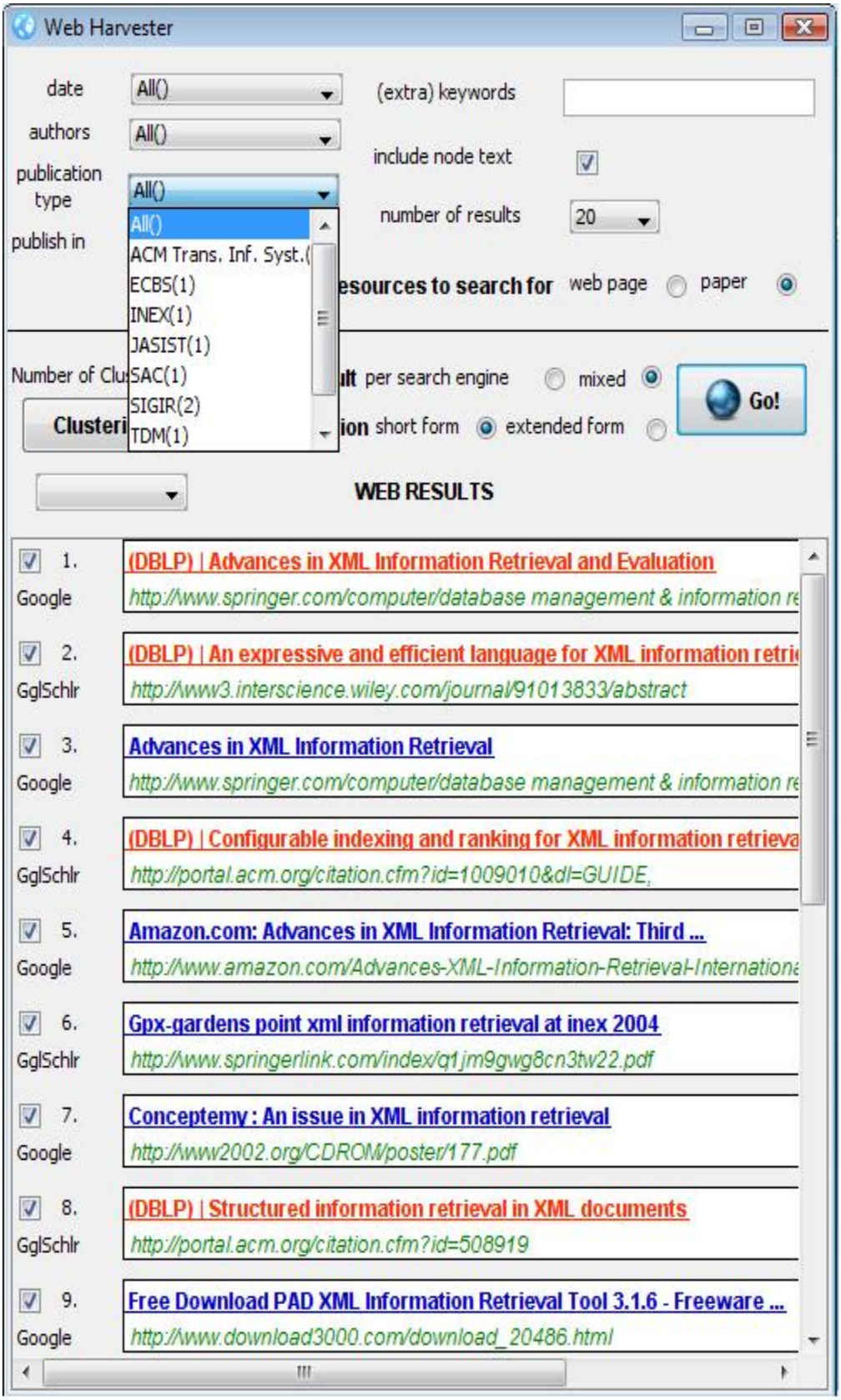
Παρατηρούμε σε αυτό το αρχικό παράθυρο ότι τα Combo Boxes είναι άδεια (σχήμα 6.2). Στα επόμενα σχήματα, βλέπουμε το ίδιο παράθυρο, αφού έχει πατηθεί το κουμπί Go! κι έχουν επιστραφεί τα αποτελέσματα της αναζήτησης. Τα Combo Boxes έχουν τώρα γεμίσει με τα αντίστοιχα facet values. Παρατηρούμε επίσης την εμφάνιση της τιμής 'other()' σε κάθε όψη. Η τιμή αυτή αναφέρεται στα papers τα οποία δεν ήταν δυνατόν να ταξινομηθούν σε κάποια κατηγορία.



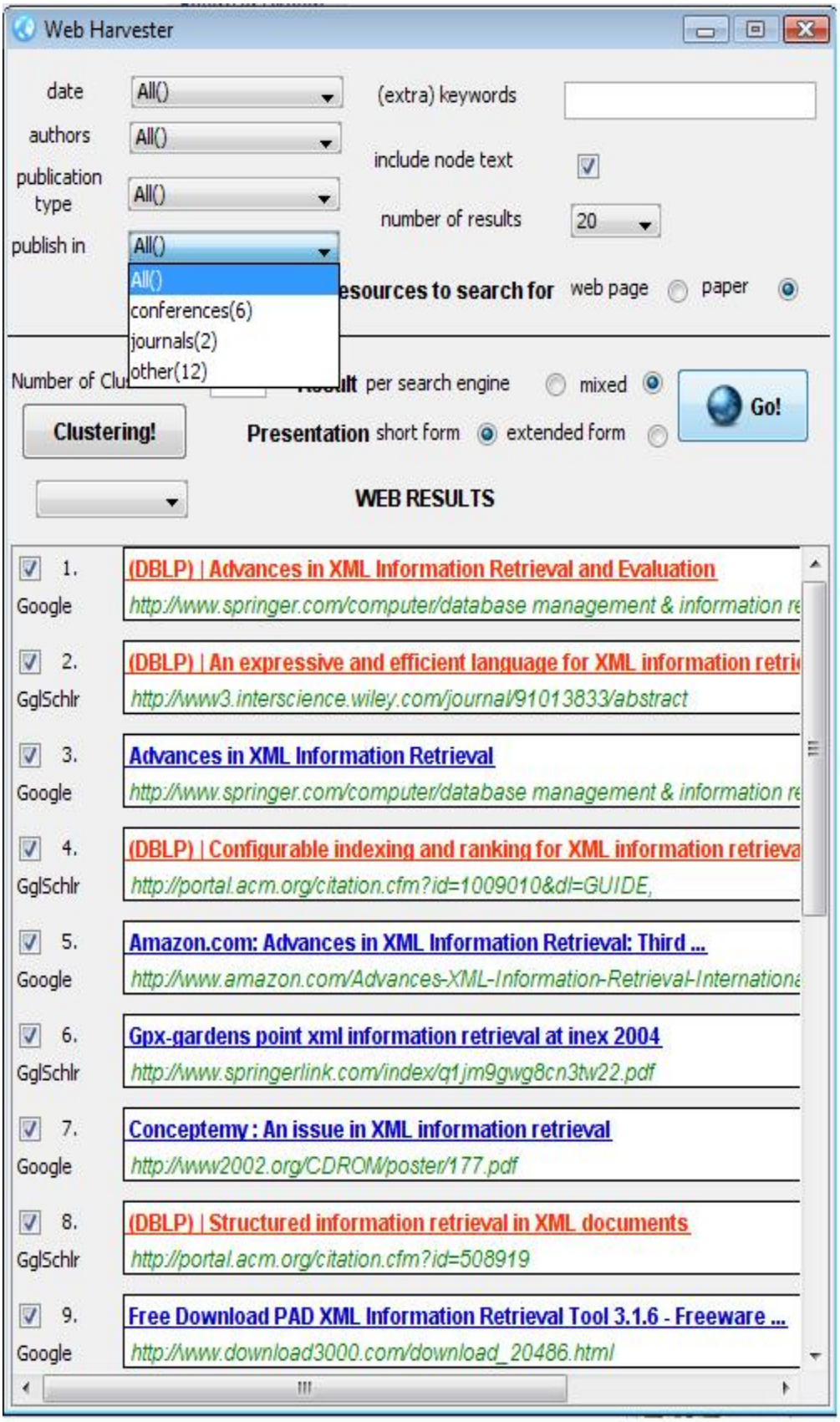
Σχήμα 6.3α Γέμισμα Combo Box με ημερομηνίες



Σχήμα 6.3β Γέμισμα Combo Box με συγγραφείς

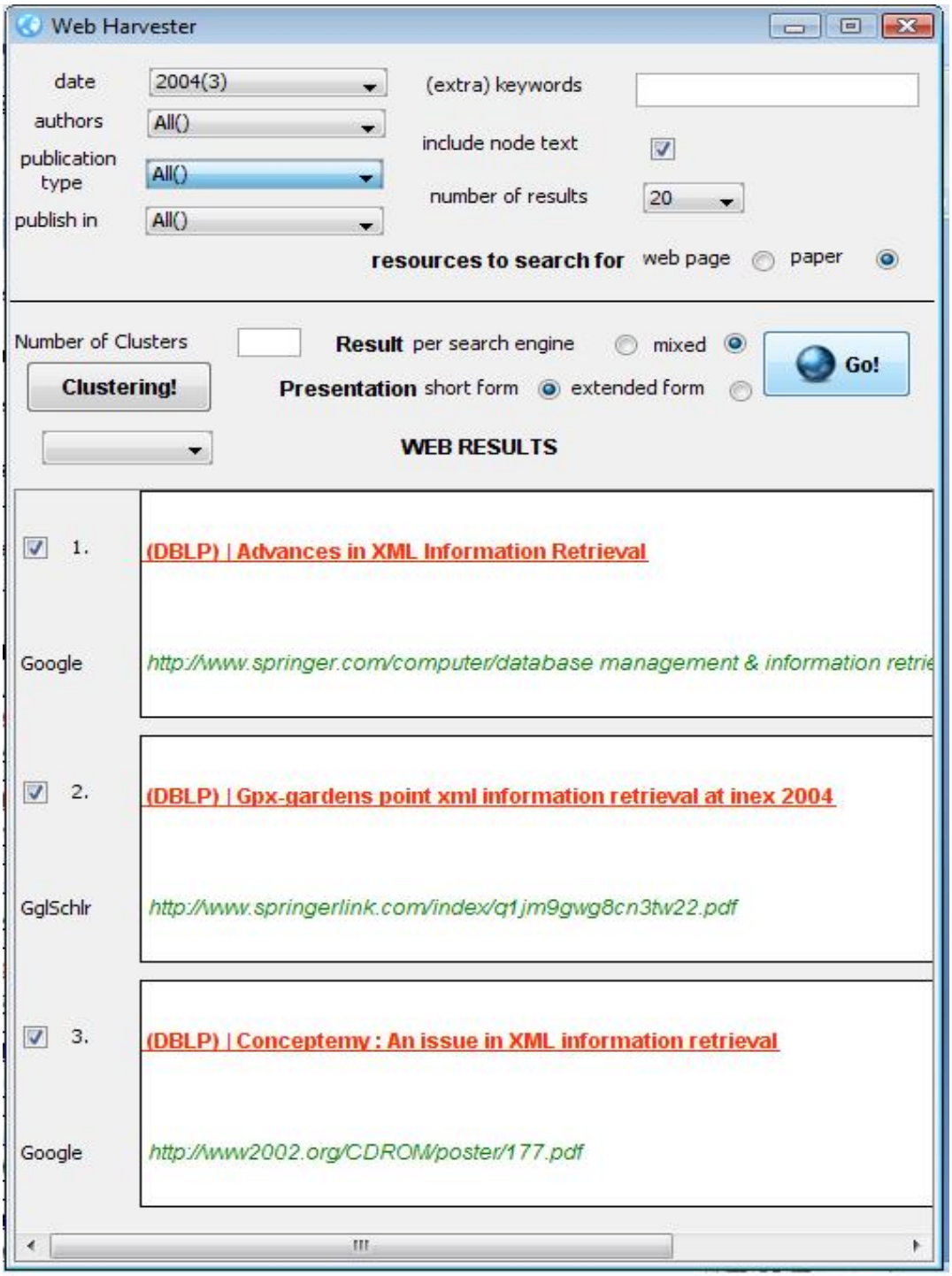


Σχήμα 6.3γ Γέμισμα Combo Box με ονόματα συνεδρίων ή περιοδικών δημοσίευσης


Web Harvester
date
All()
(extra) keywords
authors
All()
include node text
publication type
All()
number of results
20
publish in
All()
All()
conferences(6)
journals(2)
other(12)
esources to search for
web page
paper
Number of Clu
lt per search engine
mixed
Go!
Clustering!
Presentation
short form
extended form
WEB RESULTS
1. Google (DBLP) | Advances in XML Information Retrieval and Evaluation
http://www.springer.com/computer/database management & information re
2. GglSchlr (DBLP) | An expressive and efficient language for XML information retri
http://www3.interscience.wiley.com/journal/91013833/abstract
3. Google Advances in XML Information Retrieval
http://www.springer.com/computer/database management & information re
4. GglSchlr (DBLP) | Configurable indexing and ranking for XML information retrieva
http://portal.acm.org/citation.cfm?id=1009010&dl=GUIDE,
5. Google Amazon.com: Advances in XML Information Retrieval: Third ...
http://www.amazon.com/Advances-XML-Information-Retrieval-Internationa
6. GglSchlr Gpx-gardens point xml information retrieval at inex 2004
http://www.springerlink.com/index/q1jm9gwg8cn3tw22.pdf
7. Google Conceptemy : An issue in XML information retrieval
http://www2002.org/CDROM/poster/177.pdf
8. GglSchlr (DBLP) | Structured information retrieval in XML documents
http://portal.acm.org/citation.cfm?id=508919
9. Google Free Download PAD XML Information Retrieval Tool 3.1.6 - Freeware ...
http://www.download3000.com/download_20486.html

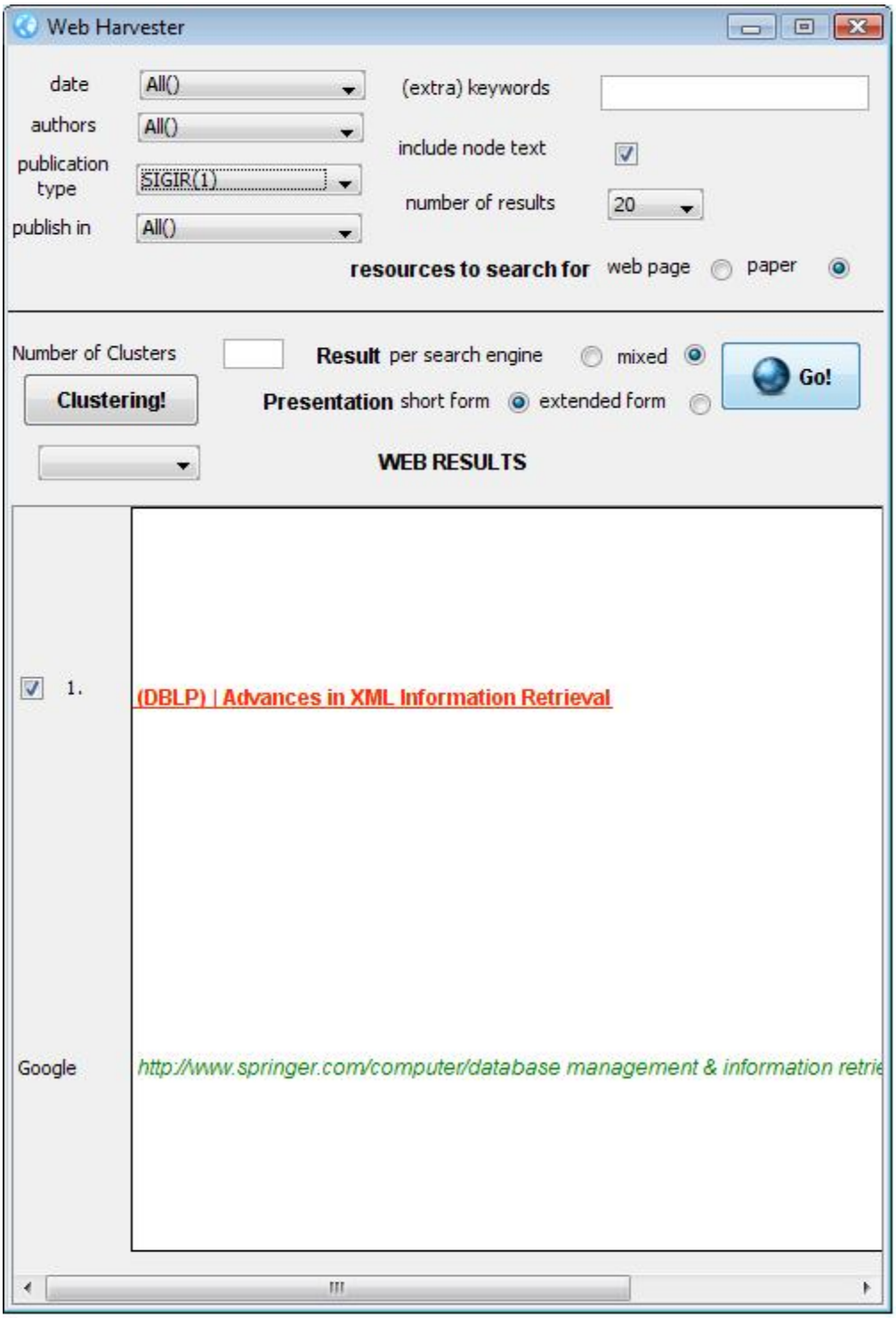
Σχήμα 6.3δ Γέμισμα Combo Box με τύπο δημοσίευσης (επιστημονική εφημερίδα/περιοδικό ή συνέδριο)

Ας υποθέσουμε τώρα ότι ο χρήστης επιλέγει την τιμή '2004' από το facet 'ημερομηνία'. Οι τιμές των υπόλοιπων facets όπως και τα εμφανιζόμενα στο χρήστη αποτελέσματα, αλλάζουν κατάλληλα. Στο σχήμα 2.4 βλέπουμε την αλλαγή στα εμφανιζόμενα στο χρήστη αποτελέσματα.



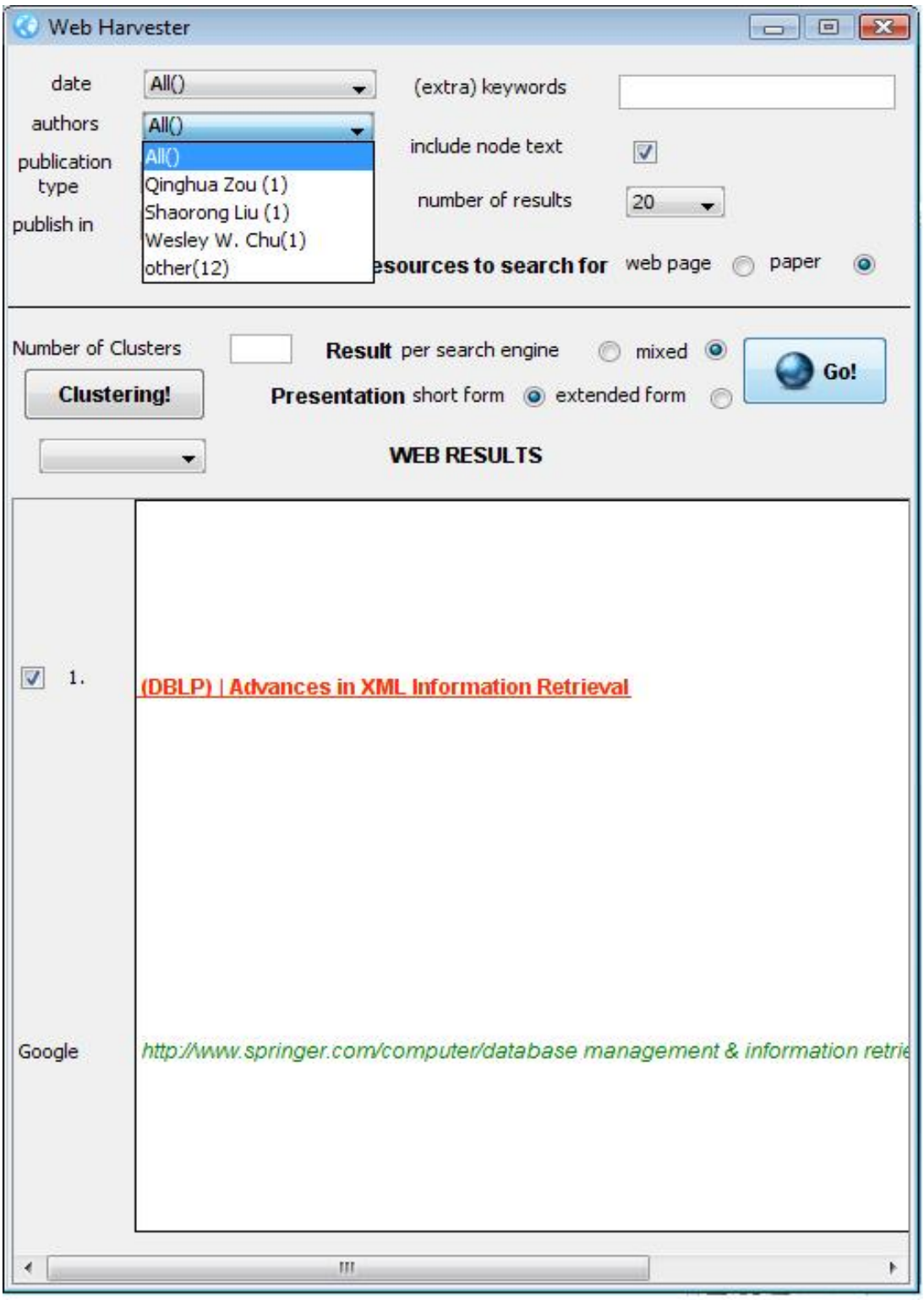
Σχήμα 6.4 Επιλογή όψης 'ημερομηνία'

Από την όψη αυτή 'δημοσίευση σε', επιλέγουμε τώρα την τιμή "SIGIR" η οποία αντιστοιχεί σε ένα αποτέλεσμα.



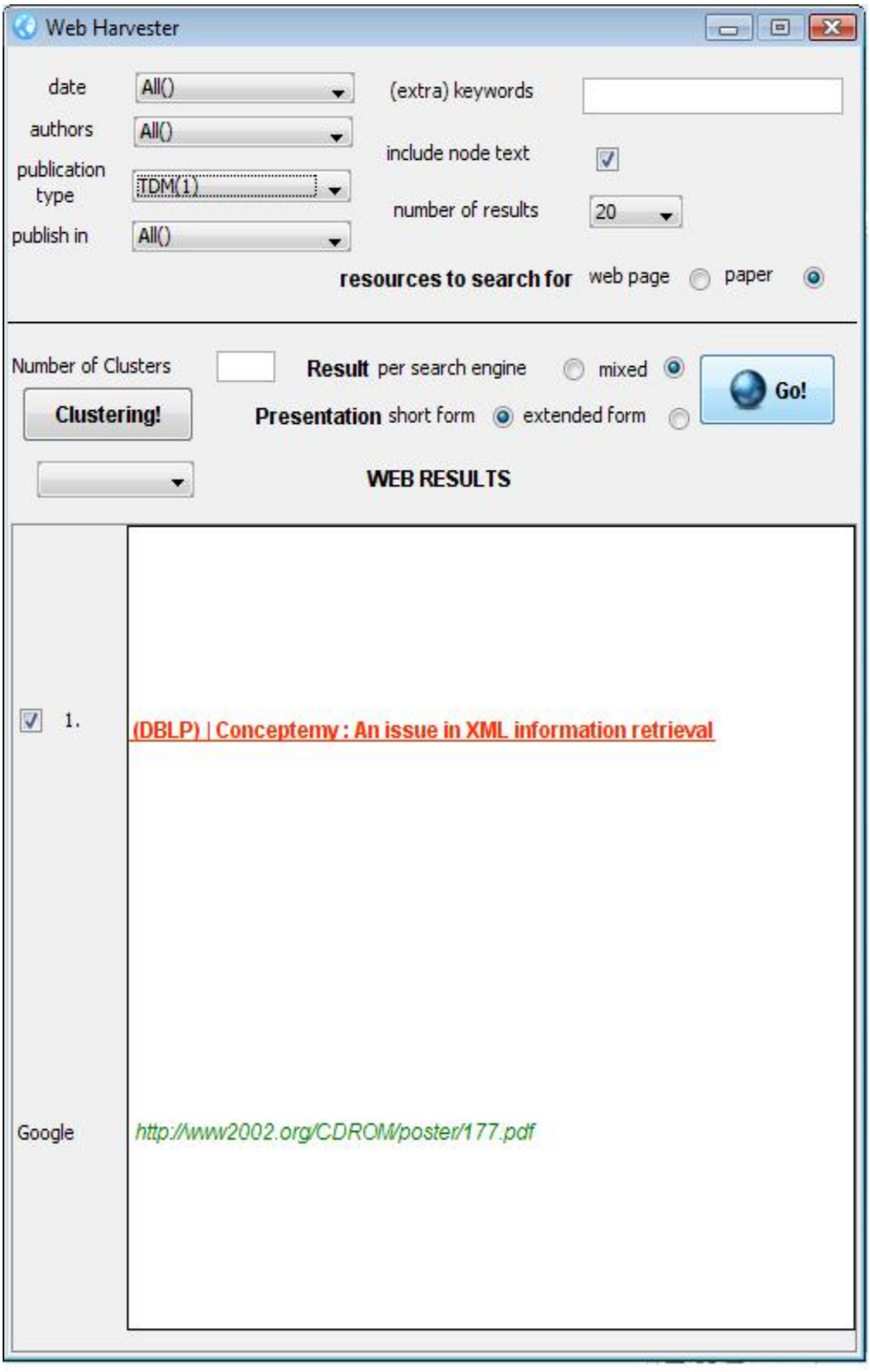
Σχήμα 6.5 Επιλογή της όψης 'δημοσίευση σε'

Στην όψη συγγραφείς, τα αποτελέσματα ανανεώθηκαν και παρέμειναν μόνο οι συγγραφείς των οποίων τα papers είναι στα αποτελέσματα. Αντίστοιχα αλλάζουν και οι τιμές στις υπόλοιπες όψεις.



Σχήμα 6.6 Επιλογή facet values

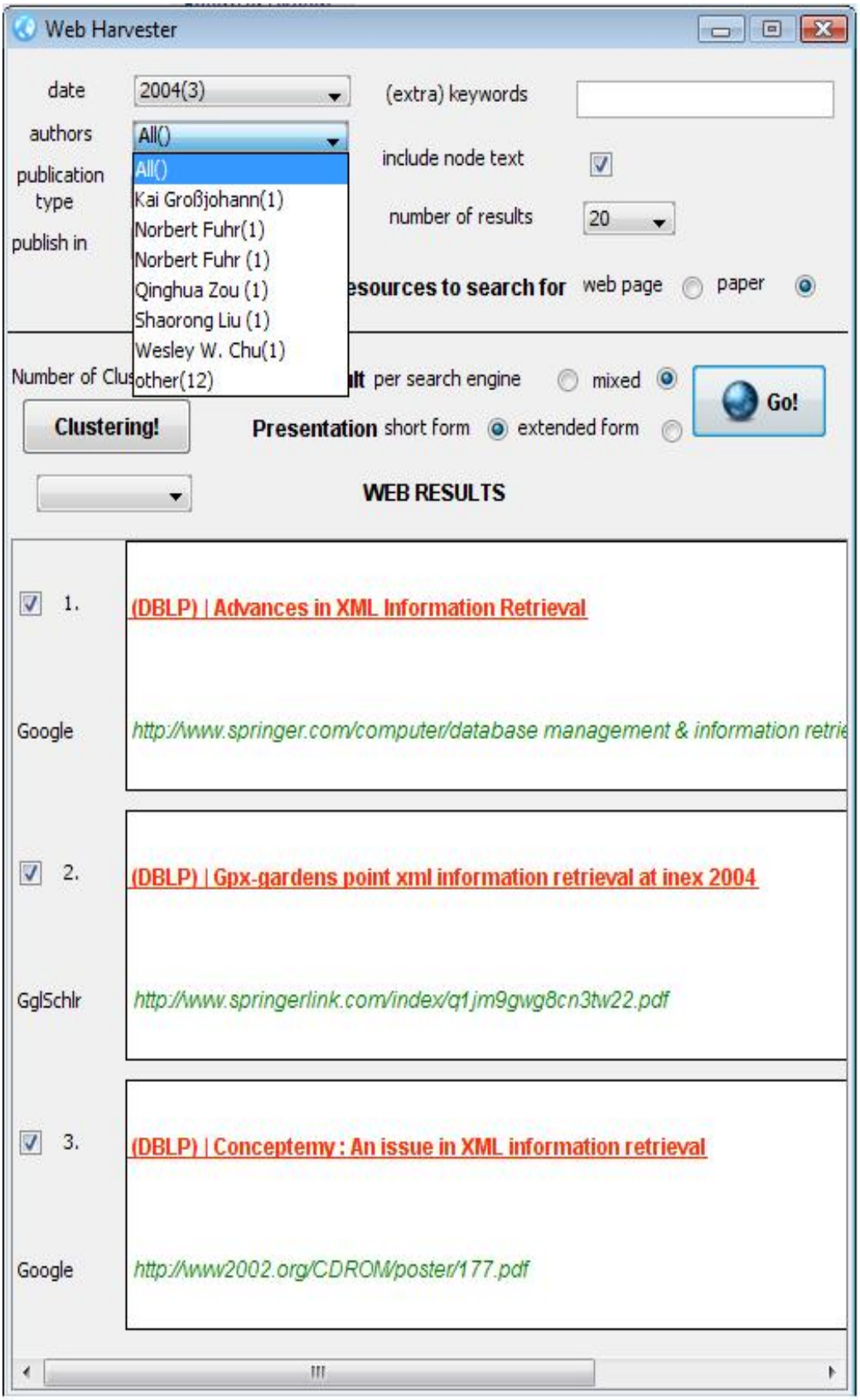
Αν αλλάξουμε επιλογή από την όψη με τις δημοσιεύσεις (Σχήμα 6.7), βλέπουμε ότι αλλάζουν και οι αντίστοιχες επιλογές συγγραφέων(Σχήμα 6.8).



Σχήμα 6.7 Αφαίρεση ενός facet values

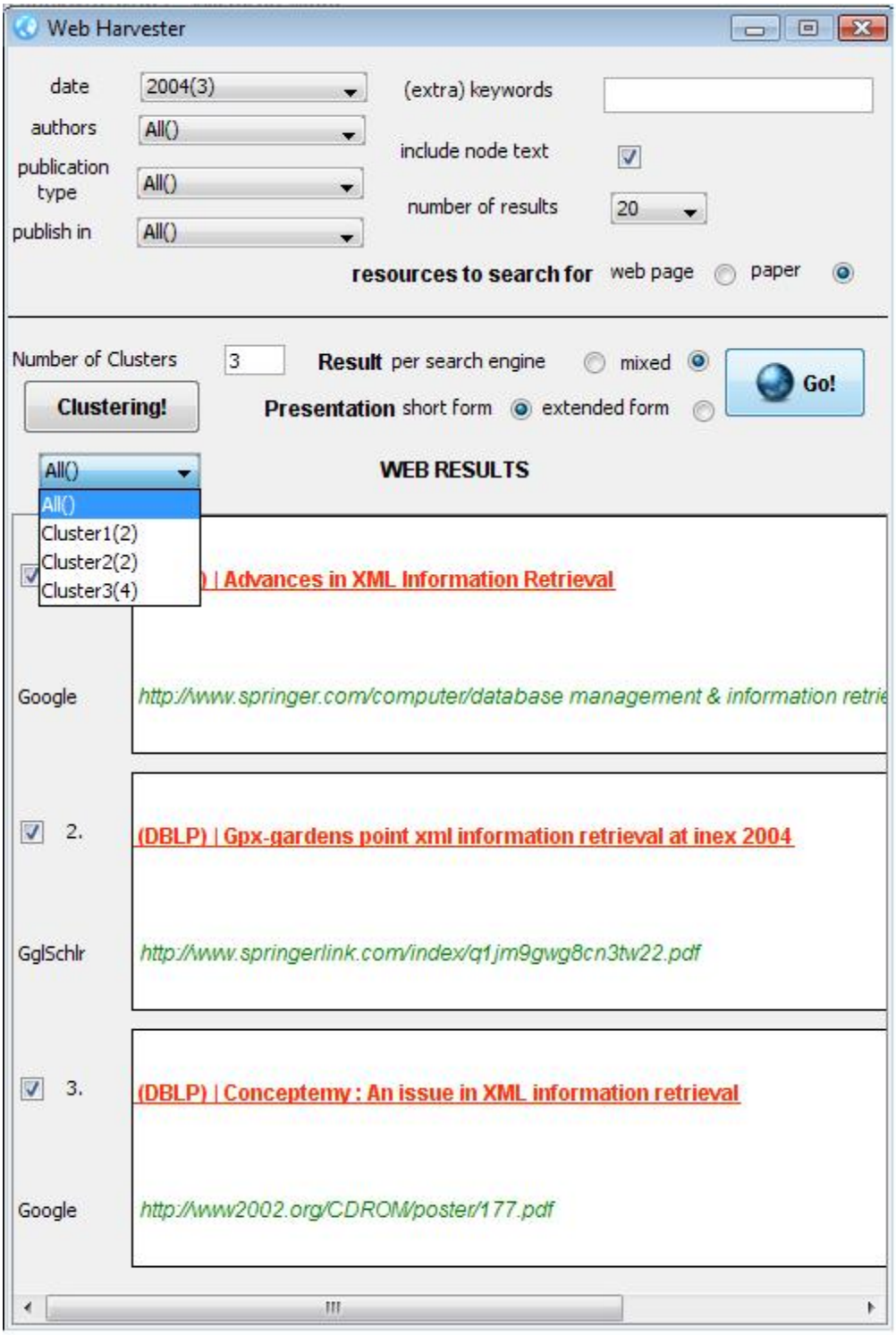
Σχήμα 6.8 Αλλαγή στα facet values και στα εμφανιζόμενα αποτελέσματα

Αν στη συνέχεια ο χρήστης αποφασίσει να αφαιρέσει τελείως το κριτήριο 'δημοσίευση σε' επιλέγοντας "All()" από το αντίστοιχο facet, τότε στην όψη 'συγγραφείς' έχουμε όλους τους συγγραφείς των οποίων οι δημοσιεύσεις έγιναν το 2004. Αντίστοιχα αλλάζουν και τα εμφανιζόμενα αποτελέσματα.



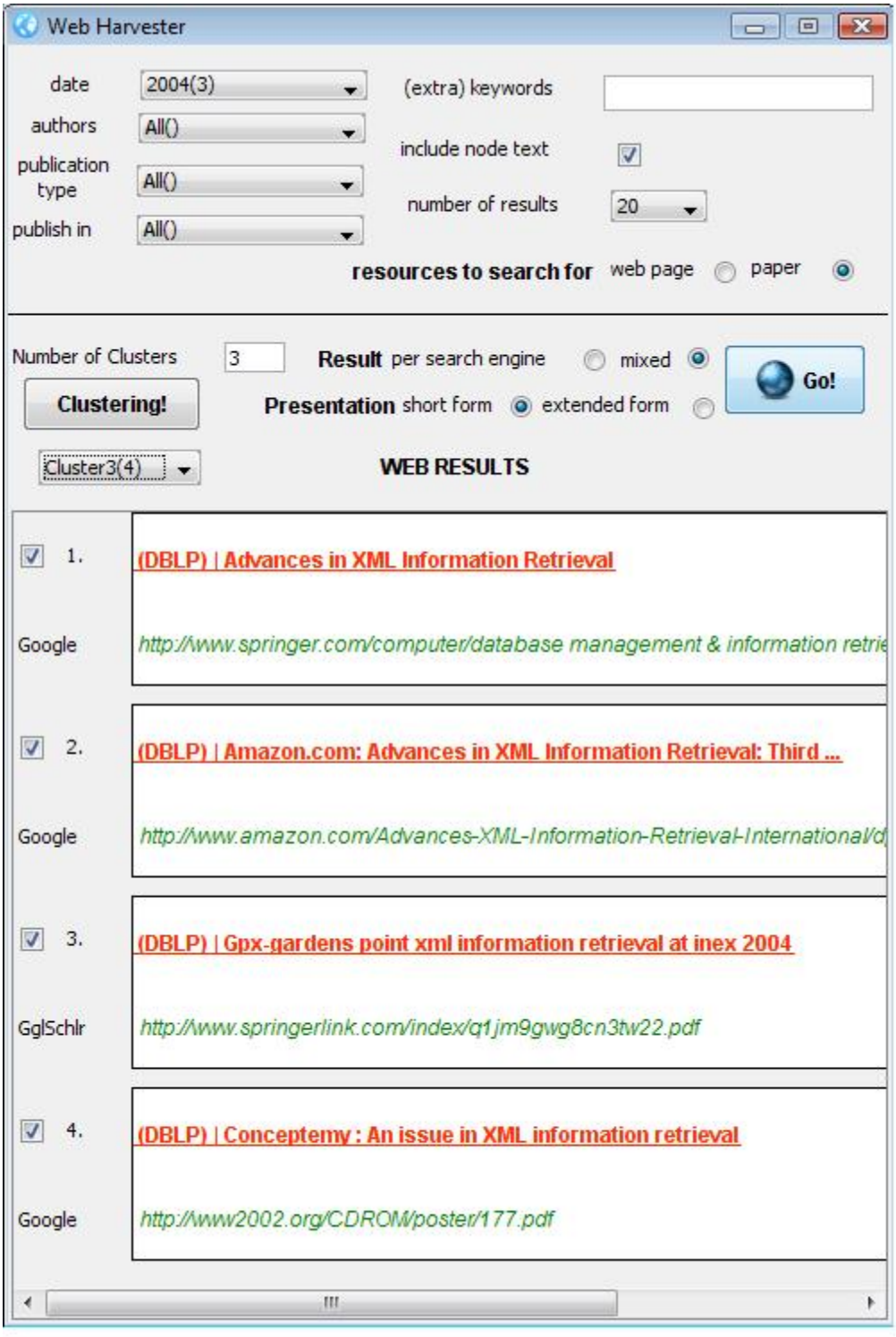
Σχήμα 6.9 Αφαίρεση κριτηρίου 'δημοσίευση σε'

Τέλος εξετάζουμε την περίπτωση που ο χρήστης επιλέξει να υλοποιήσει και clustering πατώντας το αντίστοιχο κουμπί. Το Combo Box στο οποίο θα εμφανιστούν οι προκύπτουσες συστάδες είναι προς το παρόν κενό (σχήμα 6.9). Μετά το πάτημα του κουμπιού και της επιλογής από το χρήστη για δημιουργία τριών συστάδων, έχουμε:



Σχήμα 6.10 Υλοποίηση Clustering

Επιλέγοντας μια συστάδα, εμφανίζονται τα αντίστοιχα αποτελέσματα στο χρήστη:



Σχήμα 6.11 Υλοποίηση Clustering

7

Επίλογος

Στο κεφάλαιο αυτό συνοψίζεται η παρουσίαση της διπλωματικής εργασίας.

7.1 Σύνοψη και συμπεράσματα

Η παρούσα διπλωματική εργασία πραγματεύεται το χειρισμό των αποτελεσμάτων μιας αναζήτησης στον Παγκόσμιο Ιστό χρησιμοποιώντας της τεχνικές του faceted classification και του clustering. Η υλοποίηση επεκτείνει περεταίρω το ανοιχτού κώδικα εργαλείο χαρτογράφησης σκέψεων που ονομάζεται Freemind και τις επεκτάσεις που έχουν ήδη ενσωματωθεί σε αυτό από αντίστοιχη διπλωματική εργασία.

Το σύστημα ξεκινά τη λειτουργία του με μια αναζήτηση στον Παγκόσμιο Ιστό, στην οποία όμως τα αποτελέσματα έχει ζητηθεί να αφορούν αποκλειστικά επιστημονικές δημοσιεύσεις (papers). Μέσω αλληλεπίδρασης με τη Βάση Δεδομένων DBLP, αντλούνται πληροφορίες για τα πεδία ‘ημερομηνία’, ‘συγγραφείς’, ‘δημοσίευση σε’ και ‘τύπος δημοσίευσης’ και δημιουργούμε τέσσερις όψεις για αυτά. Οι όψεις αυτές αλληλεπιδρούν μεταξύ τους και η επιλογή μιας τιμής μιας όψης, αλλάζει τόσο τις τιμές των υπόλοιπων όψεων όσο και τα εμφανιζόμενα στο χρήστη αποτελέσματα. Στη συνέχεια, γίνεται προσπάθεια άντλησης επιπλέον πληροφοριών απευθείας από τα κείμενα των επιστημονικών δημοσιεύσεων, για όσα κείμενα βρίσκονται υπό μορφή PDF στο Διαδίκτυο και επιπλέον είναι αποθηκευμένα σε HTML μορφή. Οι πληροφορίες τις οποίες θέλουμε να εξάγουμε από τα PDFs είναι η ‘θεματική ενότητα’, οι ‘γενικοί όροι’, οι ‘λέξεις κλειδιά’ και η ‘περίληψη’ (Abstract). Με τις

πληροφορίες για τα δύο πρώτα, δημιουργούνται επιπλέον δύο όψεις οι οποίες έχουν τις ίδιες ακριβώς ιδιότητες με τις προηγούμενες τέσσερις. Όλες οι παραπάνω πληροφορίες για την χρησιμοποιούνται στη διαδικασία της συσταδοποίησης. Οι συστάδες που τελικά δημιουργούνται, παρουσιάζονται στο χρήστη με τρόπο παρόμοιο με αυτό των όψεων.

Στο σημείο αυτό πρέπει να αναφερθούμε και σε ένα μειονέκτημα του συστήματος. Προκειμένου να αντληθούν πληροφορίες απευθείας από τις επιστημονικές δημοσιεύσεις, ψάχνουμε σε όσα από τα papers υπάρχουν υπό μορφή PDF στο Διαδίκτυο και μάλιστα είναι αποθηκευμένα στο Google και σε HTML μορφή. Όμως δεν πληρούν τις προϋποθέσεις αυτές όλα τα papers. Αυτό έχει το μειονέκτημα, ότι όλες οι όψεις δεν αναφέρονται στον ίδιο αριθμό αποτελεσμάτων κι αυτό είναι κάτι που ίσως δεν είναι πολύ ξεκάθαρο στο χρήστη. Ο λόγος που ασχοληθήκαμε μόνο με τα συγκεκριμένα papers για την υλοποίηση της συσταδοποίησης ήταν η απόδοση του συστήματος, αφού για να λάβουμε υπόψη τα υπόλοιπα, έπρεπε να τα κατεβάζουμε τοπικά στο δίσκο, διαδικασία που επιφέρει μεγάλο κόστος χρόνου και μνήμης.

Η συγκεκριμένη διπλωματική εργασία συνεισφέρει σημαντικά στην περιοχή του χειρισμού των αποτελεσμάτων μίας αναζήτησης, καθώς αντιμετωπίσθηκαν και επιλύθηκαν με επιτυχία πολλά σημαντικά προβλήματα οργάνωσης των αποτελεσμάτων διευκολύνοντας το χρήστη στο χειρισμό μεγάλου όγκου αυτών.

7.2 Μελλοντικές επεκτάσεις

Στη συνέχεια παρουσιάζουμε κάποιες επεκτάσεις που θα θέλαμε να ενσωματωθούν μελλοντικά στο σύστημα. Κάποιες από αυτές έχουν σκοπό να βελτιώσουν το υπάρχον σύστημα, ενώ άλλες να προσθέσουν σε αυτό επιπλέον λειτουργίες.

- ✓ Αρχικά θα μπορούσαμε να ενσωματώσουμε στην υλοποίηση έναν αλγόριθμο πρόβλεψης του βέλτιστου αριθμού συστάδων κατά τη διαδικασία της συσταδοποίησης. Τώρα η επιλογή του αριθμού των συστάδων έχει ανατεθεί στο χρήστη ο οποίος κρίνει με βάση τα αποτελέσματα που βλέπει και τις ανάγκες του. Ίσως όμως η επιλογή του να μην είναι η βέλτιστη.
- ✓ Επειδή όπως είχαμε πληροφορηθεί από την έναρξη ακόμα της υλοποίησης της διπλωματικής εργασίας, η ψηφιακή βιβλιοθήκη DBLP θα δημιουργήσει στο προσεχές μέλλον ένα API για αλληλεπίδραση με εφαρμογές όπως η δική μας σε αυτή, θα ήταν καλό να γίνει ενσωμάτωση του API στο σύστημα όταν αυτό γίνει διαθέσιμο.
- ✓ Επίσης, θα θέλαμε να πραγματοποιηθούν πειράματα με μεγάλου μεγέθους συλλογές, προκειμένου να εκτιμηθεί η ποιότητα του Clustering.

Πέρα από αυτές τις δύο κινήσεις που είναι βελτιωτικές, προτείνουμε και μερικές ακόμα που αφορούν την επέκταση του συστήματος και των δυνατοτήτων που γίνονται διαθέσιμες στους

χρήστες. Οι επεκτάσεις οι οποίες προτείνονται βασίζονται στην εφαρμογή Magnet στην οποία έγινε αναφορά στο κεφάλαιο 2. Προκειμένου οι προτάσεις να γίνουν πλήρως κατανοητές, συνίσταται να ανατρέξει ο αναγνώστης στο σχήμα 2.8.

- ✓ Η πρώτη πρόταση αφορά την εμφάνιση του 'ιστορικού' του χρήστη στο interface ανά πάσα στιγμή. Με τον όρο 'ιστορικό' εννοούμε το σύνολο των επιλεχθέντων τιμών από τις διάφορες όψεις, όπως φαίνεται στο πάνω αριστερό τμήμα του σχήματος 2.8. Ο χρήστης θα μπορούσε επίσης από εκεί να ακυρώσει μια επιλογή (πράγμα βέβαια το οποίο ήδη υλοποιείται μέσω του Combo Box της αντίστοιχης όψης).
- ✓ Επιπλέον, προτείνεται η υλοποίηση των 'συμπληρωματικών' επιλογών όπως φαίνεται στον Advisor "Modify" του ίδιου σχήματος. Ο χρήστης θα μπορεί με τη επιλογή αυτή να συνδυάζει περισσότερες από μια τιμές από κάθε όψη στα συνολικά αποτελέσματα.

8

Βιβλιογραφία

[WP1] Wikipedia, The Free Encyclopedia – "faceted Classification" < http://en.wikipedia.org/wiki/Faceted_classification >

[SK05] Vineet Sinha, David R. Krager *"Magnet: Supporting Navigation in Semistructured Data Environments"* SIGMOD 2005, Baltimore, Maryland USA

[Hea06] Marti A. Hearst, "Clustering Versus Faceted Categories for Information Exploration", Communications of ACM April 2006/Vol. 49, No 4

[Wil06] Travis Wilson, "The Strict Faceted Classification", available at http://facetmap.com/pub/ .

[DIW05] Wisam Dakka, Panagiotis G. Ipeirotis, Kenneth R. Woods, "Automatic Construction of Multifaceted Browsing Interfaces", CIKM 2005, Copyright 2005, Bremen, Germany

[YSLH03] Ka-Ping-Yee, Kirsten Swearingen, Kevin Li, Marti Hearst, *"Faceted Metadata for Image Search and Browsing"*, CHI 2003, copyright 2003, Ft. Lauderdale, Florida, USA

[WP2] Wikipedia, The Free Encyclopedia – "faceted search" http://en.wikipedia.org/wiki/Faceted_search

[WP3] Wikipedia, The Free Encyclopedia – "taxonomies"
http://en.wikipedia.org/wiki/Taxonomies

[WP4] Wikipedia, The Free Encyclopedia – "cluster analysis"
http://en.wikipedia.org/wiki/Cluster_analysis

[WP5] Wikipedia, The Free Encyclopedia – "CGI"
http://en.wikipedia.org/wiki/Common_Gateway_Interface

[WP6] Wikipedia, The Free Encyclopedia – "html language"
http://en.wikipedia.org/wiki/Html_language

[WP7] Wikipedia, The Free Encyclopedia – "regular expression"
http://en.wikipedia.org/wiki/Regural_Expression

[Hea3] Marti A. Hearst, *"Design Recommendations for Hierarchical Faceted Search Interfaces"*

[MHS] Eetu Makela, Eero Hyvonen, Teemu Sidoroff, "View-Based User Interfaces for Information Retrieval on the Semantic Web"

[WP8] Wikipedia, The Free Encyclopedia – "mindmap"
http://en.wikipedia.org/wiki/Mind_map

[YGEL08] Ori Ben-Yitzak, Nadav Golbandi, Navad Har'El, Ronny Lempel, *"Beyond Basic Faceted Search"*, WSDM 2008, Palo Alto, California USA

[WP9] Wikipedia, The Free Encyclopedia – "Depth-first search"
http://en.wikipedia.org/wiki/depth-first_search

[SKH03] Vineet Sinha, David R. Karger, David F. Huynh, *"Assisted Browsing for Semistructured Data",* WWW 2003, Budapest, Hungary

[DDI06] Wisam Dakka, Rishabh, Panagiotis G. Ipeirotis, "Automatic Discovery of Useful Facet Terms", SIGIR 2006, Seattle, Washington USA