Διασπορά πληροφορίας βασισμένη σε σημασιολογικές συσχετίσεις

Κατζαγιαννάκη Γ. Ειρήνη – Ηλέκτρα

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας, οι χρήστες αποστέλλουν προφίλ στα οποία δηλώνουν τα ενδιαφέροντά τους. Αυτά τα ενδιαφέροντα μπορούν να θεωρηθούν ως συνεχείς επερωτήσεις προς ένα σύστημα διασποράς πληροφορίας. Το σύστημα συλλέγει συνεχώς νέα κείμενα από τις πηγές πληροφορίας, τα φιλτράρει σε σχέση με τα προφίλ των χρηστών και παραδίδει τη σχετική πληροφορία στους αντίστοιχους χρήστες. Τα συστήματα επιλεκτικής διασποράς πληροφορίας αποτελούν μία αναγκαιότητα σήμερα λόγω κυρίως του μεγάλου όγκου της πληροφορίας που διαδίδεται μέσω του παγκόσμιου ιστού, καθώς ενημερώνουν το χρήστη για την πληροφορία που τον ενδιαφέρει, χωρίς αυτός να δαπανά χρόνο να την εντοπίσει.

Τα περισσότερα συστήματα επιλεκτικής διασποράς πληροφορίας βασίζονται στη λεκτική αναζήτηση. Πιο συγκεκριμένα, εκφράζουν τα κείμενα και τα προφίλ των χρηστών ως σύνολα λέξεων και ελέγχουν την ταυτοσημότητα των λέξεων ανάμεσα στα δύο αυτά σύνολα για να αποφασίσουν την αποστολή ενός κειμένου σε ένα χρήστη.

Ωστόσο, συχνά οι χρήστες ενός συστήματος επιλεκτικής διασποράς πληροφορίας χρησιμοποιούν πολλούς διαφορετικούς όρους για να δηλώσουν την ίδια έννοια, όρους που χαρακτηρίζονται ως συνώνυμα. Παράλληλα, όταν κάποιος χρήστης αναζητά πληροφορία για κάποιον όρο, σίγουρα τον ενδιαφέρει και η πληροφορία που αναφέρεται σε όρους ειδικότερους από αυτόν. Επομένως είναι απαραίτητο ένα τέτοιο

σύστημα να διατηρεί μηχανισμούς που λαμβάνουν υπόψη τις σημασιολογικές συσχετίσεις ανάμεσα στους όρους κατά τη σύγκριση των προφίλ και των κειμένων.

Στην παρούσα εργασία υλοποιήθηκε ένα σύστημα επιλεκτικής διασποράς πληροφορίας το οποίο λαμβάνει υπόψη τις σημασιολογικές συσχετίσεις των όρων. Πιο συγκεκριμένα ένα προφίλ θεωρείται σχετικό με ένα κείμενο, όχι μόνο στην περίπτωση που οι όροι του εμφανίζονται στο κείμενο, αλλά και όταν τα συνώνυμα ή τα υπώνυμα (ειδικότεροι όροι) των όρων του παρουσιάζονται στο κείμενο.

Το σύστημα διαχειρίζεται προφίλ εκφρασμένα σε δύο από τα πιο διαδεδομένα μοντέλα στο χώρο της ανάκτησης πληροφορίας, στο Boolean μοντέλο και στο Vector Space μοντέλο. Για την αύξηση της απόδοσης του συστήματος δημιουργείται μία δομή ευρετηρίασης των προφίλ, και όχι των κειμένων, καθώς τα προφίλ είναι περισσότερα και πιο στατικά. Όταν εμφανίζεται κάποιο κείμενο από τις πηγές πληροφορίας, εκτελείται ένας αλγόριθμος σύγκρισης του κειμένου με τα προφίλ που υπάρχουν στη δομή, ο οποίος λαμβάνει υπόψη του τη σημασιολογία των όρων. Τελικά αποστέλλονται στους χρήστες τα κείμενα τα οποία περιέχουν όρους λεκτικά όμοιους ή σχετικούς με τους όρους του προφίλ τους. Η απόδοση του συστήματος έχει αξιολογηθεί βάσει πειραμάτων που πραγματοποιήθηκαν.

Επόπτης: Δημήτριος Πλεξουσάκης
Αναπληρωτής Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Information dissemination based on semantic relations

Katzagiannaki G. Irini – Ilektra

Master of Science Thesis

Computer Science Department
University of Crete

Abstract

In a selective information dissemination (SDI) system, users submit profiles consisting of a number of long standing queries to represent their information needs. The system then continuously collects new documents from underlying information sources, filters them against the user profiles, and delivers relevant information to corresponding users. SDI systems are very important nowadays due to the vast amount of information that flows in the World Wide Web, as they inform users for relevant information, without requiring them to spend time to locate it.

The majority of SDI systems are based on lexical search. In particular, they represent documents and user profiles as sets of terms and check the identicalness between these two sets, in order to make the decision for sending a document to a user.

Users of SDI systems may use many different terms to express the same meaning, terms that are called synonyms. Simultaneously, when users seek information about a term, they are also interested in information about terms that are hyponyms (special terms) of this initial term. As a result, it is necessary for such a system to contain mechanisms that take into account the semantic relationships between terms during matching of user profiles with documents.

In the present thesis, an SDI system has been implemented, which takes into account the semantic relationships between terms. In particular, a user profile is considered relevant with a document, if its terms or the synonyms or hyponyms of them appear in the document.

The system deals with profiles that are represented in the two most popular models in information retrieval, namely the Boolean model and the Vector Space model. In order to improve the system's performance, an index structure of profiles – rather than of documents – has been created, as profile information constitutes a larger volume and is more static. When a document arrives from the information sources, a matching algorithm for the document and the profiles in the index structure is executed. This algorithm takes into account the semantics of terms. Finally documents that contain the terms of a profile or relative terms of profile terms are delivered to user. The system has been evaluated based on experiments that have been conducted.

Supervisor: Dimitrios Plexousakis
Associate Professor of Computer Science
University of Crete

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επόπτη της εργασίας μου κ. Δημήτρη Πλεξουσάκη για την πολύτιμη βοήθειά του και την άψογη συνεργασία μας καθ' όλη τη διάρκεια της παρούσας εργασίας.

Παράλληλα θα ήθελα να ευχαριστήσω τα μέλη της ομάδας Πληροφοριακών Συστημάτων και Ανάπτυξης Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΤΕ) και ιδιαίτερα αυτούς με τους οποίους συνεργάστηκα.

Οφείλω επίσης ένα θερμό ευχαριστώ στα μέλη της εισηγητικής επιτροπής κ. Πάνο Κωνσταντόπουλο, καθηγητή του Τμήματος Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης, και κ. Βασίλη Χριστοφίδη, επίκουρο καθηγητή του ίδιου τμήματος, για τις εποικοδομητικές τους παρατηρήσεις.

Περισσότερο όμως θα ήθελα να δώσω ένα μεγάλο ευχαριστώ στους γονείς μου Γιώργο και Στέλλα, στη γιαγιά μου Όλγα, στον αδερφό μου Θανάση και στον αρραβωνιαστικό μου Μανώλη για την αγάπη, τη συμπαράσταση και τη δύναμη που μου δίνουν όλα αυτά τα χρόνια.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο Ι

1.1 Εισαγωγή

Η ανάπτυξη της τεχνολογίας και η εξάπλωση του παγκόσμιου ιστού έχει ως αποτέλεσμα την εύκολη ανάκτηση και διάδοση πληροφορίας. Οι χρήστες έχουν εύκολη πρόσβαση στην πληροφορία, αλλά παράλληλα αντιμετωπίζουν το πρόβλημα της υπερφόρτωσης πληροφορίας. Αυτό το γεγονός καθιστά χρονοβόρα τη διαδικασία εύρεσης της πληροφορίας που τους ενδιαφέρει, καθώς απαιτεί να περιπλανηθούν ανάμεσα σε ένα μεγάλο όγκο πληροφορίας.

Ο μηχανισμός της επιλεκτικής διασποράς πληροφορίας (Selective Dissemination of Information – SDI) [1, 2, 3, 4] βοηθάει τους χρήστες να αντιμετωπίσουν το συγκεκριμένο πρόβλημα. Σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας, οι χρήστες αποστέλλουν προφίλ τα οποία περιέχουν επερωτήσεις που αναπαριστούν τα ενδιαφέροντά τους. Το σύστημα συλλέγει συνεχώς νέα κείμενα από τις πηγές πληροφορίας, τα φιλτράρει σε σχέση με τα προφίλ των χρηστών και παραδίδει τη σχετική πληροφορία στους αντίστοιχους χρήστες. Το βασικό πλεονέκτημα των συγκεκριμένων συστημάτων είναι η λειτουργικότητα και διευκόλυνση που παρέχουν στους χρήστες, καθώς αυτοί περιμένουν παθητικά να λάβουν την πληροφορία που τους ενδιαφέρει χωρίς να δαπανούν χρόνο να την εντοπίσουν.

Ο μηχανισμός της επιλεκτικής διασποράς πληροφορίας εμφανίζεται τόσο σαν αυτόνομη υπηρεσία όσο και σαν υπηρεσία ενσωματωμένη σε μεγαλύτερα συστήματα. Γενικότερα πολλές υπηρεσίες που παρουσιάζονται στον παγκόσμιο ιστό μπορούν να χρησιμοποιήσουν έναν τέτοιου είδος μηχανισμό, προσαρμοσμένο βέβαια στη μορφή και στις ανάγκες τους.

Κλασσικό παράδειγμα τέτοιων υπηρεσιών είναι οι μηχανές αναζήτησης. Διατηρώντας τα ενδιαφέροντα του χρήστη, τον ενημερώνουν για κάθε σχετική με αυτά πληροφορία που παρουσιάζεται στις πηγές τους. Αντίστοιχα, πολλές ψηφιακές βιβλιοθήκες διαθέτουν μία υπηρεσία επιλεκτικής διασποράς πληροφορίας, καθιστώντας το χρήστη συνεχώς ενήμερο για τα νέα άρθρα, τη δουλειά κάποιου συγκεκριμένου συγγραφέα ή τις νέες εκδόσεις, με βάση πάντα το προφίλ του. Στις περισσότερες περιπτώσεις ο χρήστης αποστέλλει τις ανάγκες του σε πληροφορία

(προφίλ) και λαμβάνει μέσω του ηλεκτρονικού ταχυδρομείου την αντίστοιχη πληροφορία, τη στιγμή που αυτή δημοσιεύεται. Παράλληλα, συστήματα που ασχολούνται με το ηλεκτρονικό εμπόριο διατηρούν το προφίλ του χρήστη – καταναλωτή και τον ενημερώνουν αυτόματα όταν κάποιο προϊόν, που φαίνεται να τον ικανοποιεί, παρουσιάζεται στο εμπόριο. Τέλος, διάφορα portals παρέχουν τη συγκεκριμένη υπηρεσία στους χρήστες τους, θέτοντάς τα ιδιαίτερα ανταγωνιστικά στον παγκόσμιο ιστό.

Ένας από τους σημαντικούς στόχους που πρέπει να υλοποιούν όλα τα συστήματα επιλεκτικής διασποράς πληροφορίας είναι η ακρίβεια της πληροφορίας που παραδίδουν στους χρήστες. Για να αντιμετωπίζουν το πρόβλημα της υπερφόρτωσης πληροφορίας, τα συστήματα αυτά οφείλουν να αποστέλλουν στους χρήστες κείμενα, τα οποία πραγματικά ικανοποιούν τα ενδιαφέροντά τους.

Ωστόσο, τα περισσότερα συστήματα διασποράς πληροφορίας στηρίζονται στη λεκτική αναζήτηση. Πιο συγκεκριμένα, εκφράζουν τα κείμενα και τα προφίλ των χρηστών ως σύνολα λέξεων και ελέγχουν την ταυτοσημότητα των λέξεων ανάμεσα στα δύο αυτά σύνολα. Εάν τα δύο σύνολα περιέχουν πολλές όμοιες λεκτικά λέξεις, θεωρούνται «σχετικά», οπότε και το κείμενο αποστέλλεται στο χρήστη. Η συγκεκριμένη προσέγγιση παρουσιάζει δύο βασικά προβλήματα. Αρχικά, μπορεί κάποιος όρος να παρουσιάζεται τόσο σε ένα κείμενο, όσο και σε ένα προφίλ, αλλά με εντελώς διαφορετική σημασία, με αποτέλεσμα ο χρήστης να λαμβάνει κείμενα τα οποία τελικά δεν εκφράζουν τα ενδιαφέροντά του. Από την άλλη πλευρά, κείμενα τα οποία περιέχουν όρους – συνώνυμα των όρων του προφίλ του χρήστη, δεν αποστέλλονται ποτέ στο χρήστη. Τα δύο παραπάνω γεγονότα μειώνουν την ποιότητα της υπηρεσίας που προσφέρεται από τα συγκεκριμένα συστήματα.

Συχνά οι χρήστες ενός συστήματος επιλεκτικής διασποράς πληροφορίας χρησιμοποιούν πολλούς διαφορετικούς όρους για να δηλώσουν την ίδια έννοια, όρους που χαρακτηρίζονται ως συνώνυμα. Παράλληλα, όταν κάποιος χρήστης αναζητά πληροφορία για κάποιον όρο, σίγουρα τον ενδιαφέρει και η πληροφορία που αναφέρεται σε όρους ειδικότερους από αυτόν. Για παράδειγμα, ο χρήστης που ενδιαφέρεται για το «Μηχανοκίνητο όχημα» ενδιαφέρεται και για κείμενα που ασχολούνται με το «Αυτοκίνητο». Άρα η απόδοση ενός συστήματος επιλεκτικής διασποράς πληροφορίας αυξάνεται όταν αυτό περιλαμβάνει μηχανισμούς διαχείρισης των συνωνύμων και των ειδικότερων όρων, μηχανισμούς που ελέγχουν τη σημασιολογική σχετικότητα ανάμεσα στους όρους των προφίλ και των κειμένων.

Επομένως είναι θεμιτό κατά τη διαδικασία του φιλτραρίσματος των κειμένων από τα προφίλ των χρηστών να ληφθεί υπόψη και η σημασιολογία των όρων [5]. Πιο συγκεκριμένα ένα προφίλ θεωρείται σχετικό με ένα κείμενο, όχι μόνο στην περίπτωση που οι όροι του εμφανίζονται στο κείμενο, αλλά και όταν τα συνώνυμα ή τα υπώνυμα (ειδικότεροι όροι) των όρων του παρουσιάζονται στο κείμενο.

Με τη συγκεκριμένη μεθοδολογία αντιμετωπίζεται το δεύτερο πρόβλημα που αναφέρθηκε παραπάνω, καθώς τελικά αποστέλλονται στο χρήστη όλα τα κείμενα που περιέχουν τους όρους του προφίλ του ή όρους σχετικούς εννοιολογικά με αυτούς. Ωστόσο δεν αντιμετωπίζεται το πρώτο πρόβλημα, γεγονός που είναι αναμενόμενο. Εάν ο χρήστης αποστείλει ένα πολύ γενικό προφίλ (προφίλ με λίγους όρους), το σύστημα δεν είναι σε θέση να προσδιορίσει την έννοια των όρων του προφίλ του χρήστη. Προσθέτοντας ωστόσο ο χρήστης περισσότερους όρους στο προφίλ του, περιορίζει τα κείμενα που περιέχουν όλους αυτούς τους όρους ή τους σχετικούς τους. Οπότε σε αυτήν την περίπτωση μειώνεται η πιθανότητα να λάβει ο χρήστης κείμενα τα οποία δεν ανταποκρίνονται στα ενδιαφέροντά του.

Καταλήγουμε λοιπόν στο συμπέρασμα, ότι λαμβάνοντας υπόψη, πέρα από την λεκτική ταυτοσημότητα, και την έννοια των όρων που συμμετέχουν στο φιλτράρισμα, ένα σύστημα επιλεκτικής διασποράς πληροφορίας γίνεται πιο αποτελεσματικό. Εάν οι χρήστες χρησιμοποιούν ένα ικανοποιητικό πλήθος όρων στο προφίλ τους και ειδικότερα μάλιστα σύνθετους όρους, πολλά από τα κείμενα που λαμβάνουν είναι σχετικά με τα ενδιαφέροντά τους. Επομένως αυξάνεται η ακρίβεια του συστήματος.

Ωστόσο, η ανάκτηση της σημασιολογίας των όρων απαιτεί την επικοινωνία του συστήματος με κάποιο θησαυρό ή κάποιο λεξικό όρων, το οποίο αναφέρει την έννοια των όρων και τις σημασιολογικές συσχετίσεις ανάμεσα στους όρους. Η συγκεκριμένη επικοινωνία προκαλεί δυστυχώς κάποια καθυστέρηση, με αποτέλεσμα τα σύστημα από τη μία πλευρά να είναι πιο αποτελεσματικό όσον αφορά την ποιότητα του φιλτραρίσματος, αλλά λιγότερο αποδοτικό όσον αφορά το χρόνο απόκρισης στο χρήστη.

Σκοπός του συστήματος που σχεδιάστηκε και υλοποιήθηκε είναι η παροχή μίας υπηρεσίας επιλεκτικής διασποράς πληροφορίας, η οποία λαμβάνει υπόψη τη σημασιολογία, την έννοια των όρων που χρησιμοποιούνται στα προφίλ και στα κείμενα. Το σύστημα διατηρεί τα προφίλ των χρηστών και καθώς λαμβάνει κείμενα από τις πηγές πληροφορίας, τα συγκρίνει με αυτά τα προφίλ. Τέλος αποστέλλει στους

χρήστες όλα τα κείμενα που περιέχουν συνώνυμα ή υπώνυμα (ειδικότερους όρους) των όρων των προφίλ. Με βάση τις παραπάνω παρατηρήσεις το σύστημα παρουσιάζει ακρίβεια όσον αφορά τα κείμενα που επιστρέφονται στο χρήστη, γεγονός που αποτελεί και τη βασική συνεισφορά του συστήματος στο χώρο ανάκτησης πληροφορίας.

1.2 Συστήματα επιλεκτικής διασποράς πληροφορίας

Ένα πολύ απλό είδος συστήματος επιλεκτικής διασποράς πληροφορίας είναι οι λίστες ηλεκτρονικού ταχυδρομείου (mailing lists) [6]. Οι χρήστες εγγράφονται σε λίστες που ασχολούνται με αντικείμενα που τους ενδιαφέρουν και λαμβάνουν σχετικά μηνύματα μέσω e – mail. Βασικό μειονέκτημα της συγκεκριμένης υπηρεσίας είναι ότι τα ενδιαφέροντα ενός χρήστη μπορεί να μην ταιριάζουν απόλυτα με το αντικείμενο μίας λίστας, με αποτέλεσμα ο χρήστης να λαμβάνει μηνύματα που τελικά να μην τον ενδιαφέρουν.

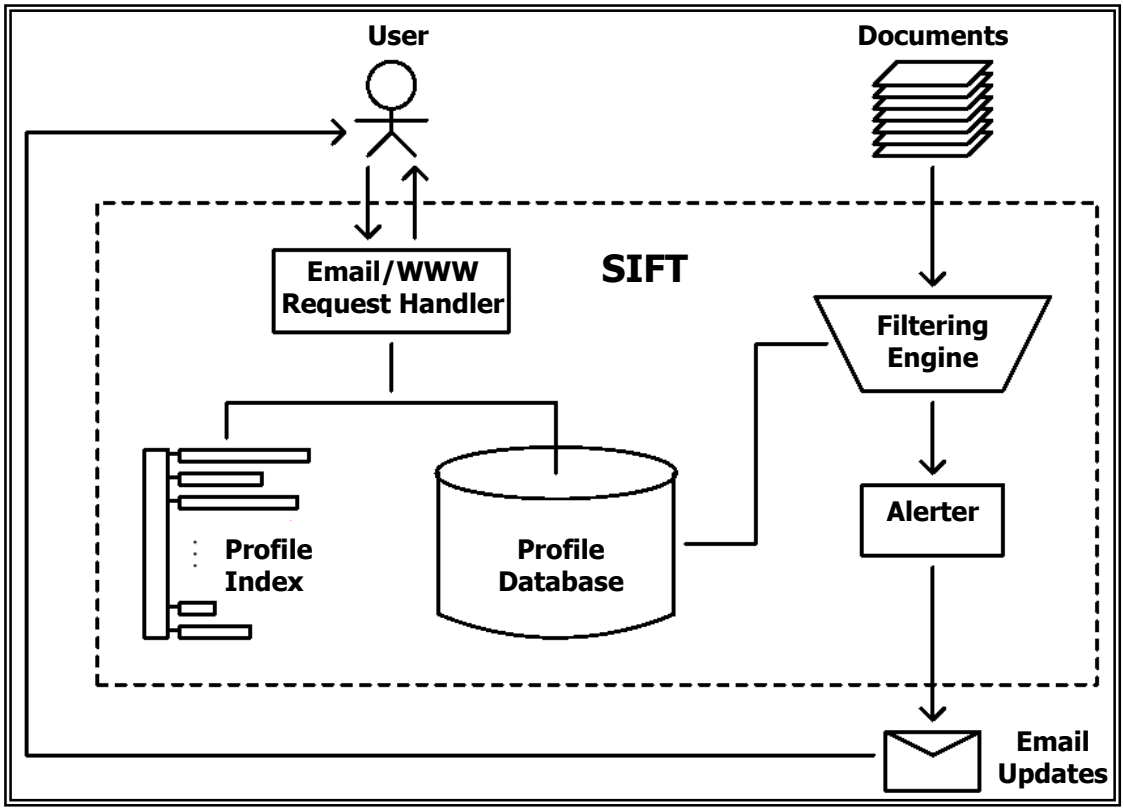
Ένα πειραματικό σύστημα διασποράς πληροφορίας υλοποιήθηκε από το MIT και καλείται Boston Community Information System [7]. Το σύστημα αυτό στέλνει τη νέα πληροφορία μέσω ενός ραδιοφωνικού καναλιού σε όλους τους χρήστες (broadcasting). Στη συνέχεια οι χρήστες, οι οποίοι έχουν θέσει τα προφίλ τους, φιλτράρουν τη συγκεκριμένη πληροφορία τοπικά. Αυτή η υπηρεσία είναι αρκετά καλή στην περίπτωση που το broadcasting είναι φτηνό, αλλά δεν αρμόζει σε επικοινωνίες σημείου – προς – σημείο όπως είναι το Internet. Ένα άλλο μοντέλο το οποίο έχει προταθεί για τη διασπορά πληροφορίας σε ευρύ μέσο είναι οι Broadcast disks [8, 9, 10, 11, 12]. Όπως προαναφέρθηκε, όμως, το ενδιαφέρον μας επικεντρώνεται σε επικοινωνίες σημείου – προς – σημείο.

Το XFilter [13] αποτελεί ένα άλλο σύστημα φιλτραρίσματος κειμένων, το οποίο παρέχει αποτελεσματικό ταίριασμα ανάμεσα σε κείμενα XML και σε προφίλ χρηστών. Τα ενδιαφέροντα των χρηστών αναπαριστώνται σαν επερωτήσεις με τη χρήση της γλώσσας XPath [14]. Ενώ τα κλασσικά συστήματα διασποράς πληροφορίας αναφέρονται κυρίως σε απλά κείμενα, το XFilter μπορεί να λειτουργήσει σε οποιαδήποτε εφαρμογή που τα δεδομένα είναι εκφρασμένα σε XML. Επίσης, το συγκεκριμένο σύστημα παρέχει πιο ακριβές φιλτράρισμα των κειμένων, καθώς

εκμεταλλεύεται την πληροφορία του σχήματος των XML κειμένων, κάτι που δεν επιτυγχάνεται με τη χρήση απλών λέξεων κλειδιών.

Η αρχιτεκτονική Information Bus [15] για την κατασκευή κατανεμημένων συστημάτων προτείνει ένα μοντέλο publish – subscribe [16, 17, 18], στο οποίο οι εκδότες (publishers) στέλνουν δεδομένα στους συνδρομητές (subscribers). Συστήματα που έχουν χρησιμοποιήσει τη συγκεκριμένη αρχιτεκτονική επικεντρώνουν το ενδιαφέρον τους στη διασπορά της πληροφορίας με διάφορα πρωτόκολλα επικοινωνίας, θέμα που δεν αποτέλεσε αντικείμενο της συγκεκριμένης εργασίας.

Ένα από τα πιο αντιπροσωπευτικά συστήματα επιλεκτικής διασποράς πληροφορίας είναι το SIFT (Stanford Information Filtering Tool) [19, 20], το οποίο παρουσιάζεται και στο Σχήμα 1. Το SIFT συλλέγει USENET Netnews και άρθρα από διάφορες λίστες ηλεκτρονικού ταχυδρομείου. Ο χρήστης εγγράφεται στο σύστημα είτε στέλνοντας email με το προφίλ του είτε συμπληρώνοντας μία φόρμα μέσω ενός Web Browser. Όλα τα προφίλ των χρηστών αποθηκεύονται σε μία βάση δεδομένων. Καθώς ο SIFT server λαμβάνει νέα κείμενα, η μηχανή φιλτραρίσματος του συστήματος τα επεξεργάζεται με βάση τα αποθηκευμένα προφίλ. Τέλος, τα κείμενα που ταιριάζουν τα προφίλ αποστέλλονται μέσω e-mail στους αντίστοιχους χρήστες.



Σχήμα 1. Σύστημα SIFT

Τα μοντέλα που χρησιμοποιούνται για την αναπαράσταση των προφίλ και των κειμένων στο SIFT είναι το Boolean και το Vector Space μοντέλο, δύο τυποποιημένες γλώσσες στο χώρο της ανάκτησης πληροφορίας. Στην περίπτωση του Vector Space μοντέλου χρησιμοποιείται κάποια μετρική ομοιότητας (dot product) και ένα κατώφλι σχετικότητας (relevance threshold) για το ταίριασμα των προφίλ με τα κείμενα. Τα προφίλ του Boolean μοντέλου ορίζουν λέξεις που πρέπει να περιλαμβάνονται στα κείμενα και λέξεις που δεν πρέπει να υπάρχουν στα κείμενα. Για τα συγκεκριμένα προφίλ υποστηρίζονται οι λογικές πράξεις της σύζευξης και της διάζευξης, καθώς και η άρνηση.

Όπως προαναφέρθηκε, οι χρήστες αποστέλλουν τα προφίλ τους στο SIFT είτε μέσω e – mail είτε μέσω μίας φόρμας στο Web. Πιο συγκεκριμένα, ο κάθε χρήστης στέλνει ένα προφίλ ανά αντικείμενο του ενδιαφέροντός του, το οποίο περιλαμβάνει την επερώτηση (εκφρασμένη με τη μορφή του μοντέλου Boolean είτε με τη μορφή του μοντέλου Vector Space), τη συχνότητα που επιθυμεί να τον ενημερώνει το σύστημα, το μέγεθος της πληροφορίας που επιθυμεί να δέχεται κάθε φορά, τη διάρκεια εγκυρότητας του προφίλ και τέλος το e – mail του με το προσδιοριστικό του προφίλ.

Η έρευνα στο SIFT έχει επικεντρωθεί στη δημιουργία μηχανισμών δημιουργίας ευρετηρίων των προφίλ, με σκοπό τη μείωση του χρόνου της διαδικασίας σύγκρισης των προφίλ και των κειμένων. Υπάρχουν δύο βασικοί λόγοι για τους οποίους επιλέχτηκε να γίνεται ευρετηρίαση στα προφίλ και όχι στα κείμενα, όπως συμβαίνει στα συστήματα ανάκτησης πληροφορίας. Αρχικά το σύνολο των προφίλ είναι πολύ μεγάλο και σχετικά στατικό. Από την άλλη πλευρά το σύνολο των κειμένων δεν είναι στατικό, καθώς τακτικά παρουσιάζονται νέα κείμενα στο σύστημα. Η τεχνική ευρετηρίασης των προφίλ που υλοποιήθηκε στο SIFT επιλέγει τους «σημαντικούς όρους» κάθε προφίλ και εισάγει μόνο αυτούς στο ευρετήριο.

Επιπλέον, αρκετή μελέτη στο SIFT έχει πραγματοποιηθεί σχετικά με την κατανεμημένη διασπορά πληροφορίας (ύπαρξη πολλών servers) και την απομάκρυνση των διπλοτύπων των κειμένων, θέματα που δεν μας απασχόλησαν στην παρούσα εργασία. Τέλος, αν και το SIFT έχει ως σύστημα υψηλή απόδοση και ακρίβεια, δεν λαμβάνει καθόλου υπόψη του τη σημασιολογία των όρων των κειμένων και των προφίλ.

Όλα τα παραπάνω συστήματα διαφοροποιούνται ως προς τα κλασσικά συστήματα ανάκτησης πληροφορίας. Χαρακτηρίζονται ως τεχνολογίες "push" [21, 22] διότι δεν απαιτούν από τους χρήστες να ζητούν ρητά την πληροφορία (όπως γίνεται στα συστήματα "pull"). Τα δεδομένα αποστέλλονται στο χρήστη όταν εμφανιστούν και χωρίς οι χρήστες να γνωρίζουν την ύπαρξή τους. Επιπλέον δεν απαιτούν από τους χρήστες να δαπανήσουν χρόνο για την αναζήτηση της πληροφορίας που επιθυμούν. Το κόστος βέβαια των συγκεκριμένων συστημάτων είναι η υπερφόρτωση του παροχέα πληροφορίας, καθώς ο έλεγχος μεταφέρεται από τον κάθε χρήστη σε αυτόν.

1.3 Οργάνωση της παρούσας εργασίας

Στη συνέχεια παρουσιάζεται η δομή ενός συστήματος επιλεκτικής διασποράς πληροφορίας, καθώς και τα μοντέλα αναπαράστασης των προφίλ των χρηστών και των κειμένων, όπως αυτά ορίζονται σε ένα τέτοιο σύστημα (Κεφάλαιο ΙΙ). Παράλληλα στο Κεφάλαιο ΙΙ, περιγράφονται οι πιο δημοφιλείς μέθοδοι ευρετηρίασης των προφίλ για τα προαναφερόμενα μοντέλα.

Στο Κεφάλαιο ΙΙΙ, παρουσιάζεται ο αλγόριθμος που χρησιμοποιήθηκε στο σύστημα, καθώς και η μεθοδολογία σύμφωνα με την οποία εισάγεται η σημασιολογία. Το Κεφάλαιο IV περιλαμβάνει μία περιγραφή του συστατικών μερών του συστήματος, ενώ το Κεφάλαιο V περιγράφει την υλοποίησή του.

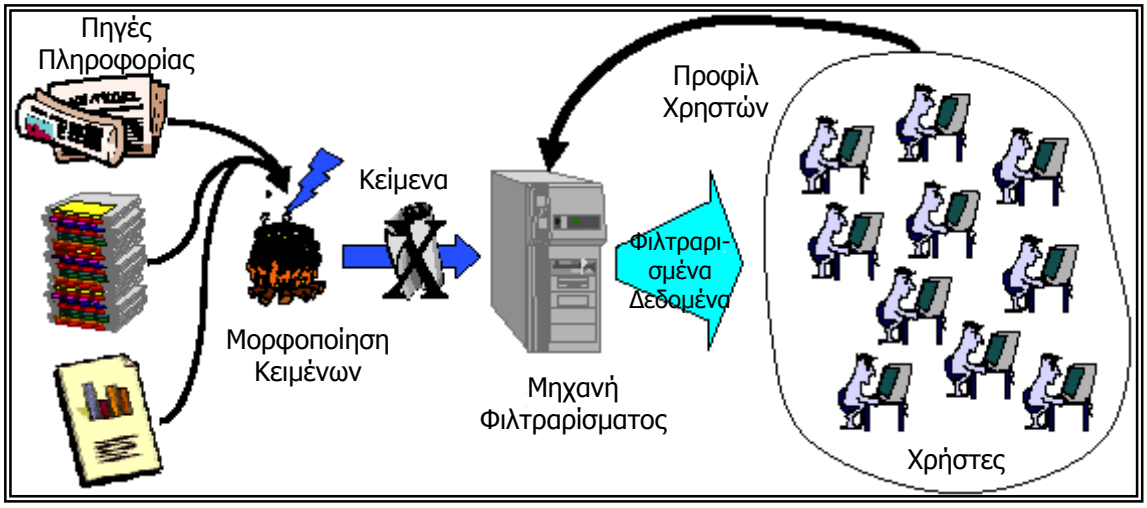
Τέλος, παρουσιάζονται κάποια πειραματικά αποτελέσματα (Κεφάλαιο VI), συμπεράσματα και μελλοντικές εξελίξεις της συγκεκριμένης υπηρεσίας.

Κεφάλαιο ΙΙ

Σύστημα επιλεκτικής διασποράς πληροφορίας

2.1 Δομή συστήματος επιλεκτικής διασποράς πληροφορίας

Η ανατομία ενός συστήματος επιλεκτικής διασποράς πληροφορίας φαίνεται στο Σχήμα 2. Το σύστημα δέχεται κείμενα από κάποιες πηγές πληροφορίας, τα οποία μορφοποιούνται και αποστέλλονται στη μηχανή φιλτραρίσματος. Παράλληλα οι χρήστες αποστέλλουν τα προφίλ τους στην ίδια μηχανή. Ως βασικό συστατικό του συστήματος παρουσιάζεται η μηχανή φιλτραρίσματος, η οποία είναι υπεύθυνη για τη σύγκριση των προφίλ των χρηστών και των κειμένων. Μετά τη διαδικασία της σύγκρισης, η μηχανή αποστέλλει τα σχετικά κείμενα στους αντίστοιχους χρήστες.



Σχήμα 2. Ανατομία συστήματος επιλεκτικής διασποράς πληροφορίας

Συμπερασματικά, τα δομικά μέρη ενός συστήματος επιλεκτικής διασποράς πληροφορίας είναι τα εξής:

- Η μηχανή φιλτραρίσματος, η οποία υλοποιεί τη διαδικασία του «ταιριάσματος» ανάμεσα στα προφίλ των χρηστών και στα κείμενα, και την αποστολή των σχετικών κειμένων στους χρήστες.
- Ο διαχειριστής των προφίλ των χρηστών, ο οποίος αποδέχεται τις αιτήσεις των χρηστών για πληροφορία.

- Ο διαχειριστής των κειμένων, ο οποίος είναι υπεύθυνος για την κατάλληλη μορφοποίηση των κειμένων.

Ο σχεδιασμός και η υλοποίηση ενός συστήματος επιλεκτικής διασποράς πληροφορίας απαιτεί τον τυπικό ορισμό των κείμενων και των προφίλ του συστήματος, καθώς και μία κοινή αναπαράστασή τους. Παράλληλα για την υλοποίηση της σύγκρισης των κειμένων και των προφίλ απαιτείται η δημιουργία δομών και η εφαρμογή ενός αλγορίθμου ταιριάσματος.

Στο χώρο της ανάκτησης πληροφορίας έχει εμφανιστεί μία σειρά μοντέλων για την αναπαράσταση των κειμένων και των προφίλ των χρηστών. Ως επικρατέστερα φαίνονται το Boolean μοντέλο και το Vector Space μοντέλο. Στις επόμενες ενότητες παρουσιάζονται τα δύο αυτά μοντέλα, καθώς και οι δομές που χρησιμοποιούνται για τη σύγκριση σε καθένα από τα δύο μοντέλα και ο αλγόριθμος ταιριάσματος.

2.2 Μοντέλα αναπαράστασης προφίλ και κειμένων σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας

2.2.1 Όρος κειμένου ή προφίλ

Θεωρούμε ένα πεπερασμένο αλφάβητο Σ. Μία λέξη είναι μία πεπερασμένη, μη κενή σειρά από γράμματα του αλφαβήτου Σ. Έστω V ένα λεξιλόγιο, το οποίο αποτελεί ένα πεπερασμένο σύνολο από λέξεις. Ένας όρος ορίζεται ως μία λέξη του λεξιλογίου V ή ως μία φράση, δηλαδή ένας συνδυασμός λέξεων του λεξιλογίου V.

Ο κάθε όρος του λεξιλογίου έχει ένα βαθμό, ο οποίος δηλώνει τη συχνότητα εμφάνισης της όρου στη δεδομένη συλλογή των κειμένων. Επομένως ο όρος που εμφανίζεται πιο συχνά ή πιο σπάνια στα κείμενα έχει τον υψηλότερο ή το χαμηλότερο βαθμό αντίστοιχα. Πιο συγκεκριμένα ο βαθμός ενός όρου t ορίζεται ως το πλήθος των κειμένων που περιέχουν τον όρο προς το συνολικό πλήθος των κειμένων, δηλαδή:

$$\text{Βαθμός όρου t} = \frac{\text{Πλήθος κειμένων που περιέχουν τον όρο}}{\text{Πλήθος όλων των κειμένων}}$$

2.2.2 Κείμενα και προφίλ χρηστών

Σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας, ο χρήστης στέλνει το προφίλ του για να δηλώσει τις ανάγκες του ως προς την πληροφορία που επιθυμεί. Στη συνέχεια το σύστημα συλλέγει τα νέα κείμενα από τις πηγές πληροφορίας, τα φιλτράρει με βάσει τα προφίλ του χρήστη και παραδίδει τα σχετικά κείμενα σε αυτόν.

Το κάθε κείμενο περιγράφεται με το σύνολο των μεταδεδομένων του. Τα μεταδεδομένα που επιλέχθηκαν είναι ο τίτλος, οι συγγραφείς και η ηλεκτρονική διεύθυνση (URL) του κειμένου, καθώς και κάποιες λέξεις – κλειδιά (keywords), οι οποίες περιγράφουν το περιεχόμενο του κειμένου.

Ο κάθε χρήστης αποστέλλει μόνο ένα προφίλ στο σύστημα, στο οποίο εκφράζει τα ενδιαφέροντά του. Καθώς τα ενδιαφέροντα ενός χρήστη ποικίλουν, το προφίλ του χωρίζεται σε περισσότερες κατηγορίες ενδιαφερόντων, δηλαδή σε περισσότερα υπο – προφίλ. Αυτά τα υπο – προφίλ συνδέονται μεταξύ τους με το διάζευξη (OR), ενώ οι όροι μέσα στο κάθε υπο – προφίλ συνδέονται μεταξύ τους με σύζευξη (AND). Για παράδειγμα, το προφίλ

Information dissemination

Databases, Networks

Information retrieval, clustering

ερμηνεύεται ως εξής: ο χρήστης ενδιαφέρεται είτε για κείμενα που αφορούν το information dissemination, είτε για κείμενα που αφορούν τα databases και networks, είτε για κείμενα που αφορούν τα information retrieval και clustering.

Το προφίλ του χρήστη μπορεί να αντιμετωπιστεί ως μία συνεχώς εκτελούμενη επερώτηση (continuous query) απέναντι στο σύστημα. Δηλαδή τα προφίλ των χρηστών εκτελούνται συνεχώς και το σύστημα προωθεί κείμενα στους χρήστες όσο υπάρχουν έγκυρα προφίλ τα οποία ταιριάζουν με αυτά τα κείμενα. Για αυτό το λόγο, στη συνέχεια του κειμένου οι όροι προφίλ και επερώτηση χρησιμοποιούνται ως συνώνυμα.

2.2.3 Boolean μοντέλο

Σύμφωνα με το Boolean μοντέλο [23, 24, 25], ένα κείμενο αναπαρίσταται σαν ένα σύνολο όρων, το οποίο περιλαμβάνει μόνο τις λέξεις – κλειδιά του κειμένου,

καθώς ο τίτλος, οι συγγραφείς και η ηλεκτρονική διεύθυνση του κειμένου δεν περιλαμβάνουν χρήσιμη πληροφορία για αυτό. Επομένως ένα κείμενο D που περιέχει n όρους συμβολίζεται στο Boolean μοντέλο ως:

$$D = (t_1, t_2, \ldots, t_n),$$

όπου t_i, $1 \leq i \leq n$, είναι ένας όρος του λεξιλογίου V.

Ένα προφίλ ή μία επερώτηση στο Boolean μοντέλο αποτελεί μία λογική έκφραση, η οποία αποτελείται από κατηγορήματα που ενώνονται με τους λογικούς τελεστές AND, OR, AND NOT και με τους τελεστές εγγύτητας (proximity operators). Ανάλογα με την ύπαρξη τελεστών εγγύτητας, διακρίνουμε τις επερωτήσεις σε δύο κατηγορίες, στις «επερωτήσεις χωρίς τελεστές εγγύτητας» (proximity – free queries) και στις «επερωτήσεις με τελεστές εγγύτητας» (proximity queries).

Μία «επερώτηση χωρίς τελεστές εγγύτητας» είναι μία επερώτηση, η οποία δεν περιέχει τελεστή εγγύτητας, δηλαδή είναι μία έκφραση η οποία λαμβάνει οποιαδήποτε από τις παρακάτω μορφές:

- w, όπου w είναι ένας όρος του λεξιλογίου V (είτε μία ακριβής λέξη είτε μία φράση).
- Q1 AND Q2, όπου Q1, Q2 είναι επερωτήσεις οι οποίες δεν περιέχουν τελεστές εγγύτητας.
- Q1 OR Q2, όπου Q1, Q2 είναι επερωτήσεις οι οποίες δεν περιέχουν τελεστές εγγύτητας.
- Q1 AND NOT Q2, όπου Q1, Q2 είναι επερωτήσεις οι οποίες δεν περιέχουν τελεστές εγγύτητας.
- (Q), όπου Q είναι μία επερώτηση που δεν περιέχει τελεστές εγγύτητας.

Μία επερώτηση χωρίς τελεστή εγγύτητας είναι θετική, όταν δεν περιέχει τον τελεστή της άρνησης (AND NOT).

Μία «επερώτηση με τελεστές εγγύτητας» [26, 27] είναι μία επερώτηση, η οποία περιέχει τελεστή εγγύτητας, δηλαδή είναι μία έκφραση η οποία έχει τη μορφή:

- Q1 $<_{[l,\ u]}$ Q2, όπου τα Q1, Q2 είναι θετικές επερωτήσεις χωρίς τελεστές εγγύτητας ή επερωτήσεις με τελεστές εγγύτητας, το [l, u] είναι ένα διάστημα και $l, u \in N$.

Ο τελεστής εγγύτητας $<_{[l,\ u]}$ ορίζει ότι ο πρώτος τελεστέος προηγείται του δεύτερου και ότι υπάρχουν μεταξύ τους l έως u λέξεις. Δηλαδή, η έκφραση «α $<_{[l,\ u]}$ β» σημαίνει ότι ο όρος α είναι πριν τον όρο β και χωρίζεται από το β με τουλάχιστον l και το πολύ u λέξεις.

Στον παραπάνω ορισμό παρατηρείται ότι οι επερωτήσεις με τελεστή εγγύτητας δεν περιέχουν τον τελεστή της άρνησης. Αυτό είναι αναμενόμενο, καθώς δεν είναι λογικό να οριστεί η απόσταση ή η σειρά δύο όρων από τη στιγμή που ο ένας από τους δύο δεν επιθυμείται να περιέχεται στη αναπαράσταση του κειμένου.

Τα παραδοσιακά συστήματα ανάκτησης πληροφορίας χρησιμοποιούν τους τελεστές εγγύτητας kW και kN, όπου το k είναι ένας φυσικός αριθμός. Η έκφραση «α kW β» δηλώνει ότι ο όρος α προηγείται του όρου β και ότι χωρίζεται από το β με το πολύ k λέξεις. Με τη χρήση του τελεστή $<_{[l, u]}$, αυτό μπορεί να εκφραστεί ως «α $<_{[0, k]}$ β». Από την άλλη πλευρά, ο τελεστής kN χρησιμοποιείται για να δηλώσει την απόσταση δύο όρων (το πολύ k λέξεις υπάρχουν ανάμεσα στους δύο όρους), αλλά δεν εκφράζει κάποια σειρά. Στην περίπτωσή μας, η έκφραση «α kN β» ισοδυναμεί με την έκφραση «(α $<_{[0, k]}$ β) OR (β $<_{[0, k]}$ α)». Πέρα από αυτές τις λειτουργίες, ο τελεστής εγγύτητας $<_{[l, u]}$ μπορεί να εκφράσει απλούς περιορισμούς σειράς ανάμεσα σε λέξεις με τη χρήση του $<_{[0, \infty]}$. Επομένως ο τελεστής $<_{[l, u]}$ από τη μία πλευρά καλύπτει τις λειτουργίες των παραδοσιακών τελεστών και από την άλλη έχει κάποια επιπλέον λειτουργικότητα.

2.2.4 Vector Space μοντέλο

Στο Vector Space μοντέλο [28, 29], ένα κείμενο αναπαρίσταται σαν ένα διάνυσμα ζευγών (όρος, βάρος), καθένα από τα οποία αντιστοιχεί σε μία από τις λέξεις – κλειδιά του κειμένου. Επομένως ένα κείμενο D με n όρους αναπαρίσταται σαν ένα διάνυσμα n διαστάσεων, δηλαδή:

$$D = <(x_1, w_1), \ldots, (x_n, w_n)>$$

όπου x_i, $1 \leq i \leq n$, είναι ένας όρος του λεξιλογίου V και $w_i > 0$, $1 \leq i \leq n$, είναι το βάρος της όρου t_i.

Για τον υπολογισμό της διανυσματικής αναπαράστασης ενός κειμένου, ακολουθείται η παρακάτω διαδικασία: Αρχικά εντοπίζονται οι λέξεις – κλειδιά του κειμένου. Από αυτές τις λέξεις διαγράφονται οι stop – words, δηλαδή οι λέξεις που έχουν μεγάλη συχνότητα, αλλά μικρή σημασία όσον αφορά το περιεχόμενο του κειμένου. Σε καθένα από τους όρους που παραμένουν αντιστοιχίζεται ένα βάρος που δηλώνει πόσο στατιστικά σημαντικός είναι αυτός ο όρος. Τέλος, πραγματοποιείται κανονικοποίηση του διανύσματος, με σκοπό το χειρισμό των κειμένων που έχουν

διαφορετικά μήκη. Η κανονικοποίηση πραγματοποιείται με τη διαίρεση του βάρους του κάθε όρου με το μήκος του διανύσματος.

Το βάρος ενός όρου μπορεί να υπολογιστεί εάν πολλαπλασιάσουμε τη συχνότητα του όρου (term frequency – tf) με την αντίστροφη συχνότητα του κειμένου (inverse document frequency – idf) [30, 31]. Ο όρος tf είναι ίσος με τη συχνότητα που εμφανίζεται ο όρος στο κείμενο. Ο όρος idf εκφράζει τη μοναδικότητα του όρου, καθώς αναφέρεται στη σχέση του όρου ως προς τη συλλογή των κειμένων. Επομένως κάποιος όρος που εμφανίζεται σπάνια στα κείμενα, έχει μεγάλη τιμή idf. Αντίθετα κάποιος όρος που παρουσιάζεται συχνά σε ένα μεγάλο πλήθος κειμένων έχει μικρή τιμή idf. Όπως παρατηρούμε, η συχνότητα idf υπολογίζεται με βάση κάποια συλλογή από κείμενα. Όμως σε ένα σύστημα διασποράς πληροφορίας, δεν υπάρχει κάποια σταθερή συλλογή με κείμενα. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η χρήση μίας συλλογής αναφοράς, η οποία αποτελείται από τα κείμενα που έχουν επεξεργαστεί πιο πρόσφατα. Ωστόσο, υπάρχει η περίπτωση να παρουσιαστεί κάποιο κείμενο που περιέχει κάποιον όρο για τον οποίο δεν έχει υπολογιστεί ο όρος idf. Τότε μπορούμε να αντιστοιχίσουμε στον όρο μία πολύ υψηλή τιμή idf, όπως για παράδειγμα την πιο μεγάλη idf τιμή που έχει ήδη εμφανιστεί.

Τυπικά, το βάρος ενός όρου t ενός κειμένου d δίνεται από τη σχέση:

$$w_{t,d} = tf_{t,d} * idf_t = tf_{t,d} * \log_2\left(\frac{N}{df_t}\right)$$

όπου

- $w_{t,d}$ είναι το βάρος του όρου t στο κείμενο d
- $tf_{t,d}$ είναι η συχνότητα του όρου t στο κείμενο d (term frequency)
- idf_t είναι η σχέση (relevance) του όρου t ως προς τη συλλογή των κειμένων (inverse document frequency)
- N είναι το συνολικό πλήθος των κειμένων στη συλλογή
- df_t είναι το πλήθος των κειμένων που περιέχουν τον όρο t (document frequency)

Αντίστοιχα, μία επερώτηση Q με n όρους αναπαρίσταται στο Vector Space μοντέλο σαν ένα διάνυσμα ζευγών (όρος, βάρος) n-διαστάσεων, δηλαδή

$$Q = <(y_1, w_1), \ldots, (y_n, w_n)>$$

όπου t_i, $1 \leq i \leq n$, είναι ένας όρος του λεξιλογίου V και $w_i > 0$ είναι το βάρος της όρου t_i. Το βάρος w_i ενός όρου υπολογίζεται ως το γινόμενο των πλήθους των εμφανίσεων

του i όρου στην επερώτηση με τον παράγοντα idf του όρου, δηλαδή το βάρος ενός όρου t ενός προφίλ δίνεται από τη σχέση:

$$w_t = A * idf_t = A * \log_2\left(\frac{N}{df_t}\right)$$

όπου

- w_t είναι το βάρος του όρου t στο προφίλ
- A είναι το πλήθος των εμφανίσεων του όρου στο προφίλ
- idf_t είναι η σχέση (relevance) του όρου t ως προς τη συλλογή των κειμένων (inverse document frequency)
- N είναι το συνολικό πλήθος των κειμένων στη συλλογή
- df_t είναι το πλήθος των κειμένων που περιέχουν τον όρο t (document frequency)

Αν και ο παραπάνω τύπος για το βάρος ενός όρου κειμένου παρουσιάζεται ευρέως στη βιβλιογραφία, πρέπει να τονιστεί ότι ειδικά στην περίπτωσή μας ισχύει A = 1. Δηλαδή ο όρος του κάθε προφίλ εμφανίζεται μία φορά στο προφίλ.

Ο βαθμός ομοιότητας ανάμεσα σε ένα κείμενο και μία επερώτηση στηρίζεται στα βάρη των αντίστοιχων όρων. Για να προσδιορίσουμε αυτό το βαθμό ομοιότητας χρησιμοποιούμε μία μετρική που καλείται cosine similarity measure [32]. Για ένα κείμενο $D = <(x_1,w_1), \ldots, (x_n,w_n)>$ και μία επερώτηση $Q = <(y_1,z_1), \ldots, (y_n,z_n)>$, το cosine similarity measure ορίζεται ως εξής:

$$sim(D,Q) = \frac{D * Q}{\|D\| * \|Q\|} = \frac{\sum_{i=1}^{n} w_i z_i}{\sqrt{\sum_{i=1}^{n} w_i^2 \sum_{i=1}^{n} z_i^2}}$$

όπου $\|D\|$ είναι η νόρμα του διανύσματος D και $\|Q\|$ η νόρμα του διανύσματος Q. Όπως φαίνεται και από τον ορισμό, $sim(D, Q) \in (0, 1)$. Εάν τα δύο διανύσματα D και Q είναι κανονικοποιημένα ως προς το μήκος τους, τότε το cosine similarity measure απλοποιείται ως εξής:

$$sim(D,Q) = D * Q = \sum_{i=1}^{n} w_i z_i$$

Για να προσδιορίσουμε εάν κάποιο κείμενο ικανοποιεί μία επερώτηση, ο χρήστης ορίζει ένα κατώφλι ομοιότητας (relevance threshold), πάνω από το οποίο τα κείμενα θεωρούνται σχετικά με το προφίλ του. Επομένως για μία δεδομένη

επερώτηση Q και ένα δεδομένο κατώφλι ομοιότητας θ, ένα κείμενο D είναι σχετικό με την επερώτηση Q εφόσον

$$\mathrm{sim}(D, Q) > \theta$$

2.2.5 Σύγκριση των δύο μοντέλων

Τα μοντέλα Boolean και Vector Space είναι πολύ δημοφιλή. Ωστόσο παρουσιάζουν μία σειρά από διαφορές [33], οι οποίες τα θέτουν ιδιαίτερα ανταγωνιστικά στο χώρο της ανάκτησης πληροφορίας.

Αρχικά και με τις δύο προσεγγίσεις, μπορούν να σχηματιστούν πολύ ευέλικτες και εκφραστικές επερωτήσεις από τους χρήστες. Ωστόσο το Boolean μοντέλο απευθύνεται κυρίως σε ειδικευμένους χρήστες, καθώς δεν είναι διαισθητικά κατανοητό στον απλό χρήστη. Απαιτεί γνώση της Boolean λογικής και εξοικείωση για την επιλογή των σωστών όρων. Για τον ειδικευμένο χρήστη είναι ιδιαίτερα απλό και εύκολο στην κατανόηση και μπορεί να τον οδηγήσει στην πραγματοποίηση επερωτήσεων με ένα πλήθος περιορισμών. Από την άλλη πλευρά, το Vector Space μοντέλο είναι διαισθητικά ελκυστικό και αρκετά κατανοητό στον απλό χρήστη. Είναι πιο κοντά στον τρόπο που σκέφτεται ο άνθρωπος, καθώς λαμβάνει υπόψη του το γεγονός ότι κάποια κείμενα είναι καλύτερα, πιο σχετικά από κάποια άλλα.

Όπως προαναφέρθηκε, το Vector Space μοντέλο χρησιμοποιεί βάρη στους όρους, δηλώνοντας έτσι τη σπουδαιότητα του κάθε όρου. Στο Boolean μοντέλο είναι δύσκολο να χρησιμοποιηθεί αυτή η τεχνική. Η ανυπαρξία βαρών σε όρους των επερωτήσεων έχει σαν αποτέλεσμα να μη μπορούν οι χρήστες να δώσουν κάποια βαρύτητα σε όρους ή προτάσεις στις επερωτήσεις τους. Η ανυπαρξία βαρών σε όρους των κειμένων έχει σαν αποτέλεσμα να λαμβάνονται αυστηρές δυαδικές αποφάσεις ευρετηρίαση (indexing) και να μη μπορεί να γίνει χρήση της γνώσης της συχνότητας ενός όρου σε ένα κείμενο ή της σπάνιας εμφάνισης ενός όρου σε μία συλλογή κειμένων. Όλες αυτές οι λειτουργίες πραγματοποιούνται στο Vector Space μοντέλο, καθιστώντας πιο ακριβή την αναζήτηση.

Ωστόσο, το Vector Space μοντέλο υστερεί όσον αφορά κάποιες υποθέσεις που έχουν ληφθεί. Αρχικά υποθέτει ότι οι όροι είναι ορθογώνιοι, δηλαδή ανεξάρτητοι μεταξύ τους, γεγονός το οποίο δεν ισχύει. Επιπλέον δεν υπάρχει καμία θεωρητική βάση για το σύνολο όρων που έχουν επιλεχθεί στο διάνυσμα. Τα βάρη των όρων

συχνά υπολογίζονται αυθαίρετα και οι μετρικές ομοιότητας λειτουργούν το ίδιο, ανεξάρτητα από το μοντέλο. Όλα αυτά οφείλονται κυρίως στο γεγονός ότι το συγκεκριμένο μοντέλο δε στηρίζεται σε κάποιο θεωρητικό υπόβαθρο. Αντίθετα, το Boolean μοντέλο στηρίζεται στην Boolean Algebra, γεγονός που το καθιστά τυποποιημένη προσέγγιση.

Η ακρίβεια και η παρουσίαση των αποτελεσμάτων υστερούν στο Boolean μοντέλο. Πιο συγκεκριμένα, αυτό το μοντέλο μπορεί εύκολα να οδηγήσει στην παραγωγή μηδενικών ή μεγάλου πλήθους αποτελεσμάτων. Οι συζευκτικές επερωτήσεις (δηλαδή οι επερωτήσεις που περιέχουν τον τελεστή AND) συχνά καταλήγουν σε λίγα ή μηδενικά αποτελέσματα, ενώ οι διαζευκτικές επερωτήσεις (δηλαδή οι επερωτήσεις που περιέχουν τον τελεστή OR) συχνά καταλήγουν σε πάρα πολλά αποτελέσματα. Έτσι το μοντέλο παρέχει ακριβή αποτελέσματα μόνο αν ο χρήστης γνωρίζει το σωστό τρόπο με τον οποίο πρέπει να εκφράσει την επερώτηση. Πρόβλημα στην ακρίβεια των αποτελεσμάτων παρουσιάζεται και στο Vector Space μοντέλο, καθώς είναι δύσκολος ο καθορισμός συνωνύμων και σχέσεων ανάμεσα σε εκφράσεις σε αυτό το μοντέλο. Ο προσδιορισμός φραστικών σχέσεων είναι από την άλλη πλευρά πολύ εύκολος στο Boolean μοντέλο.

Όσον αφορά την παρουσίαση των αποτελεσμάτων, το Boolean μοντέλο παρουσιάζει κάποια προβλήματα. Αρχικά τα αποτελέσματα δεν είναι ταξινομημένα με κάποιο χρήσιμο τρόπο, για παράδειγμα κατά σειρά «βαθμολογίας» (ranking), άρα ο χρήστης είναι αναγκασμένος να ερευνήσει όλη την ανακτημένη πληροφορία. Επίσης δεν πραγματοποιείται σχεδόν κανένας έλεγχος όσον αφορά το μέγεθος των αποτελεσμάτων. Από την άλλη πλευρά, το Vector Space μοντέλο χρησιμοποιεί ranking, οπότε διευκολύνει το χρήστη στην πλοήγηση στα αποτελέσματα. Ωστόσο για την επίτευξη ενός αυστηρού ranking, αυτό το μοντέλο απαιτεί πολλούς όρους, ενώ ένα αντίστοιχης ποιότητας αποτέλεσμα θα μπορούσε να ανακληθεί με τη χρήση δύο ή τριών όρων ενωμένων με AND στο Boolean μοντέλο.

Όσον αφορά την υπολογιστική πολυπλοκότητα των μοντέλων, παρατηρείται ότι και τα δύο είναι απλά και φτηνά να προγραμματιστούν. Μάλιστα το Boolean μοντέλο αποτελεί το πρώτο μοντέλο που υλοποιήθηκε σε υπολογιστικό σύστημα για την ανάκτηση ηλεκτρονικής πληροφορίας.

Τέλος πρέπει να σημειωθεί ότι το Vector Space μοντέλο αποτελεί μία πολύ καλή βάση για την ανάπτυξη ενός πλήθους λειτουργιών σχετικά με την ανάκτηση πληροφορίας όπως είναι η ευρετηρίαση (indexing), η ανάδραση σχετικότητας

(relevance feedback), η κατηγοριοποίηση των κειμένων και η ανάκτηση κατά ομάδες (clustering retrieval).

2.3 Δομές ευρετηρίων για τα προφίλ σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας

Σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας αποθηκεύεται ένας μεγάλος αριθμός από προφίλ, ενώ τα κείμενα ελέγχονται ανεξάρτητα για «ταίριασμα» (matching) με τα προφίλ. Επομένως για την καλύτερη απόδοση τέτοιων συστημάτων, απαιτείται η ύπαρξη δομών ευρετηρίου για τις επερωτήσεις των χρηστών (προφίλ) και όχι για τα κείμενα. Έτσι όταν παρουσιάζεται ένα καινούργιο κείμενο, πρέπει να ελεγχθεί σε σχέση με την υπάρχουσα δομή ευρετηρίου που υπάρχει για τα προφίλ των χρηστών.

2.3.1 Δομές ευρετηρίων των προφίλ του Boolean μοντέλου

Για τη δημιουργία δομών ευρετηρίων για τις επερωτήσεις των χρηστών (προφίλ) που εκφράζονται με το Boolean μοντέλο έχει προταθεί ένα σύνολο μεθόδων. Για όλες τις μεθόδους έχουν πραγματοποιηθεί κάποιες υποθέσεις, οι οποίες περιγράφονται στη συνέχεια.

Κάθε προφίλ αποτελεί ένα σύνολο από διακριτές λέξεις (w_1, w_2, …, w_k), οι όποιες συνδέονται με το λογικό τελεστή AND. Επομένως ένα προφίλ ταιριάζει με ένα κείμενο εφόσον όλες οι λέξεις του εμφανίζονται στο κείμενο.

Σε κάθε λέξη, η οποία περιέχεται στις επερωτήσεις των χρηστών, αντιστοιχίζεται ένα σύνολο από προφίλ, το οποίο περιλαμβάνει τα προφίλ στα οποία περιέχεται η συγκεκριμένη λέξη. Όλα αυτά τα σύνολα οργανώνονται σε μία ανάστροφη λίστα [34] ή σε ένα δένδρο. Ένα hash table, το οποίο ονομάζεται ευρετήριο (directory), αποτελεί τη δομή που πραγματοποιεί την αντιστοιχία ανάμεσα στην λέξη και στην αντίστοιχη λίστα. Θεωρούμε ότι το ευρετήριο βρίσκεται στην κύρια μνήμη, ενώ τα σύνολα στο δίσκο.

Οι δομές ευρετηρίων δεν περιλαμβάνουν πληροφορίες για τους χρήστες που έχουν υποβάλλει τα προφίλ. Καθώς κάθε προφίλ προσδιορίζεται μοναδικά από ένα

προσδιοριστή (profile identifier), η πληροφορία αυτή είναι αποθηκευμένη σε κάποιο άλλο σημείο του δίσκου και αναφέρεται από αυτόν τον προσδιοριστή.

Επιπλέον για την υλοποίηση της σύγκρισης ανάμεσα σε ένα κείμενο και ένα προφίλ χρησιμοποιούνται δύο επιπλέον δομές: το σύνολο διακριτών λέξεων (distinct word set), το οποίο αποτελείται από τις διακριτές λέξεις που εμφανίζονται σε ένα κείμενο και ο πίνακας εμφάνισης (occurrence table), ο οποίος αντιστοιχεί μία λέξη σε T, εάν αυτή εμφανίζεται στο κείμενο, και σε F διαφορετικά.

Στη συνέχεια παρουσιάζουμε τις τέσσερις πιο δημοφιλείς μεθόδους δημιουργία δομών ευρετηρίων για τα προφίλ.

2.3.1.1 Μέθοδοι Brute Force

Στις μεθόδους Brute Force [35] τα προφίλ των χρηστών αποθηκεύονται σειριακά στο δίσκο χωρίς να γίνεται η οποιαδήποτε ταξινόμησή τους. Επομένως όλα τα προφίλ αποτιμούνται όταν κάποιο νέο κείμενο παρουσιάζεται. Πιο συγκεκριμένα αρχικά κατασκευάζεται ο πίνακας εμφάνισης. Στη συνέχεια ελέγχονται διαδοχικά όλα τα προφίλ και προκύπτει το σύνολο των προφίλ τα οποία ταιριάζουν στο κείμενο. Αυτή η απλή μέθοδος ονομάζεται Τυχαία Brute Force Μέθοδος (Random Brute Force Method). Η απόδοση του συστήματος μπορεί να βελτιωθεί, εάν υπάρχει πληροφορία σχετικά με τη συχνότητα εμφάνισης της κάθε λέξης στο κείμενο. Σε αυτήν την περίπτωση εξετάζονται αρχικά οι λέξεις με την πιο μικρή συχνότητα, δηλαδή οι πιο σπάνιες λέξεις, με σκοπό να αποκλειστούν κάποια προφίλ πιο γρήγορα. Αυτή η μέθοδος καλείται Καταταγμένη Brute Force Μέθοδος (Ranked Brute Force Method).

Οι μέθοδοι Brute Force παρουσιάζουν το βασικό μειονέκτημα ότι λόγω της απουσίας δομής ευρετηρίου, πρέπει να ελεγχθούν όλα τα προφίλ, γεγονός που μειώνει αισθητά την απόδοση του αλγορίθμου. Θεωρώντας ότι στο σύστημα υπάρχουν N προφίλ και M κείμενα, η πολυπλοκότητα της συγκεκριμένης μεθόδου είναι N x M.

2.3.1.2 Μέθοδος Μέτρησης

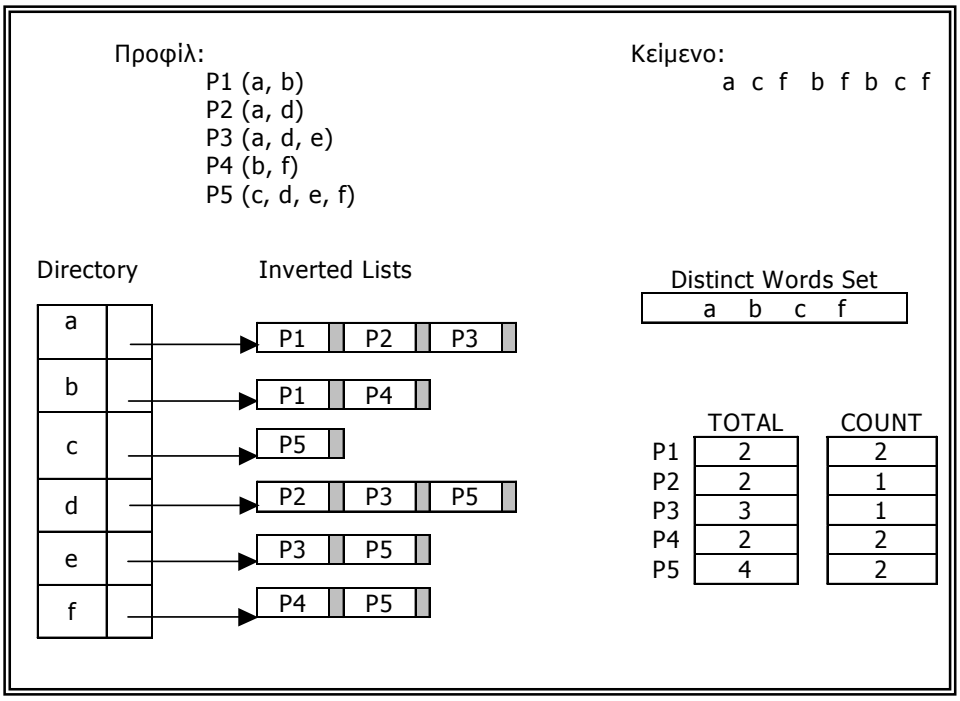
Στη μέθοδο Μέτρησης (Counting Method) [35], δημιουργείται μία ανάστροφη δομή ευρετηρίου για τα προφίλ των χρηστών. Πιο συγκεκριμένα κάθε λέξη

αντιστοιχίζεται σε μία εγγραφή (posting), η οποία περιλαμβάνει τους προσδιοριστές (identifiers) των προφίλ που την περιέχουν. Όλες οι εγγραφές διαμορφώνουν μία ανάστροφη λίστα. Επομένως για την επεξεργασία ενός κειμένου, αρκεί να ελεγχθούν μόνο τα προφίλ που υπάρχουν στις λίστες των λέξεων του κειμένου.

Σε αυτή τη μέθοδο χρησιμοποιούμε δύο επιπλέον πίνακες στην κύρια μνήμη, τους TOTAL και COUNT. Ο πίνακας TOTAL έχει μία καταχώρηση για κάθε προφίλ, στην οποία αποθηκεύεται το πλήθος των λέξεων του προφίλ. Ο πίνακας COUNT έχει επίσης μία καταχώρηση για κάθε προφίλ. Κάθε καταχώρησή του αρχικοποιείται σε 0 και στη συνέχει κρατάει τον αριθμό των εμφανίσεων του προφίλ στις λίστες.

Όταν παρουσιάζεται ένα νέο κείμενο, αρχικοποιείται ο πίνακας COUNT σε 0 και κατασκευάζεται το σύνολο διακριτών λέξεων. Στη συνέχεια χρησιμοποιείται το directory για να ανακτηθεί η λίστα της κάθε διακριτής λέξης του κειμένου. Για κάθε προφίλ που εμφανίζεται στη λίστα, αυξάνεται η αντίστοιχη καταχώρηση τον πίνακα TOTAL κατά 1. Επομένως ένα προφίλ ταιριάζει το κείμενο, όταν η καταχώρησή του στον πίνακα COUNT γίνεται ίση με την καταχώρησή του στον πίνακα TOTAL.

Η διαδικασία του «ταιριάσματος» φαίνεται στο Σχήμα 3 με ένα παράδειγμα, στο οποίο θεωρούμε πέντε προφίλ και ένα κείμενο. Όπως διακρίνεται από τους πίνακες TOTAL και COUNT, μόνο τα προφίλ P1 και P4 ταιριάζουν στο κείμενο.



Σχήμα 3. Δομές δεδομένων για τη μέθοδο Μέτρησης

2.3.1.3 Μέθοδοι Κλειδιού

Η διαφοροποίηση των μεθόδων Κλειδιού (Key Methods) [35] από τις μεθόδους Μέτρησης στηρίζεται στο γεγονός ότι ένα προφίλ εμφανίζεται στη λίστα μόνο μίας λέξης, η οποία καλείται κλειδί (key). Σημειώνουμε ότι σαν κλειδί σε καμία περίπτωση δεν επιλέγεται μία λέξη η οποία ακολουθεί τον τελεστή άρνησης (AND NOT). Στη μέθοδο Τυχαίου Κλειδιού (Random Key) αυτή η λέξη επιλέγεται τυχαία. Στη μέθοδο Καταταγμένου Κλειδιού (Ranked Key), η λέξη που επιλέγεται είναι αυτή με το μικρότερο βαθμό, δηλαδή με τη μικρότερη συχνότητα εμφάνισης στα κείμενα. Επομένως για την υλοποίηση της δεύτερης μεθόδου είναι απαραίτητα η συγκεκριμένη πληροφορία. Ως αποτέλεσμα, οι πιο συχνές λέξεις έχουν λιγότερα προφίλ να σχετίζονται με αυτές, επομένως κατά μέσο όρο λιγότερα προφίλ πρέπει να εξεταστούν ανά κείμενο. Άρα το βασικό όφελος της μεθόδου είναι η μείωση του πλήθους των προφίλ που εξετάζονται σε κάθε κείμενο.

Οι εγγραφές (postings) των ανάστροφων λιστών περιλαμβάνουν πέρα από τον προσδιοριστή του προφίλ, το μήκος του προφίλ και τις υπόλοιπες λέξεις πέρα από το κλειδί. Οι εγγραφές της ίδιας λίστας αποθηκεύονται σειριακά σε blocks.

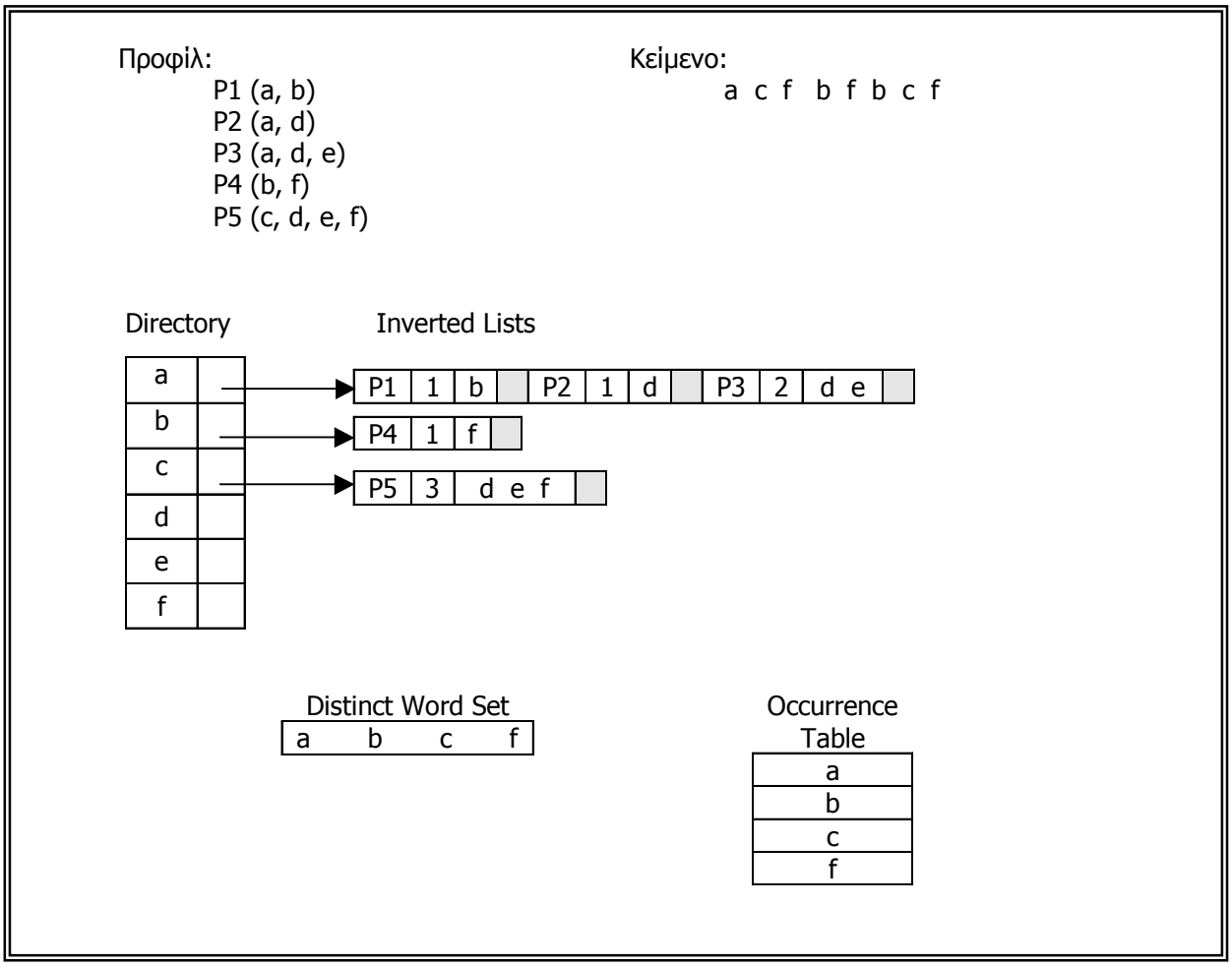
Όταν παρουσιάζεται ένα νέο κείμενο, κατασκευάζεται το σύνολο διακριτών λέξεων και ο πίνακας εμφάνισης. Χρησιμοποιώντας το ευρετήριο (directory), ανακτώνται οι λίστες των διακριτών λέξεων του κειμένου. Στη συνέχεια εξετάζεται κάθε προφίλ στη λίστα εάν ταιριάζει με το κείμενο (με βάση τον πίνακα εμφάνισης).

Στο Σχήμα 4, φαίνεται ένα παράδειγμα που δείχνει τη λειτουργία της μεθόδου Καταταγμένου Κλειδιού. Θεωρούμε ότι η λέξη a έχει τη μικρότερη συχνότητα εμφάνισης, στη συνέχεια η λέξη b κτλ. Έτσι, τα προφίλ P1, P2 και P3 εισάγονται στο λίστα της λέξης a, το προφίλ P4 εισάγεται στη λίστα της λέξης b και το προφίλ P5 εισάγεται στη λίστα της λέξης c, με αποτέλεσμα να δημιουργείται η δομή που φαίνεται στο σχήμα.

Έστω ότι το κείμενο που πρέπει να εξεταστεί έναντι της δομής των προφίλ είναι το

D (a, c, f, b, f, b, c, f)

Αρχικά δημιουργείται το σύνολο των διακριτών λέξεων του κειμένου και ο πίνακας εμφάνισης. Για την επεξεργασία του κειμένου ελέγχουμε διαδοχικά τις λέξεις a, b, c, f στο directory.



Σχήμα 4. Δομές δεδομένων για τη μέθοδο Καταταγμένου Κλειδιού

- Αρχικά ελέγχουμε τα προφίλ στη λίστα της λέξης a. Για το πρώτο προφίλ της λίστας, το P1, παρατηρούμε ότι η λέξη b εμφανίζεται στον πίνακα εμφάνισης, οπότε το P1 ταιριάζει το κείμενο. Για το προφίλ P2 παρατηρούμε, ότι η λέξη d δεν εμφανίζεται στον πίνακα εμφάνισης, οπότε το P2 απορρίπτεται. Για το προφίλ P3 παρατηρούμε ότι οι λέξεις d, e δεν εμφανίζονται στον πίνακα εμφάνισης, οπότε το P3 απορρίπτεται.
- Στη συνέχεια, ελέγχουμε τα προφίλ της λίστας της λέξης b. Σε αυτή τη λίστα υπάρχει μόνο το προφίλ P4, για το οποίο παρατηρούμε ότι η λέξη f υπάρχει στον πίνακα εμφάνισης, οπότε το P4 ταιριάζει το κείμενο.
- Ελέγχουμε έπειτα τα προφίλ της λίστας της λέξης c. Το προφίλ P5, το οποίο εμφανίζεται στη λίστα, δεν ταιριάζει στο κείμενο, καθώς οι λέξεις d, e δεν εμφανίζονται στον πίνακα εμφάνισης.

- Τέλος, καθώς η λέξη f δεν έχει κάποια προφίλ να συνδέονται μαζί της, η επεξεργασία ολοκληρώνεται.

Επομένως τελικά μόνο τα προφίλ P1 και P4 ταιριάζουν το κείμενο D.

Παρατηρούμε ότι με τη χρήση της συγκεκριμένης μεθόδου, ο αριθμός των προφίλ που εξετάζονται για κάθε κείμενο μειώνεται αισθητά. Επομένως αυξάνεται η απόδοση του συστήματος σε σχέση με την απόδοση που θα είχαμε στην περίπτωση που τα προφίλ ήταν συνδεδεμένα με όλους τους όρους που περιέχουν.

2.3.1.4 Μέθοδοι Δένδρου

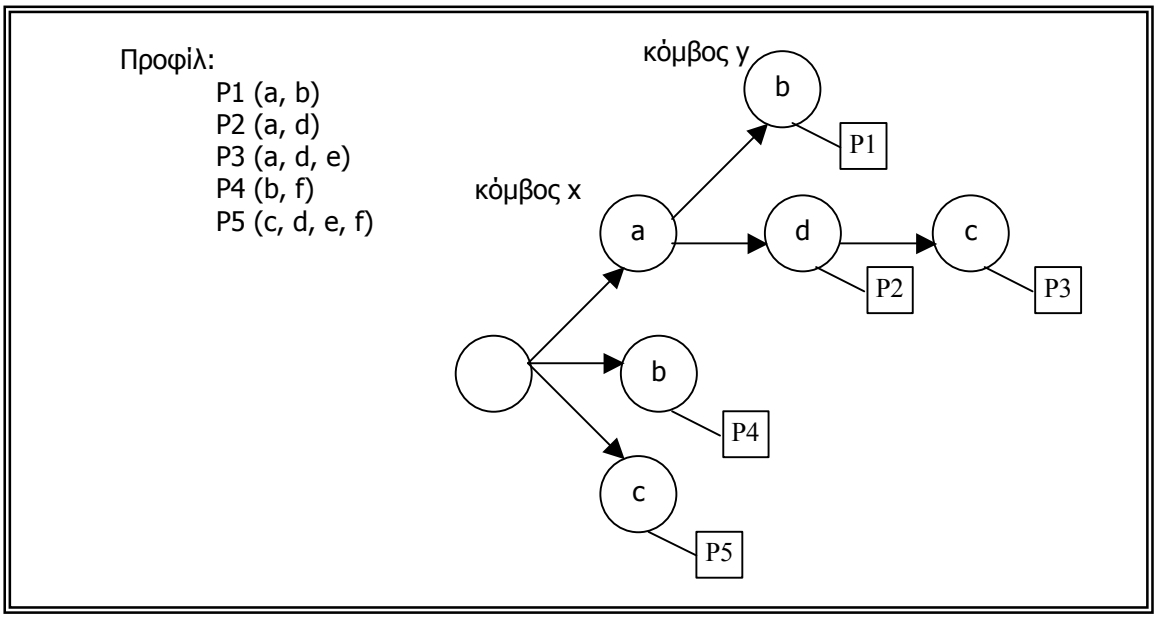
Οι μέθοδοι Δένδρου (Tree Methods) [35] εκμεταλλεύονται την παρατήρηση ότι τα προφίλ έχουν κάποιες κοινές λέξεις, οπότε αποθηκεύονται με τη μορφή δένδρου. Έστω ένα προφίλ P = (w_1, w_2, …, w_k). Ορίζουμε σαν πρόθεμα (prefix) του P το σύνολο (w_1, …, w_i), όπου $0 \leq i \leq k$, και σαν κατάληξη (postfix) του P το σύνολο (w_{i+1}, …, w_k). Ένα πρόθεμα (w_1, …, w_i) προσδιορίζει ένα προφίλ P εάν i = k ή εάν δεν υπάρχει άλλο προφίλ, εκτός από αυτά που είναι όμοια με το P, το οποίο να έχει σαν πρόθεμα το (w_1, …, w_i). Το μικρότερο πρόθεμα που προσδιορίζει ένα προφίλ καλείται προσδιοριστικό πρόθεμα για αυτό το προφίλ. Τα προσδιοριστικά προθέματα των προφίλ είναι αυτά που οργανώνονται σε δενδρική δομή.

Εάν θεωρήσουμε ότι η ρίζα του δένδρου είναι στο επίπεδο 0, ένας κόμβος n στο επίπεδο i αντιστοιχεί στο πρόθεμα σ = (w_1, …, w_i) κάποιων προσδιοριστικών προθεμάτων. Όλα τα προθέματα που είναι όμοια με το σ αντιπροσωπεύονται από τον ίδιο κόμβο n. Τα παιδιά του κόμβου αντιστοιχούν σε προθέματα (w_1, …, wi, u) κάποιων προσδιοριστικών προθεμάτων. Ο κόμβος n έχει τα εξής πεδία:

- παιδιά (children), το οποίο είναι μία λίστα από ζευγάρια (u, $p_n(u)$). Το u είναι μία λέξη τέτοια ώστε το (w_1, …, w_i, u) να είναι το πρόθεμα που αντιστοιχεί στο παιδί του κόμβου n, και το $p_n(u)$ είναι ένας δείκτης σε αυτό το παιδί.
- προφίλ (profiles), το οποίο είναι μία λίστα με τα προφίλ που έχουν το σ σαν προσδιοριστικό πρόθεμα.
- μήκος (length), το οποίο αναφέρει το μήκος της κατάληξης των προφίλ που προσδιορίζονται από το σ.
- κατάληξη (postfix), το οποίο περιλαμβάνει τις λέξεις που συγκροτούν την κατάληξη των προφίλ.

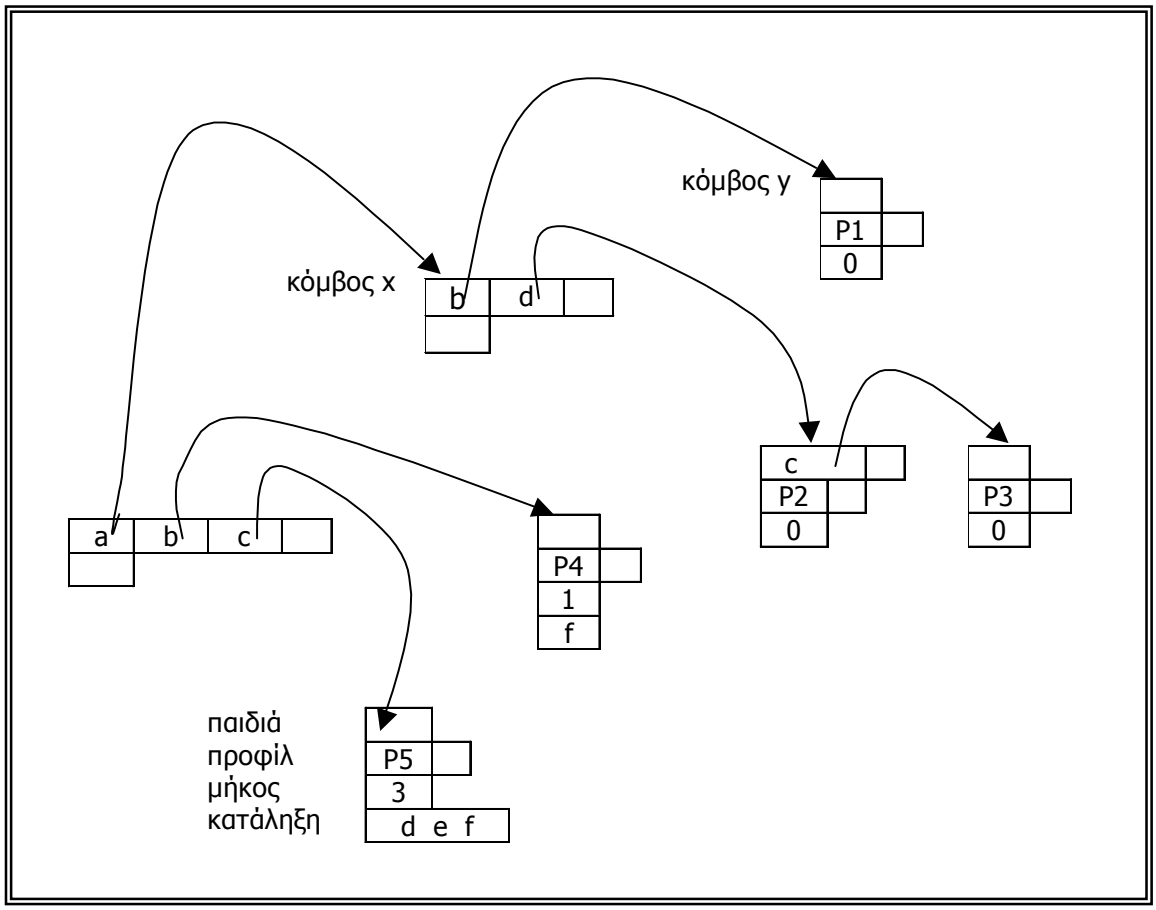
Τα δύο τελευταία πεδία δεν παρουσιάζονται πάντα.

Στο Σχήμα 5 παρουσιάζεται η δενδρική δομή για τα προφίλ που έχουν επιλεχθεί και στις προηγούμενες μεθόδους. Στο Σχήμα 6 παρουσιάζεται η εσωτερική δομή του δένδρου.



Σχήμα 5. Δομή δεδομένων για τη μέθοδο Δένδρου

Όταν παρουσιάζεται ένα νέο κείμενο, αρχικά κατασκευάζονται το σύνολο διακριτών λέξεων και ο πίνακας εμφάνισης. Για κάθε διακριτή λέξη που εμφανίζεται στο κείμενο ανακτούμε το υποδένδρο της. Παράλληλα διατηρούμε μία ουρά με τους δείκτες του δένδρου που πρέπει να επισκεφτούμε. Η ουρά αρχικοποιείται με ένα δείκτη προς τη ρίζα του υποδένδρου της λέξης. Ακολουθώντας αυτό το δείκτη, ανακτούμε τον κόμβο στον οποίο αυτόν δείχνει. Ελέγχουμε αν η λίστα «παιδιά» του συγκεκριμένου κόμβου έχει λέξεις που περιέχονται στο κείμενο. Οι δείκτες οι οποίοι αντιστοιχούν σε λέξεις που περιέχονται στο κείμενο προστίθενται στην ουρά. Στη συνέχεια ελέγχουμε το πεδίο της κατάληξης, εφόσον αυτό έχει κάποια τιμή. Εάν οι λέξεις που περιέχονται σε αυτό το πεδίο εμφανίζονται στο κείμενο, τότε τα προφίλ που υπάρχουν στο πεδίο προφίλ ταιριάζουν το κείμενο. Εάν το πεδίο της κατάληξης είναι άδειο, τότε επίσης τα προφίλ που υπάρχουν στο πεδίο προφίλ ταιριάζουν το κείμενο.



Σχήμα 6. Εσωτερική δομή δένδρου μεθόδου Δένδρου

Η μέθοδος Δένδρου είναι ιδιαίτερα αποδοτική όταν υπάρχουν πολλά κοινά προθέματα ανάμεσα στα προφίλ του συστήματος. Ένας τρόπος για να αυξηθεί το πλήθος των κοινών προθεμάτων ανάμεσα στα προφίλ είναι η ταξινόμηση των λέξεων των προφίλ. Εάν μάλιστα είναι διαθέσιμη κάποια πληροφορία σχετικά με τη συχνότητα εμφάνισης των λέξεων, η ταξινόμηση μπορεί να στηριχθεί σε αυτήν την πληροφορία. Δηλαδή όσο λιγότερο συχνή είναι η εμφάνιση μιας λέξης τόσο πιο ψηλά είναι αυτή η λέξη στην ταξινομημένη λίστα. Αυτή η μέθοδος, η οποία καλείται Καταταγμένη Μέθοδος Δένδρου (Ranked Tree Method), έχει το πλεονέκτημα ότι απορρίπτει πολύ γρήγορα κάποια προφίλ. Εάν δεν είναι διαθέσιμη η πληροφορία της συχνότητας μίας λέξης, τότε η μέθοδος ονομάζεται Τυχαία Μέθοδος Δένδρου (Random Tree Method).

2.3.1.5 Επεκτάσεις

Παραπάνω θεωρήσαμε ότι οι επερωτήσεις περιέχουν μόνο των τελεστή AND. Ωστόσο οι συγκεκριμένες τεχνικές μπορούν να επεκταθούν και στις περιπτώσεις που οι επερωτήσεις περιέχουν τους τελεστές OR, AND NOT ή τελεστές εγγύτητας [35].

Αρχικά, τα προφίλ που περιέχουν τον τελεστή OR μπορούν να μετασχηματιστούν σε DNF και κατά συνέπεια να αντιμετωπιστούν σαν συζευκτικά επερωτήματα. Όσον αφορά τα προφίλ που περιέχουν άρνηση, αυτά μπορούν να χειριστούν με βάση την εξής παρατήρηση: κατά τη διάρκεια του ταιριάσματος (matching), μία λέξη με άρνηση θεωρείται ότι ικανοποιεί το κείμενο εφόσον αυτή δε βρίσκεται στο κείμενο. Απλά απαιτείται η χρήση ενός bit για κάθε λέξη, το οποίο θα δηλώνει εάν αυτή η λέξη είναι αρνητική ή όχι. Τέλος, τα προφίλ που περιέχουν τελεστές εγγύτητας αντιμετωπίζονται σαν κανονικά συζευκτικά προφίλ. Μετά τον εντοπισμό των κειμένων που ταιριάζουν σε αυτά τα προφίλ, πραγματοποιείται κάποια επεξεργασία η οποία ελέγχει αποκλειστικά εάν ικανοποιούνται οι συνθήκες εγγύτητας.

2.3.2 Δομές ευρετηρίων των προφίλ του Vector Space μοντέλου

Όπως στην περίπτωση του μοντέλου Boolean, και στο Vector Space μοντέλο χρησιμοποιείται το ανάστροφο ευρετήριο. Σε κάθε όρο x αντιστοιχίζουμε μία λίστα με τα προφίλ που περιέχουν αυτόν τον όρο. Η αντιστοίχηση των όρων με την τοποθεσία που είναι αποθηκευμένες οι λίστες στο δίσκο υλοποιείται με ένα hash table, το οποίο ονομάζεται ευρετήριο (directory). Επομένως θεωρούμε ότι το ευρετήριο είναι αποθηκευμένο στην κύρια μνήμη, ενώ οι λίστες στο δίσκο. Στη συνέχεια παρουσιάζονται τρεις μέθοδοι που χρησιμοποιούνται για τη δημιουργία δομών ευρετηρίων των προφίλ που είναι εκφρασμένα με το συγκεκριμένο μοντέλο.

2.3.2.1 Μέθοδος Brute Force

Στη μέθοδο Brute Force [36, 37] δεν υλοποιείται κάποιο ευρετήριο των προφίλ, με αποτέλεσμα όλα τα προφίλ να αποτιμούνται, όταν παρουσιάζεται κάποιο κείμενο. Πιο συγκεκριμένα, όταν εμφανίζεται ένα νέο κείμενο, αρχικά διαμορφώνεται

η διανυσματική αναπαράστασή του. Στη συνέχεια εξετάζεται διαδοχικά με όλα τα προφίλ. Για κάθε ζεύγος (x, u) σε ένα προφίλ, εντοπίζεται το βάρος w του όρου x στο διάνυσμα του κειμένου και υπολογίζεται το γινόμενο w*u. Το άθροισμα τέτοιων γινομένων παράγει το cosine similarity measure. Επομένως το κείμενο είναι σχετικό με ένα προφίλ, εφόσον το cosine similarity measure είναι μεγαλύτερο από το κατώφλι σχετικότητας (relevant threshold) που έχει συσχετιστεί με το προφίλ.

2.3.2.2 Μέθοδος Ευρετηρίασης Προφίλ

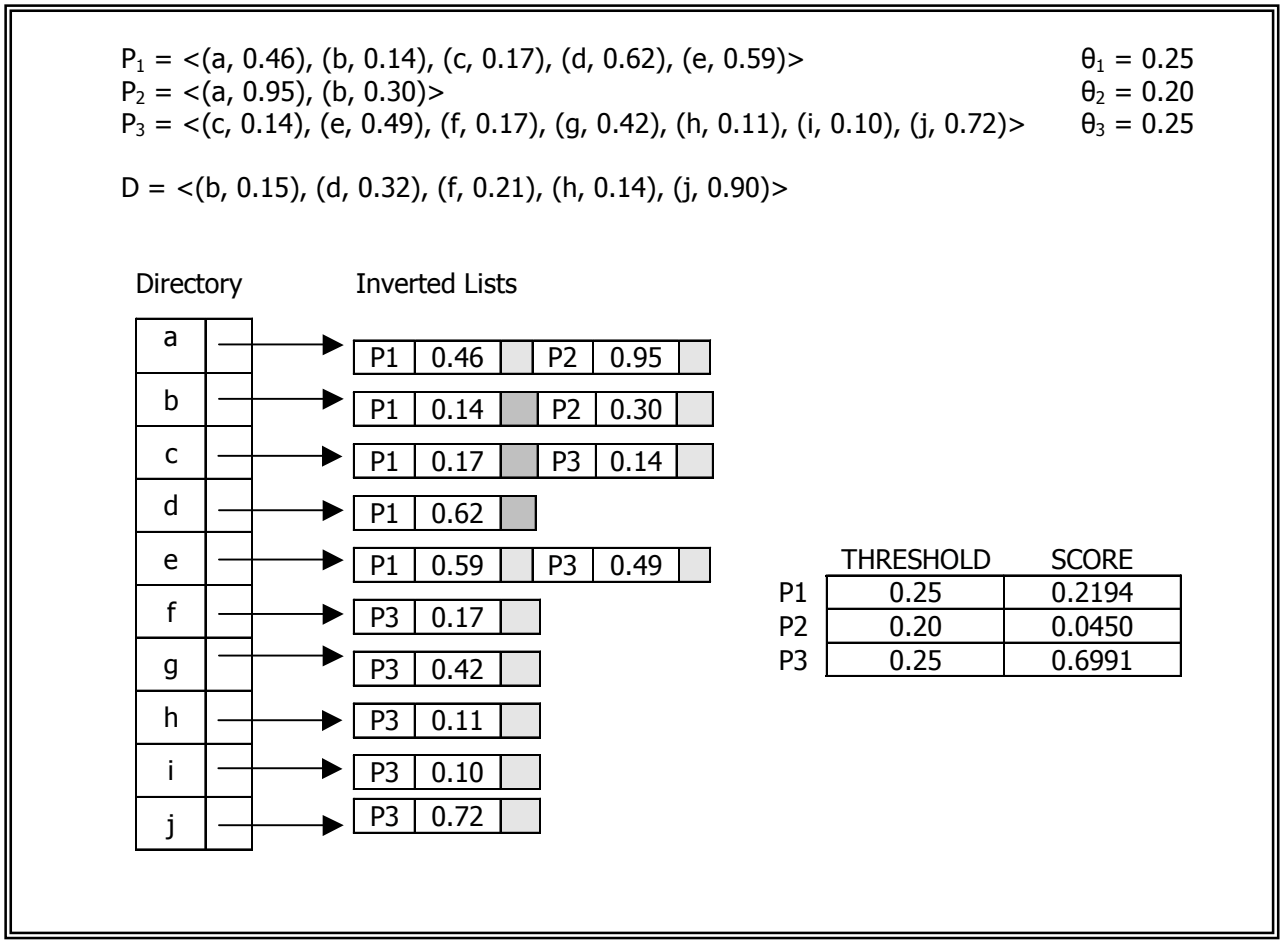
Στη μέθοδο Ευρετηρίασης Προφίλ (Profile Indexing ή PI Method) [36, 37], υλοποιείται ένα ανάστροφο ευρετήριο με προφίλ, με σκοπό να μειωθεί ο αριθμός των προφίλ που συγκρίνονται με το κείμενο. Κάθε όρος αντιστοιχίζεται σε μία εγγραφή (posting), η οποία περιλαμβάνει τους προσδιοριστές (identifiers) των προφίλ που τον περιέχουν και το βάρος του όρου σε αυτά τα προφίλ. Όλες οι εγγραφές διαμορφώνουν μία ανάστροφη λίστα. Επομένως για την επεξεργασία ενός κειμένου, αρκεί να ελεγχθούν μόνο τα προφίλ που υπάρχουν στις λίστες των όρων του κειμένου.

Για την υλοποίηση του «ταιριάσματος» (matching), χρησιμοποιούνται δύο επιπλέον πίνακες στην κύρια μνήμη, οι THRESHOLD και SCORE. Ο πίνακας THRESHOLD έχει μία καταχώρηση για κάθε προφίλ, η οποία αποθηκεύει το αντίστοιχο κατώφλι σχετικότητας. Ο πίνακας SCORE έχει επίσης μία καταχώρηση για κάθε προφίλ, η οποία διατηρεί τη βαθμολογία του προφίλ.

Όταν εμφανίζεται ένα νέο κείμενο, ο πίνακας SCORE αρχικοποιείται με 0. Για κάθε όρο του κειμένου, χρησιμοποιείται το ευρετήριο για να ανακτηθεί η ανάστροφη λίστα του. Στη συνέχεια επεξεργάζεται κάθε προφίλ που υπάρχει στη λίστα του όρου. Υποθέτοντας ότι το βάρος του όρου στο κείμενο είναι w και στο προφίλ είναι u, αυξάνεται η αντίστοιχη καταχώρηση του πίνακα SCORE με το γινόμενο w*u. Μετά την επεξεργασία όλων των όρων του κειμένου, ένα προφίλ ταιριάζει ένα κείμενο εφόσον η καταχώρησή του στον πίνακα SCORE είναι μεγαλύτερη από την καταχώρησή του στον πίνακα THRESHOLD.

Η μέθοδος φαίνεται και στο Σχήμα 7, στο οποίο παρουσιάζονται τα προφίλ με τα αντίστοιχα κατώφλια, το διάνυσμα του κειμένου και η ανάστροφη λίστα των προφίλ μαζί με τους δύο βοηθητικούς πίνακες. Ακολουθώντας τη διαδικασία που

περιγράφτηκε παραπάνω, καταλήγουμε στο συμπέρασμα ότι μόνο το προφίλ P3 ταιριάζει στο κείμενο.



Σχήμα 7. Δομή δεδομένων για τη μέθοδο Ευρετηρίασης Προφίλ

2.3.2.3 Μέθοδος Επιλεκτικής Ευρετηρίασης Προφίλ

Η μέθοδος Επιλεκτικής Ευρετηρίασης Προφίλ (Selective Profile Indexing ή SPI Method) [36, 37] διαφοροποιείται σε σχέση με την προηγούμενη ως προς το γεγονός ότι δε χρησιμοποιεί όλους τους όρους του προφίλ για την ταξινόμησή του, παρά μόνο κάποιους που θεωρεί πιο σημαντικούς. Η συγκεκριμένη επιλογή γίνεται, καθώς οι μη σημαντικοί όροι ενός προφίλ δεν είναι από μόνοι τους ικανοί να πετύχουν κάποιο σκορ που να ξεπερνάει το κατώφλι του προφίλ. Έτσι δεν έχει νόημα τα προφίλ να βρίσκονται στις λίστες των μη σημαντικών όρων.

Δεδομένου ενός διανύσματος προφίλ $P = ((y_1, u_1), \ldots, (y_p, u_p))$, ένα υποδιάνυσμα $P_s = ((y_{i1}, u_{i1}), \ldots, (y_{is}, u_{is}))$, όπου $1 \le i_1 < \ldots < i_s \le p$, είναι μη σημαντικό για ένα κατώφλι θ, εάν για κάθε κείμενο D, $\mathrm{sim}(D,P_s) \le \theta$.

Κάθε προφίλ έχει πολλά μη σημαντικά υποδιανύσματα. Για τη μείωση των postings επιλέγουμε από όλα τα μη σημαντικά υποδιανύσματα, αυτό το οποίο έχει όρους με τις χαμηλότερες idf τιμές, καθώς αυτοί οι όροι εμφανίζονται πιο συχνά στα κείμενα.

Δεδομένου ενός διανύσματος προφίλ $P = ((y_1, u_1), \ldots, (y_p, u_p))$, ένα υποδιάνυσμα $P_s = ((y_{i1}, u_{i1}), \ldots, (y_{is}, u_{is}))$, όπου $1 \le i_1 < \ldots < i_s \le p$, είναι το περισσότερο μη σημαντικό για ένα κατώφλι θ, εάν αυτό έχει το μεγαλύτερο αριθμό όρων με τις χαμηλότερες τιμές idf σε σχέση με όλα τα μη σημαντικά υποδιανύσματα στο κατώφλι θ.

Υποθέτοντας ότι οι τιμές idf είναι διακριτές, κάθε διάνυσμα προφίλ έχει ένα και μοναδικό περισσότερο μη σημαντικό υποδιάνυσμα με δεδομένο κατώφλι. Επομένως απαιτείται η ύπαρξη μίας μεθόδου, η οποία να ελέγχει εάν κάποιο υποδιάνυσμα είναι το περισσότερο μη σημαντικό. Αυτό απαιτεί την ικανότητα να υπολογιστεί η μέγιστη πιθανή ομοιότητα ανάμεσα στο υποδιάνυσμα του προφίλ και στο διάνυσμα του κειμένου. Η ομοιότητα ανάμεσα στα δύο διανύσματα είναι μέγιστη όταν αυτά βρίσκονται προς την ίδια κατεύθυνση στο χώρο που ορίζεται από το σύνολο των όρων. Εάν συμβαίνει αυτό, τότε η ομοιότητα δίνεται από τη νόρμα του υποδιανύσματος του προφίλ. Επομένως έχουμε τον παρακάτω ορισμό: Για κάθε προφίλ P και κείμενο D, $\|D\| \le 1, \mathrm{sim}(D,P) \le \|P\|$.

Τελικά, λοιπόν, για να εντοπίσουμε το πιο μη σημαντικό υποδιάνυσμα ενός προφίλ, αρκεί να ταξινομήσουμε τους όρους του με βάση τις τιμές idf και να εισάγουμε στο υποδιάνυσμα όσο περισσότερους όρους μπορούμε, χωρίς να ξεπεράσουμε την τιμή του κατωφλίου.

Επομένως έχοντας τη συγκεκριμένη πληροφορία, ταξινομούμε τα προφίλ επιλεκτικά. Για κάθε προφίλ υπολογίζουμε το περισσότερο μη σημαντικό υποδιάνυσμά του, στο συγκεκριμένο κατώφλι. Στη συνέχεια το προφίλ τοποθετείται στις λίστες των σημαντικών όρων. Σε κάθε εγγραφή (posting), πέρα από το προσδιοριστικό του προφίλ και το βάρος του σημαντικού όρου, εισάγονται και όλοι οι μη σημαντικοί όροι με τα βάρη τους. Δηλαδή, οι μη σημαντικοί όροι με τα βάρη τους αντιγράφονται σε όλες τις λίστες των σημαντικών όρων. Επιπλέον σε κάθε εγγραφή υπάρχει και ένας αριθμός που δηλώνει το πλήθος των μη σημαντικών όρων του

προφίλ που έχουν εισαχθεί. Οι εγγραφές της ίδιας λίστας αποθηκεύονται σειριακά σε blocks.

Για την επεξεργασία των προφίλ απαιτούνται δύο πίνακες, οι THRESHOLD και SCORE. Ο πίνακας THRESHOLD έχει μία καταχώρηση για κάθε προφίλ, η οποία αποθηκεύει το κατώφλι του. Ο πίνακας SCORE έχει επίσης μία καταχώρηση για κάθε προφίλ, η οποία διατηρεί τη βαθμολογία του προφίλ ως προς ένα κείμενο.

Όταν παρουσιάζεται, λοιπόν, ένα κείμενο, κατασκευάζεται αρχικά η διανυσματική του αναπαράσταση και αρχικοποιείται ο πίνακας SCORE με 0. Για κάθε όρο του κειμένου ανακτάται μέσω του ευρετηρίου (directory) η ανάστροφη λίστα του. Υποθέτουμε ότι το βάρος του όρου x στο κείμενο είναι w, στο προφίλ είναι u και τα μη σημαντικά ζευγάρια είναι (y_{i1}, u_{i1}), ..., (y_{is}, u_{is}). Ελέγχουμε την καταχώρηση του προφίλ στον πίνακα SCORE και διακρίνουμε δύο περιπτώσεις. Στην πρώτη περίπτωση, στην οποία η καταχώρηση είναι 0, προσθέτουμε το γινόμενο w*u. Στη συνέχεια, ελέγχουμε κάθε όρο y_{ij} στο διάνυσμα του κειμένου. Έστω ότι τα βάρη αυτών των όρων είναι w_{ij}. Τότε προσθέτουμε στην καταχώρηση του πίνακα SCORE το γινόμενο $u_{ij}*w_{ij}$. Στη δεύτερη περίπτωση, η καταχώρηση του πίνακα SCORE είναι μη μηδενική, γεγονός που σημαίνει ότι έχει ήδη προστεθεί η συμβολή των μη σημαντικών όρων. Τότε προσθέτουμε μόνο το γινόμενο w*u. Μετά την επεξεργασία όλων των όρων του κειμένου, ένα προφίλ ταιριάζει ένα κείμενο εφόσον η καταχώρησή του στον πίνακα SCORE είναι ίση ή μεγαλύτερη από την καταχώρησή του στον πίνακα THRESHOLD.

Στο Σχήμα 8 φαίνεται ένα παράδειγμα με τη χρήση της συγκεκριμένης μεθόδου. Η απόφαση για το εάν ένα κείμενο ταιριάζει με ένα προφίλ λαμβάνεται με τον ίδιο τρόπο όπως στην PI μέθοδο. Τα προφίλ που υπάρχουν στο σύστημα με τα κατώφλια που έχουν οριστεί από τους χρήστες είναι τα εξής:

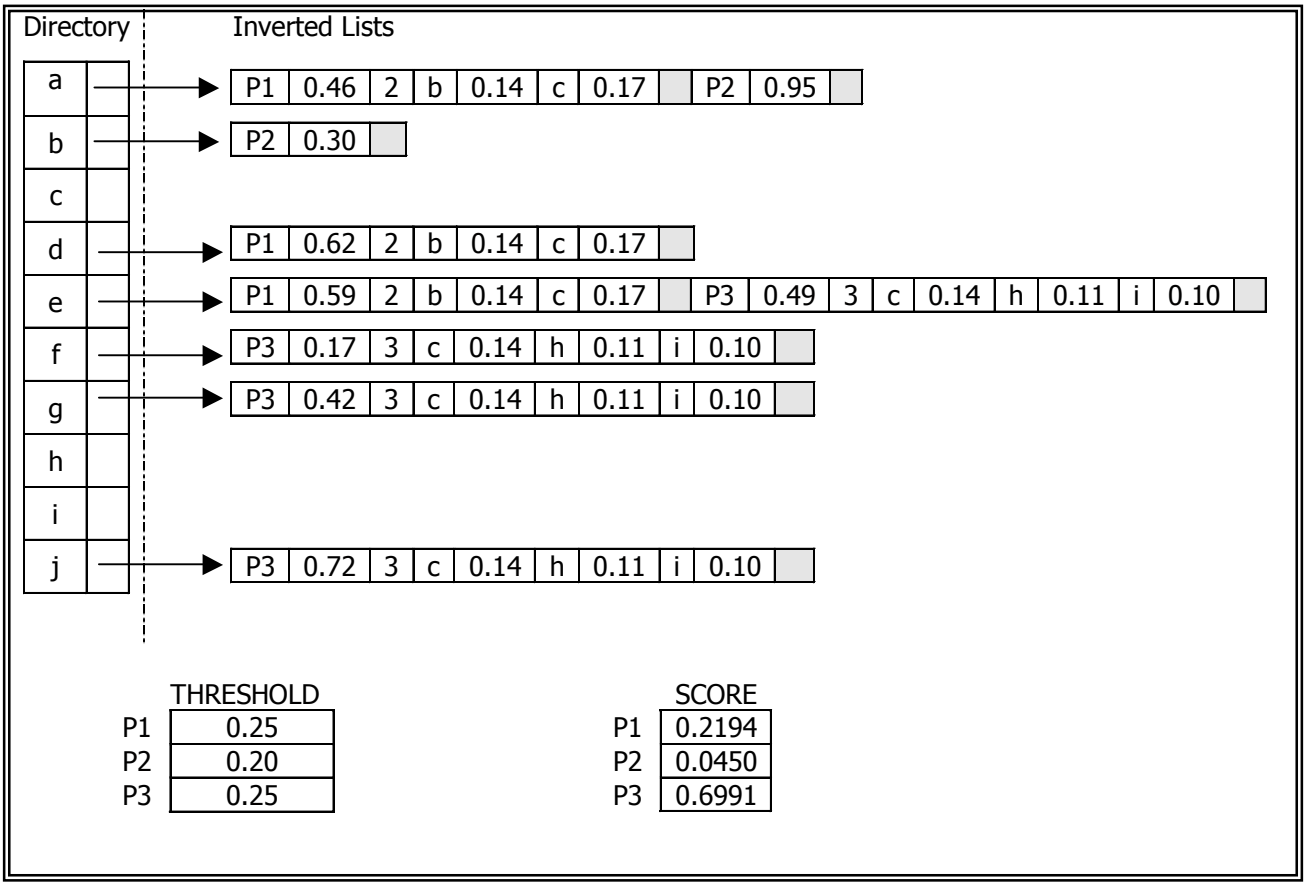
- P_1 = <(a, 0.46), (b, 0.14), (c, 0.17), (d, 0.60), (e, 0.59)>, $\theta_1 = 0.25$
- P_2 = <(a, 0.95), (b, 0.30)>, $\theta_2 = 0.20$
- P_3 = <(c, 0.14), (e, 0.49), (f, 0.17), (g, 0.42), (h, 0.11), (i, 0.10), (j, 0.72)>, $\theta_3 = 0.25$

Αρχικά πρέπει να εντοπιστούν τα περισσότερο μη σημαντικά διανύσματα των προφίλ, έτσι ώστε τα προφίλ να εισαχθούν στις λίστες μόνο των πιο σημαντικών όρων. Εκτελούμε τη διαδικασία που προαναφέρθηκε για κάθε προφίλ:

- Για το προφίλ P_1 έχουμε:

$$\|((b,0.14),(c,0.17))\| = 0.2202 \le 0.25 = \theta_1$$
$$\|((b,0.14),(c,0.17),(a,0.46))\| = 0.51 > 0.25 = \theta_1$$

Επομένως οι μη σημαντικοί όροι του προφίλ είναι οι b και c και το προφίλ θα εισαχθεί μόνο στα postings των όρων a, d, e.



Σχήμα 8. Δομή δεδομένων για τη μέθοδο Επιλεκτικής Ευρετηρίασης Προφίλ

- Για το προφίλ P_2 έχουμε:

  $$\|((b,0.30))\| = 0.3 > 0.20 = \theta_2$$

  Το προφίλ P_2 δεν έχει μη σημαντικούς όρους, άρα θα εισαχθεί στις λίστες όλων των όρων του.

- Για το προφίλ P_3 έχουμε:

  $$\|((i,0.10),(h,0.11),(c,0.14))\| = 0.2042 \le 0.25 = \theta_3$$
  $$\|((i,0.10),(h,0.11),(c,0.14),(f,0.17))\| = 0.2657 > 0.25 = \theta_3$$

  Επομένως οι μη σημαντικοί όροι του προφίλ είναι οι i, h και c και το προφίλ θα εισαχθεί μόνο στα postings των όρων e, f, g και j.

Έστω ότι εμφανίζεται το κείμενο D το οποίο έχει την παρακάτω διανυσματική αναπαράσταση:

D = <(b, 0.15), (d, 0.32), (f, 0.21), (h, 0.14), (j, 0.90)>

Ο πίνακας SCORE αρχικοποιείται με μηδενικά. Στη συνέχεια πρέπει να υπολογιστούν οι καταχωρήσεις του συγκεκριμένου πίνακα, γεγονός το οποίο γίνεται με την επεξεργασία του κάθε ζεύγους του διανύσματος του κειμένου.

- Επεξεργασία ζεύγους (b, 0.15)

  Η λίστα του όρου b περιλαμβάνει μόνο ένα προφίλ, το P_2. Επομένως τροποποιούμε την καταχώρηση του πίνακα SCORE για το συγκεκριμένο προφίλ ως εξής:

  SCORE [P_2] = (βάρος b στο P_2) * (βάρος b στο D) = 0.30 * 0.15 = 0.045

- Επεξεργασία ζεύγους (d, 0.32)

  Η λίστα του όρου d έχει μόνο το προφίλ P_1. Αρχικά προσθέτουμε τη συμβολή του όρου d στην καταχώρηση του πίνακα SCORE για το προφίλ P_1, δηλαδή

  SCORE [P_1] = (βάρος d στο P_1) * (βάρος d στο D) = 0.62 * 0.32 = 0.1984

  Στη συνέχεια πρέπει να πραγματοποιηθεί η επεξεργασία του μη σημαντικού υποδιανύσματος ((b, 0.14), (c, 0.17)) του προφίλ. Ο όρος b υπάρχει στο διάνυσμα του κειμένου, οπότε προσθέτουμε τη συμβολή του στον πίνακα SCORE, ως εξής

  SCORE [P_1] = SCORE [P_1] + (βάρος b στο P_1) * (βάρος b στο D) =

  = 0.1984 + 0.14 * 0.15 = 0.1984 + 0.021 = 0.2194.

  Ο όρος c δεν υπάρχει στο διάνυσμα του κειμένου, οπότε ολοκληρώνεται η επεξεργασία του ζεύγους.

- Επεξεργασία ζεύγους (f, 0.21)

  Η λίστα του όρου f περιλαμβάνει μόνο το προφίλ P_3. Αρχικά προσθέτουμε τη συμβολή του όρου f στην καταχώρηση του πίνακα SCORE για το προφίλ P_3, δηλαδή

  SCORE [P_3] = (βάρος f στο P_3) * (βάρος f στο D) = 0.17 * 0.21 = 0.0357

  Στη συνέχεια πρέπει να πραγματοποιηθεί η επεξεργασία του μη σημαντικού υποδιανύσματος ((c, 0.14), (h, 0.11), (i, 0.10)) του προφίλ. Οι όροι c και i δεν εμφανίζονται στο διάνυσμα του κειμένου. Ο όρος h

υπάρχει στο διάνυσμα του κειμένου, οπότε προσθέτουμε τη συμβολή του στον πίνακα SCORE, ως εξής

SCORE [P_3] = SCORE [P_3] + (βάρος h στο P_3) * (βάρος h στο D) =

= 0.0357 + 0.11 * 0.14 = 0.0357 + 0.0154 = 0.0511.

- Επεξεργασία ζεύγους (h, 0.14)

  Ο όρος h δεν έχει κανένα posting.

- Επεξεργασία ζεύγους (j, 0.90))

  Ο όρος j περιλαμβάνει στη λίστα του μόνο το προφίλ P_3. Αρχικά προσθέτουμε τη συμβολή του όρου j στην καταχώρηση του πίνακα SCORE για το προφίλ P_3. Δηλαδή

  SCORE [P_3] = SCORE [P_3] + (βάρος j στο P_3) * (βάρος j στο D) =

  = 0.0511 + 0.72 * 0.90 = 0.0511 + 0.648 = 0.6991

  Η συμβολή των όρων του μη σημαντικού διανύσματος του προφίλ P_3 έχει ήδη προστεθεί επομένως ολοκληρώνεται και η επεξεργασία του συγκεκριμένου ζεύγους.

Οι τιμές του πίνακα SCORE που υπολογίστηκαν παραπάνω φαίνονται και στο Σχήμα 8. Συγκρίνοντας αυτές τις τιμές με τις αντίστοιχες τιμές του πίνακα THRESHOLD, καταλήγουμε ότι μόνο το προφίλ P_3 ταιριάζει στο κείμενο D, καθώς

SCORE [P_3] > THRESHOLD [P_3]

Παρατηρούμε ότι και σε αυτήν την περίπτωση, το πλήθος των προφίλ που εξετάζονται ως προς ένα κείμενο είναι σχετικά μικρό. Επομένως η απόδοση του συστήματος αυξάνεται.

2.3.3 Σύγκριση δομών ευρετηρίων των προφίλ

Στον παρακάτω πίνακα παρουσιάζονται κάποια πειραματικά αποτελέσματα [35, 36, 37], τα οποία παρέχουν τη δυνατότητα εξαγωγής συμπερασμάτων για την απόδοση των μεθόδων που περιγράφηκαν παραπάνω.

Τα προφίλ που χρησιμοποιούνται στα συγκεκριμένα πειράματα έχουν πλήθος 300000, ενώ καθένα από αυτά περιέχει κατά μέσο όρο 5 λέξεις. Το κάθε κείμενο αποτελείται κατά μέσο όρο από 12000 λέξεις, οι οποίες επιλέγονται από ένα λεξιλόγιο με 1.8 εκατομμύρια λέξεις. Οι μετρικές που μελετούνται είναι ο χώρος στο δίσκο που

καταλαμβάνει η κάθε δομή σε blocks (size) και ο αριθμός των εισόδων/ εξόδων που πραγματοποιούνται κατά την επεξεργασία ενός κειμένου (I/Os).

Μέθοδος	Size (Blocks)	I/Os
Brute Force (Boolean)	9668	9668
Μέτρησης	18000	2849
Τυχαίου Κλειδιού	18000	2849
Καταταγμένου Κλειδιού	16475	1139
Τυχαίου Δένδρου	18029	2841
Καταταγμένου Δένδρου	18029	1213
Brute Force (VSM)	23731	23731
PI	29275	4495
SPI	33670	3878

Σύμφωνα με αυτά τα αποτελέσματα, οι μέθοδοι Brute Force είναι οι καλύτεροι όσον αφορά τις απαιτήσεις σε χώρο, καθώς απαιτούν το μικρότερο χώρο στο δίσκο. Όσον αφορά τον αριθμό των blocks, τα οποία διαβάζονται κατά τη διαδικασία του «ταιριάσματος», οι μέθοδοι που χρησιμοποιούν ευρετήριο υπερτερούν των μεθόδων Brute Force. Ειδικά στην περίπτωση του Boolean μοντέλου, η μέθοδος Καταταγμένου Κλειδιού έχει την καλύτερη απόδοση σχετικά με τον αριθμό των I/Os ανά κείμενο, ενώ την αντίστοιχη θέση στην περίπτωση του Vector Space μοντέλου, κατέχει η μέθοδος SPI. Τέλος αξίζει να παρατηρήσουμε ότι οι μέθοδοι του Vector Space μοντέλου απαιτούν περισσότερο χώρο στο δίσκο και μεγαλύτερο αριθμό I/Os ανά κείμενο, καθώς αποθηκεύουν στις δομές και την πληροφορία του βάρους.

Κεφάλαιο ΙΙΙ

Προτεινόμενος αλγόριθμος για τη σύγκριση των προφίλ και των κειμένων

Σε αυτό το σημείο θα παρουσιάσουμε τον τρόπο αναπαράστασης των κειμένων και των προφίλ στο σύστημά μας, τη μέθοδο ευρετηρίασης των προφίλ που επιλέχθηκε, καθώς και τον αλγόριθμο επεξεργασίας ενός κειμένου απέναντι στη δομή των προφίλ.

3.1 Μοντέλα αναπαράστασης προφίλ και κειμένων στο σύστημα

Το σύστημά μας υποστηρίζει και τα δύο μοντέλα αναπαράστασης προφίλ και κειμένων, δηλαδή και το Boolean και το Vector Space μοντέλο. Το προφίλ ενός χρήστη αναπαρίσταται ως ένα σύνολο το οποίο περιέχει υπο – προφίλ. Τα υπο – προφίλ αυτά μπορεί να είναι είτε στη μορφή του Boolean μοντέλου είτε στη μορφή του Vector Space μοντέλου.

3.1.1 Αναπαράσταση κειμένων

Η αναπαράσταση των κειμένων είναι αυτή που περιγράφηκε παραπάνω και για τα δύο μοντέλα. Το μόνο στοιχείο που αξίζει να τονιστεί είναι ο τρόπος με τον οποίο υπολογίζονται τα βάρη των όρων στη διανυσματική αναπαράσταση των κειμένων.

Σε κάθε περίπτωση, ως συλλογή αναφοράς θεωρείται το σύνολο των κειμένων που βρίσκονται τη δεδομένη χρονική στιγμή στο σύστημα. Έτσι το βάρος ενός όρου κειμένου υπολογίζεται με βάση την παρακάτω φόρμουλα, η οποία βέβαια συμφωνεί απολύτως με τον προαναφερόμενο τύπο [38].

$$\text{Βάρος όρου t} = \begin{matrix}\text{Πλήθος εμφανίσεων} \\ \text{t στο κείμενο}\end{matrix} * \log_2\left(\frac{\text{Πλήθος κειμένων στη συλλογή}}{\text{Πλήθος κειμένων που περιέχουν t}}\right)$$

Στη συνέχεια εξετάζουμε τις ειδικές περιπτώσεις, στις οποίες η συγκεκριμένη φόρμουλα δίνει αόριστες ή γενικά μη επιθυμητές τιμές.

Αρχικά, εάν ο όρος δεν υπάρχει σε κανένα άλλο κείμενο, το κλάσμα της λογαριθμικής παράστασης δίνει αόριστη τιμή (διαίρεση με το μηδέν). Ωστόσο, ο συγκεκριμένος όρος πρέπει να έχει πολύ μεγάλο βάρος, καθώς θεωρείται σπάνιος για τη συλλογή. Για να αντιμετωπίσουμε αυτό το πρόβλημα, θέτουμε τη λογαριθμική παράσταση ίση με 1 και το βάρος του όρου t προκύπτει τελικά ίσο με:

$$\text{Βάρος όρου t} = \text{Πλήθος εμφανίσεων t στο κείμενο}$$

Επιπλέον, εάν ο όρος εμφανίζεται σε όλα τα κείμενα της συλλογής, η λογαριθμική παράσταση ισούται με 0, οπότε ο όρος προκύπτει να έχει μηδενικό βάρος. Σε αυτήν την περίπτωση, ο όρος πρέπει να έχει κάποιο μικρό (αλλά όχι μηδενικό) βάρος, καθώς είναι πολύ συνηθισμένος. Αυτό επιτυγχάνεται, θέτοντας την παράσταση του λογαρίθμου ίση με 0.1, οπότε τελικά το βάρος του όρου t ισούται με:

$$\text{Βάρος όρου t} = (\text{Πλήθος εμφανίσεων t στο κείμενο}) * 0.1$$

3.1.2 Αναπαράσταση προφίλ Vector Space μοντέλου

Τα υπο – προφίλ του Vector – Space μοντέλου αναπαρίστανται όπως περιγράφηκε παραπάνω. Το μόνο στοιχείο που αξίζει να τονιστεί είναι ο τρόπος με τον οποίο υπολογίζονται τα βάρη των όρων των υπο – προφίλ, τα οποία εκφράζονται με βάση το μοντέλο Vector Space.

Όπως και στην περίπτωση των κειμένων, ως συλλογή αναφοράς θεωρείται το σύνολο των κειμένων που βρίσκονται τη δεδομένη χρονική στιγμή στο σύστημα. Έτσι το βάρος ενός όρου t υπο – προφίλ υπολογίζεται με βάση την παρακάτω φόρμουλα, η οποία βέβαια συμφωνεί απολύτως με τον προαναφερόμενο τύπο [38].

$$\text{Βάρος όρου t} = \begin{matrix}\text{Πλήθος εμφανίσεων} \\ \text{t στο υπο - προφίλ}\end{matrix} * \log_2\left(\frac{\text{Πλήθος κειμένων στη συλλογή}}{\text{Πλήθος κειμένων που περιέχουν t}}\right)$$

Ειδικά στην περίπτωση που κάποιος όρος t δεν εμφανίζεται σε κανένα κείμενο, η λογαριθμική παράσταση λαμβάνει αόριστη τιμή (διαίρεση με το μηδέν). Εφόσον όμως ο όρος είναι σπάνιος, πρέπει να έχει υψηλή τιμή βάρους. Για αυτό το λόγο θέτουμε την παράσταση του λογαρίθμου ίση με 1, οπότε το βάρος του όρου t γίνεται:

$$\text{Βάρος όρου t} = \text{Πλήθος εμφανίσεων t στο υπο – προφίλ}$$

Από την άλλη πλευρά, στην περίπτωση που ο όρος t εμφανίζεται σε όλα τα κείμενα του συστήματος, η λογαριθμική παράσταση ισούται με μηδέν. Ο συγκεκριμένος όρος πρέπει ως συνήθης όρος στο σύστημα να έχει μικρό βάρος, αλλά όχι μηδενικό. θέτοντας την παράσταση του λογαρίθμου ίση με 0.1, το βάρος του όρου t γίνεται αρκετά μικρό και ισούται με:

$$\text{Βάρος όρου t} = (\text{Πλήθος εμφανίσεων t στο υπο – προφίλ}) * 0.1$$

3.1.3 Αναπαράσταση προφίλ Boolean μοντέλου

θεωρούμε ότι το κάθε υπο – προφίλ του Boolean μοντέλου είναι μία σειρά από k διακριτές λέξεις (w_1, w_2, …, w_k). Ένα υπο – προφίλ ταιριάζει ένα κείμενο, εάν όλες οι λέξεις του εμφανίζονται στο κείμενο. Με βάση τη συγκεκριμένη θεώρηση όλα τα υπο – προφίλ πρέπει να μετασχηματιστούν σε συζευκτικά προφίλ. Λαμβάνοντας υπόψη τις μορφές που μπορεί να έχει ένα υπο – προφίλ (με ή χωρίς τελεστές εγγύτητας), εξετάζουμε, στη συνέχεια, εάν αυτό μπορεί να εκφραστεί σαν μία επερώτηση συζεύξεων.

- w. Το υπο – προφίλ αυτής της μορφής δεν έχει κανένα τελεστή, άρα είναι σε συζευκτική μορφή. Για παράδειγμα το προφίλ P = «a» μπορεί να εκφραστεί ως P = (a).

- Q1 AND Q2. Το υπο – προφίλ αυτής της μορφής είναι σε συζευκτική μορφή. Για παράδειγμα Το προφίλ P = «a AND b AND c» μπορεί να εκφραστεί ως P = (a, b, c).
- Q1 OR Q2. Δε μπορούν να εμφανιστούν υπο - προφίλ αυτής της μορφής.
- Q1 AND NOT Q2. Τα υπο – προφίλ αυτής της μορφής εκφράζονται σαν συζευκτικά προφίλ, αγνοώντας τον τελεστή άρνησης. Για να δείξουμε ότι κάποιος όρος είναι σε μορφή άρνησης χρησιμοποιούμε ένα bit, το οποίο λαμβάνει τις τιμές 0, 1 αν ο όρος είναι θετικός ή αρνητικός αντίστοιχα. Για παράδειγμα το προφίλ P = «a AND NOT b» μπορεί να εκφραστεί ως P = (a, $\neg$b). Σημειώνουμε ότι στο σύστημά μας δεν είναι αποδεκτά τα υπο – προφίλ του τύπου ($\neg$b), καθώς τέτοιας μορφής ικανοποιούνται από ένα μεγάλο πλήθος κειμένων και τελικά δε θα έχουν καμία αξία για τον τελικό χρήστη.
- Q1 $<_{[l,\ u]}$ Q2. Τα υπο – προφίλ αυτής της μορφής εκφράζονται σαν συζευκτικές επερωτήσεις, αγνοώντας τον τελεστή εγγύτητας. Ο τελεστής εγγύτητας λαμβάνεται υπόψη μετά την εύρεση των κειμένων που ταιριάζουν στο προφίλ και πριν την αποστολή τους στο χρήστη. Για παράδειγμα το προφίλ P = «a $<_{[l,\ u]}$ b» εκφράζεται σαν P = (a, b). Πιο συγκεκριμένα, μετά την εύρεση του κειμένου που ταιριάζει στο προφίλ, πραγματοποιείται κάποια επεξεργασία, η οποία ελέγχει εάν ικανοποιούνται τα κριτήρια εγγύτητας του προφίλ. Η επεξεργασία αυτή έχει σαν προϋπόθεση την ύπαρξη offsets στους όρους της αναπαράστασης του κειμένου. Με βάση αυτά τα offsets είναι δυνατό να ελέγξουμε τα κριτήρια εγγύτητας.

Καταλήγουμε λοιπόν στο συμπέρασμα ότι όλα τα υπο - προφίλ του Boolean μοντέλου μπορούν να εκφρασθούν με τη μορφή (w_1, w_2, …, w_k). Αυτό το γεγονός μας παρέχει τη δυνατότητα ανάπτυξης μίας κοινής μεθόδου επεξεργασίας όλων των συγκεκριμένων υπο – προφίλ ανεξαρτήτως μορφής. Η μόνη διαφορά που παρουσιάζεται είναι η προσθήκη κάποιων ελέγχων στην περίπτωση που υπάρχει στο προφίλ τελεστής άρνησης ή τελεστής εγγύτητας.

3.2 Δομή ευρετηρίου προφίλ στο σύστημα

Στη συνέχεια περιγράφεται η δομή ευρετηρίου των προφίλ που χρησιμοποιήθηκε στο σύστημά μας και ο αλγόριθμος επεξεργασίας ενός κειμένου έναντι της συγκεκριμένης δομής.

Όπως και στις δομές που προαναφέρθηκαν, κάθε όρος σχετίζεται με μία ανάστροφη λίστα από εγγραφές (postings) που περιέχει κάποια από τα προφίλ που αναφέρουν τον όρο. Σε κάθε εγγραφή, εάν το προφίλ ανήκει στο Vector Space μοντέλο, αναφέρεται κανονικά το βάρος κάθε όρου. Διαφορετικά εάν το προφίλ είναι Boolean, αναφέρεται το βάρος 1 σε κάθε θετικό όρο και το βάρος –1 σε κάθε αρνητικό.

Επιπλέον χρησιμοποιούνται δύο πίνακες, οι THRESHOLD και SCORE, οι οποίοι διατηρούν μία καταχώρηση για κάθε προφίλ. Για κάθε προφίλ, ο πίνακας THRESHOLD αποθηκεύει την ελάχιστη βαθμολογία που πρέπει να έχει το προφίλ απέναντι στο κείμενο για να ταιριάζουν. Για ένα προφίλ στο Boolean μοντέλο, αυτό το ελάχιστο ισούται με το πλήθος των θετικών όρων. Για ένα προφίλ στο Vector Space μοντέλο, αυτό το ελάχιστο καθορίζεται από το χρήστη και καλείται κατώφλι σχετικότητας (relevance threshold). Ο πίνακας SCORE διατηρεί τη βαθμολογία (score) που έχει κάθε προφίλ απέναντι στο κείμενο.

Όταν εμφανίζεται ένα κείμενο, διαμορφώνουμε τη διανυσματική του αναπαράσταση και αρχικοποιούμε τις καταχωρήσεις του πίνακα SCORE σε 0. Η επεξεργασία για τα προφίλ του Vector Space μοντέλου είναι η ίδια με την SPI μέθοδο. Για τα προφίλ του Boolean μοντέλου ακολουθούμε την ίδια διαδικασία, με τη μόνη διαφορά ότι προσθέτουμε στον πίνακα SCORE μόνο τη συμβολή των θετικών όρων, δηλαδή των όρων που έχουν βάρος 1. Μετά την κατάλληλη επεξεργασία και τον υπολογισμό των καταχωρήσεων του πίνακα SCORE, τα προφίλ που έχουν τιμή στον πίνακα SCORE μεγαλύτερη ή ίση με την αντίστοιχη τιμή στον πίνακα THRESHOLD, ταιριάζουν το κείμενο.

3.3 Αλγόριθμος επεξεργασίας ενός κειμένου έναντι της δομής ευρετηρίου των προφίλ

Στο Σχήμα 9 φαίνεται ένα παράδειγμα στο οποίο χρησιμοποιείται η συγκεκριμένη μέθοδος. Τα προφίλ που υπάρχουν στο σύστημα είναι

- Προφίλ του μοντέλου Boolean

  B1 = (a, b)

  B2 = (a, d)

  B3 = (a, d, e)

  B4 = (b, f)

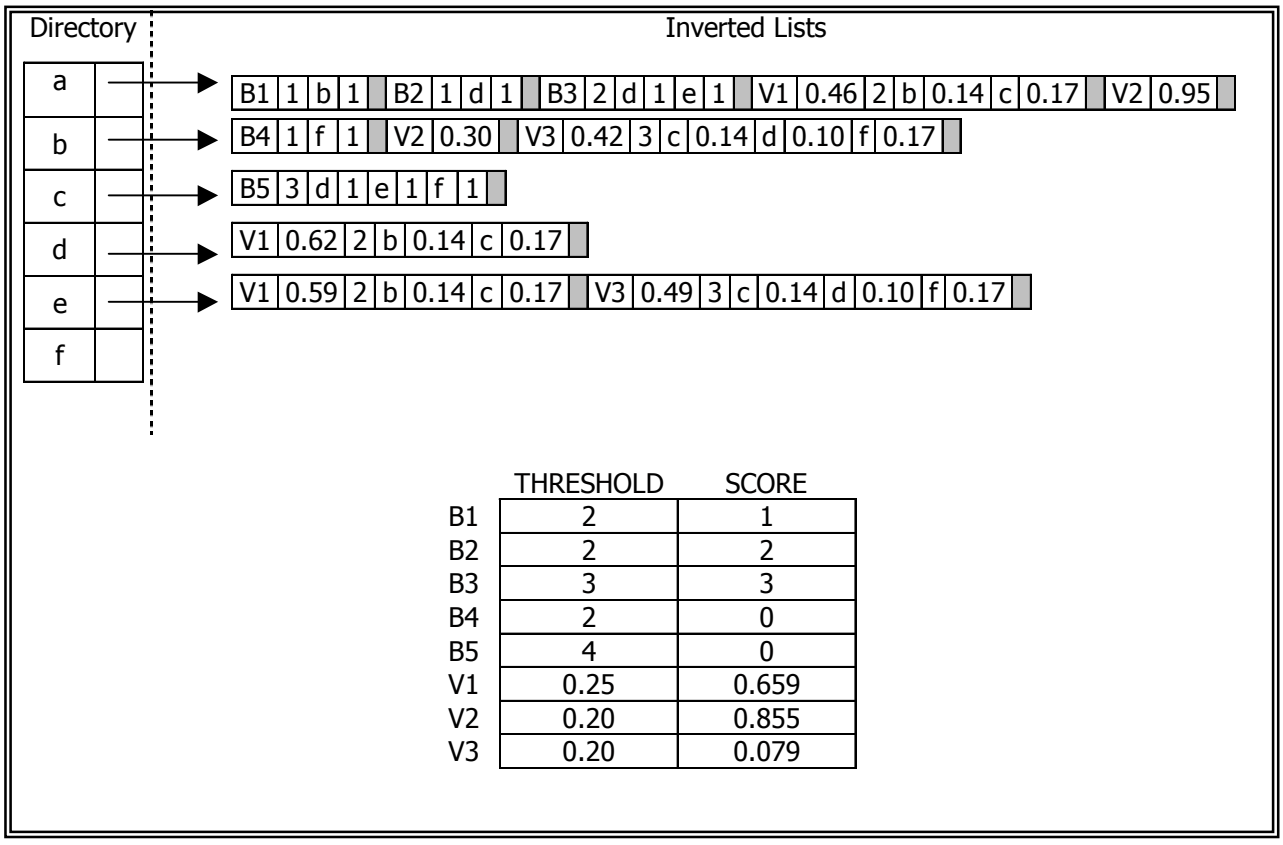
  B5 = (c, d, e, f)

- Προφίλ του μοντέλου Vector Space

  V1 = <(a, 0.46), (b, 0.14), (c, 0.17), (d, 0.62), (e, 0.59)>, με θ1 = 0.25

  V2 = <(a, 0.95), (b, 0.30)>, με θ2 = 0.20

  V3 = <(b, 0.42), (c, 0.14), (d, 0.10), (e, 0.49), (f, 0.17)>, με θ3 = 0.20



	THRESHOLD	SCORE
B1	2	1
B2	2	2
B3	3	3
B4	2	0
B5	4	0
V1	0.25	0.659
V2	0.20	0.855
V3	0.20	0.079

Σχήμα 9. Δομή ευρετηρίου για τα προφίλ του συστήματός μας

Υποθέτουμε ότι η λέξη a έχει τη μικρότερη συχνότητα εμφάνισης, στη συνέχεια η λέξη b κτλ. Έτσι, τα προφίλ B1, B2 και B3 εισάγονται στο posting της λέξης a, το προφίλ B4 εισάγεται στο posting της λέξης b και το προφίλ B5 εισάγεται στο posting της λέξης c. Για τα προφίλ που είναι εκφρασμένα στο Vector Space μοντέλο, βρίσκουμε τα περισσότερο μη σημαντικά υποδιανύσματα, ακολουθώντας τη διαδικασία που περιγράφτηκε αναλυτικά στη μέθοδο SPI. Για το προφίλ V1, αυτό το υποδιάνυσμα είναι το ((b, 0.14), (c, 0.17)). Το προφίλ V2 δεν έχει μη σημαντικούς όρους. Για το προφίλ V3, το περισσότερο μη σημαντικό υποδιάνυσμα είναι το ((c, 0.14), (d, 0.10), (f, 0.17)).

Έστω ότι εμφανίζεται το κείμενο D (a, d, e), του οποίου η αναπαράσταση με βάρη είναι D <(a, 0.9), (d, 0.3), (e, 0.1)>. Ο πίνακας SCORE αρχικοποιείται με μηδενικά. Διαδοχικά, επεξεργαζόμαστε όλα τα ζεύγη του διανύσματος του κειμένου:

- Επεξεργασία ζεύγους (a, 0.9)
  Στη λίστα του όρου a υπάρχουν τα προφίλ B1, B2, B3, V1 και V2.
  - Για το προφίλ B1, ο όρος b δεν υπάρχει στο διάνυσμα του κειμένου, οπότε μόνο ο όρος a συμβάλει στο score. Επομένως,
    SCORE [B1] = 1
  - Για το προφίλ B2, οι όροι d και e υπάρχουν στο διάνυσμα του κειμένου. Επομένως όλοι οι όροι του προφίλ συμβάλουν στο score, δηλαδή
    SCORE [B3] = 3
  - Για το προφίλ B3, ο όρος d υπάρχει στο διάνυσμα του κειμένου. Επομένως όλοι οι όροι του προφίλ συμβάλουν στο score, δηλαδή
    SCORE [B2] = 2
  - Για το προφίλ V1, οι μη σημαντικοί όροι δεν υπάρχουν στο διάνυσμα του κειμένου. Επομένως,
    SCORE [V1] = (βάρος a στο D) * (βάρος a στο V1) = 0.9 *0.46 = 0.414
  - Για το προφίλ V2 έχουμε
    SCORE [V2] = (βάρος a στο D) * (βάρος a στο V2) = 0.9 * 0.95 = 0.855
- Επεξεργασία ζεύγους (d, 0.3)
  Στη λίστα του όρου d υπάρχει μόνο το προφίλ V1.
  - Για το προφίλ V1, προσθέτουμε τη συμβολή του όρου d, δηλαδή

SCORE [V1] = SCORE [V1] + (βάρος d στο D) * (βάρος d στο V1) = 0.3 * 0.62 = 0.414 + 0.186 = 0.6

- Επεξεργασία ζεύγους (e, 0.1)

  Στη λίστα του όρου e υπάρχουν τα προφίλ V1 και V3.

  - Για το προφίλ V1, προσθέτουμε τη συμβολή του όρου e, δηλαδή
    SCORE [V1] = SCORE [V1] + (βάρος e στο D) * (βάρος e στο V1) = 0.6 + 0.1 * 0.59 = 0.6 * 0.059 = 0.659
  - Για το προφίλ V3, προσθέτουμε αρχικά τη συμβολή του όρου e, δηλαδή
    SCORE [V3] = (βάρος e στο D) * (βάρος e στο V3) = 0.1 * 0.49 = 0.049
    Από τους μη σημαντικούς όρους, μόνο ο d υπάρχει στο διάνυσμα του κειμένου, οπότε το σκορ του συγκεκριμένου προφίλ γίνεται
    SCORE [V3] = SCORE [V3] + (βάρος d στο D) * (βάρος d στο V3) = 0.049 + 0.3 * 0.1 = 0.049 + 0.03 = 0.079.

Σε αυτό το σημείο η επεξεργασία ολοκληρώνεται. Τα σκορ όλων των προφίλ ως προς το κείμενο D φαίνονται και στο Σχήμα 9. Τα προφίλ που ταιριάζουν στο κείμενο είναι αυτά που έχουν τιμή στην καταχώρησή τους στον πίνακα SCORE μεγαλύτερη ή ίση με την αντίστοιχη τιμή στον πίνακα THRESHOLD. Αυτά τα προφίλ είναι τα B2, B3, V1 και V2.

Η συγκεκριμένη μέθοδος συνδυάζει τα πλεονεκτήματα των μεθόδων Key και SPI με την έννοια ότι το πλήθος των προφίλ που εξετάζονται ανά κείμενο είναι μειωμένο. Επιπλέον έχει το βασικό πλεονέκτημα ότι εφαρμόζει μία κοινή μέθοδο επεξεργασίας των προφίλ των δύο μοντέλων.

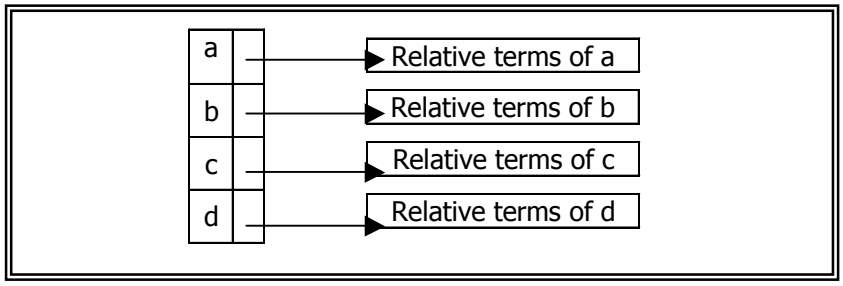
3.4 Εισαγωγή σημασιολογίας στο σύστημα

Συχνά οι χρήστες χρησιμοποιούν πολλές διαφορετικές λέξεις για να περιγράψουν την ίδια έννοια, λέξεις που χαρακτηρίζονται ως συνώνυμα. Ένα σύστημα διασποράς πληροφορίας οφείλει να λαμβάνει υπόψη του αυτό το γεγονός, με σκοπό πάντα να παρέχει μεγαλύτερη ακρίβεια στους χρήστες. Επομένως ένα τέτοιο σύστημα πρέπει να περιλαμβάνει μηχανισμούς που διαχειρίζονται τα συνώνυμα.

Πέρα από τα συνώνυμα, ένα σύστημα διασποράς πληροφορίας οφείλει να διαχειριστεί και όρους οι οποίοι είναι ευρύτεροι ή ειδικότεροι όροι κάποιων άλλων όρων. Όταν κάποιος χρήστης αναζητά πληροφορία σχετικά με μία γενική έννοια, τότε σίγουρα τον ενδιαφέρουν κείμενα, τα οποία αναφέρονται σε κάποιες έννοιες ειδικότερες από αυτήν. Για παράδειγμα ο χρήστης που στέλνει το προφίλ «Μηχανοκίνητο όχημα» στο σύστημα, ενδιαφέρεται για κείμενα που περιέχουν τον ειδικότερο όρο «Αυτοκίνητο».

3.4.1 Δομή σχετικών όρων

Για τη διαχείριση και επεξεργασία των σχετικών όρων έχει υλοποιηθεί μία ειδική δομή. Αυτή η δομή, η οποία παρουσιάζεται στο Σχήμα 10, διατηρεί για κάθε όρο τους σχετικούς του όρους, δηλαδή τα συνώνυμά του, τους ειδικότερους όρους του, οι οποίοι καλούνται υπώνυμα (hyponyms) και τους γενικότερους όρους του, οι οποίοι καλούνται υπερώνυμα (hypernyms).



Σχήμα 10. Δομή σχετικών όρων

Η δημιουργία της δομής στηρίζεται σε ένα θησαυρό (thesaurus). Όταν το σύστημα αναζητά τους σχετικούς όρους ενός όρου προφίλ ή κειμένου, ρωτά το θησαυρό για τα συνώνυμα, τα υπερώνυμα και τα υπώνυμα του όρου. Η απάντηση του θησαυρού είναι αυτή που αποθηκεύεται στη δομή των σχετικών όρων.

3.4.2 Ανάκτηση σχετικών όρων

Για την ανάκτηση των σχετικών όρων των όρων που εμφανίζονται στο σύστημα από το θησαυρό έχουν αναπτυχθεί δύο διαφορετικές μεθοδολογίες. Στην

πρώτη μεθοδολογία, η ερώτηση προς το θησαυρό πραγματοποιείται όταν εμφανίζεται κάποιο κείμενο ή προφίλ στο σύστημα. Τότε ελέγχονται όλοι οι όροι του κειμένου ή προφίλ. Εάν κάποιος όρος δεν εμφανίζεται στο hash table της δομής σχετικών όρων, το σύστημα ρωτάει το θησαυρό και αποθηκεύει την απάντηση στη δομή. Έτσι οποιαδήποτε χρονική στιγμή αναζητηθούν οι σχετικοί όροι του όρου, αυτοί ανακτώνται από τη δομή σχετικών όρων.

Στη δεύτερη μεθοδολογία, δε χρησιμοποιείται η δομή σχετικών όρων. Το σύστημα ανακτά την πληροφορία για τους σχετικούς όρους τη χρονική στιγμή που τη χρειάζεται, οπότε δεν υπάρχει κάποιος λόγος να αποθηκευτεί αυτή η πληροφορία για μετέπειτα χρήση. Ποιο συγκεκριμένα η ερώτηση προς το θησαυρό πραγματοποιείται κατά τη διαδικασία του «ταιριάσματος». Δηλαδή το σύστημα ρωτάει το θησαυρό για κάθε όρο που συναντάει κατά την επεξεργασία ενός κειμένου απέναντι στο ευρετήριο των προφίλ, χρησιμοποιεί την απάντηση όπως θα εξηγηθεί παρακάτω και δεν την αποθηκεύει.

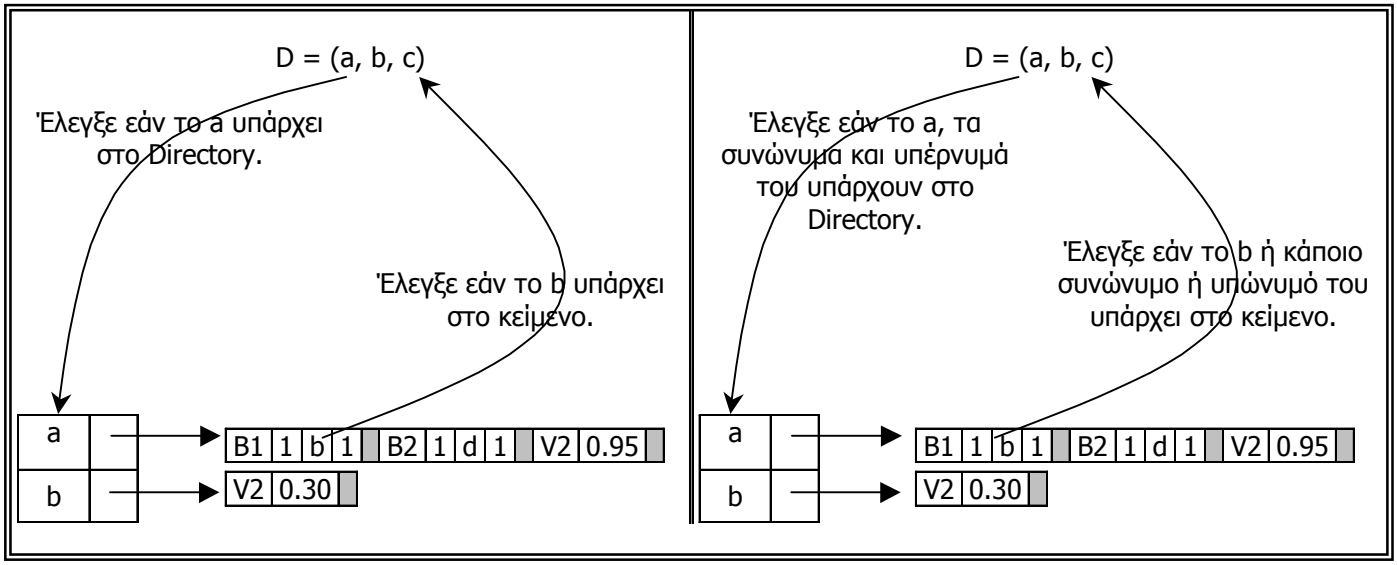
3.4.3 Αλγόριθμος επεξεργασίας ενός κειμένου έναντι της δομής ευρετηρίου των προφίλ με χρήση σημασιολογίας

Η εισαγωγή σημασιολογίας στο σύστημά μας επιφέρει κάποιες μεταβολές στην επεξεργασία ενός κειμένου ως προς το ευρετήριο των προφίλ. Η διαδικασία «ταιριάσματος» που περιγράφηκε πριν ελέγχει την ταυτοσημότητα ανάμεσα στους όρους του κειμένου και των προφίλ, δηλαδή εάν όλοι (στην περίπτωση του Boolean μοντέλου) ή κάποιοι όροι (στην περίπτωση του Vector Space μοντέλου) του προφίλ υπάρχουν στο κείμενο. Έχοντας εισάγει σημασιολογία στο σύστημά μας, ελέγχουμε επιπλέον εάν κάποιο από τα συνώνυμα ή τα υπώνυμα των όρων του προφίλ υπάρχουν στο κείμενό μας. Επομένως τελικά ένα προφίλ ταιριάζει σε ένα κείμενο εάν οι όροι του ή κάποιο από τα συνώνυμα/ υπώνυμα των όρων του υπάρχουν στο κείμενο.

Ως αποτέλεσμα στη διαδικασία ταιριάσματος πραγματοποιούνται δυο ουσιαστικές μεταβολές. Η πρώτη αλλαγή αναφέρεται στο σημείο που κάθε όρος του κειμένου ελέγχεται εάν υπάρχει στο ευρετήριο (directory) και εάν ναι, τότε ανακτάται η ανάστροφη λίστα του. Σε αυτήν την περίπτωση δεν ελέγχουμε μόνο τον όρο του κειμένου, αλλά και όλα τα συνώνυμα και υπερώνυμά του, τα οποία ανακτούμε από τη

δομή ευρύτερων όρων ή απευθείας από το θησαυρό. Η δεύτερη αλλαγή αναφέρεται στο σημείο που ελέγχουμε εάν ο όρος του προφίλ, που υπάρχει στις εγγραφές (postings) του ευρετηρίου, υπάρχει στο κείμενο και εάν ναι προσθέτουμε τη συμβολή του στο σκορ του προφίλ. Σε αυτήν την περίπτωση δεν ελέγχουμε μόνο τον όρο αλλά και τα συνώνυμα και τα υπώνυμά του, τα οποία επίσης ανακτούμε από τη δομή ευρύτερων όρων ή το θησαυρό.

Στο Σχήμα 11 παρουσιάζονται οι μεταβολές που πραγματοποιούνται στο σύστημα εισάγοντας τη σημασιολογία. Αριστερά φαίνεται η διαδικασία ταιριάσματος όπως ήταν πριν. Δεξιά παρουσιάζεται η ίδια διαδικασία με τις δύο μεταβολές που πραγματοποιήθηκαν. Αξίζει να παρατηρήσουμε ότι στις δύο μεταβολές που πραγματοποιούνται, αντικαθίσταται ουσιαστικά ο όρος με ένα διάνυσμα, το οποίο στην πρώτη περίπτωση περιέχει τον όρο, τα συνώνυμα και τα υπερώνυμά του και στη δεύτερη τον όρο, τα συνώνυμα και τα υπώνυμά του.



Σχήμα 11. Μεταβολή στη διαδικασία επεξεργασίας κειμένου

Όπως είναι προφανές, τα συνώνυμα, υπώνυμα, υπερώνυμα ενός όρου διατηρούν τα χαρακτηριστικά του όρου. Πιο συγκεκριμένα, στην περίπτωση του μοντέλου Boolean, εάν ο όρος είναι αρνητικός (ακολουθεί του τελεστή AND NOT), τότε και οι σχετικοί του όροι θεωρούνται αρνητικοί. Επιπλέον, εάν ο όρος λαμβάνει μέρος σε μία συνθήκη εγγύτητας, τότε και οι σχετικοί του όροι λαμβάνουν μέρος σε αυτήν. Για την περίπτωση του Vector Space μοντέλου, πρέπει να παρατηρήσουμε ότι οι σχετικοί όροι διατηρούν το ίδιο βάρος με τον όρο από οποίο προέρχονται.

Αποτέλεσμα της συγκεκριμένης διαδικασίας είναι η αποστολή κειμένων στους χρήστες τα οποία περιέχουν πέρα από τους όρους του προφίλ του και συνώνυμα ή υπώνυμα των όρων αυτών. Έτσι οι χρήστες λαμβάνουν κείμενα, τα οποία ίσως ποτέ να μην είχαν λάβει με μία κλασσική υπηρεσία ανάκτησης πληροφορίας, ενώ στην πραγματικότητα τα κείμενα αυτά ικανοποιούν τα ενδιαφέροντά τους.

Τέλος, παρουσιάζουμε τον αλγόριθμο ταιριάσματος σε μορφή ψευδοκώδικα, ο οποίος υπολογίζει το σκορ για τα προφίλ του συστήματος ως προς ένα κείμενο:

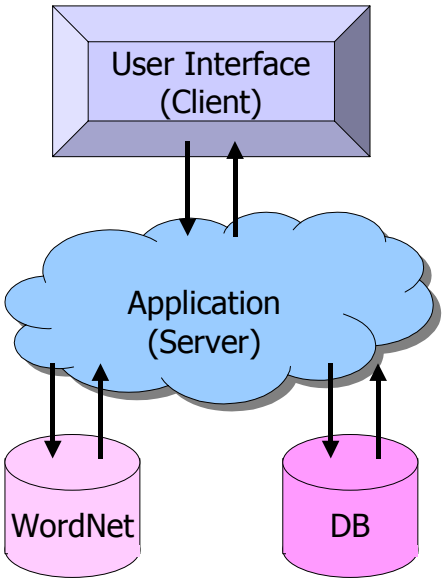
```
Για κάθε όρο t του κειμένου d
{
 Φτιάξε το διάνυσμα V = {t, συνώνυμα του t, υπέρνυμα του t}
 Για κάθε στοιχείο v του V
  Εάν v υπάρχει στο directory του ευρετηρίου των προφίλ
   Για κάθε posting της λίστας του v
   {
    Εάν (score του υπο – προφίλ του posting == 0)
    {
     // υπολόγισε το score της λέξης στο hash table
     Εάν BM
      Score = 1
     Εάν VSM
      Score = βάρος του t * βάρος του v

     // υπολόγισε το score των λέξεων στο posting
     Για κάθε όρο x στο posting
     {
      Φτιάξε το διάνυσμα R = {x, συνώνυμα του x, υπώνυμα του x}
      Εάν κάποιο από τα στοιχεία του R, έστω το r, υπάρχει στο d
      {
         Εάν BM
          Score = Score + βάρος του x;
         Εάν VSM
          Score = Score + βάρος του r * βάρος του x
      }
     }
    }
    αλλιώς    // score!= 0, μόνο για VSM
     // υπολόγισε το score της λέξης στο hash table
     Score = Score + βάρος του t * βάρος του v

    Κάνε update το score του υπο - προφίλ στον πίνακα Score
   }
}
```

Κεφάλαιο IV

Σύστημα

Το σύστημά μας αποτελείται από τα τέσσερα μέρη που φαίνονται στο Σχήμα 12. Βασικότερο και κεντρικό συστατικό του συστήματος αποτελεί ο server, ο οποίος είναι η εφαρμογή που διατηρεί όλη τη λογική. Ο χρήστης επικοινωνεί με τον client μέσω ενός Web browser. Ο server επικοινωνεί τόσο με τον client από τον οποίο κυρίως δέχεται τα προφίλ των χρηστών, με τον θησαυρό WordNet από τον οποίο ανακτά τους σχετικούς όρους και μία βάση δεδομένων, στην οποία αποθηκεύονται τα κείμενα και τα προφίλ των χρηστών.

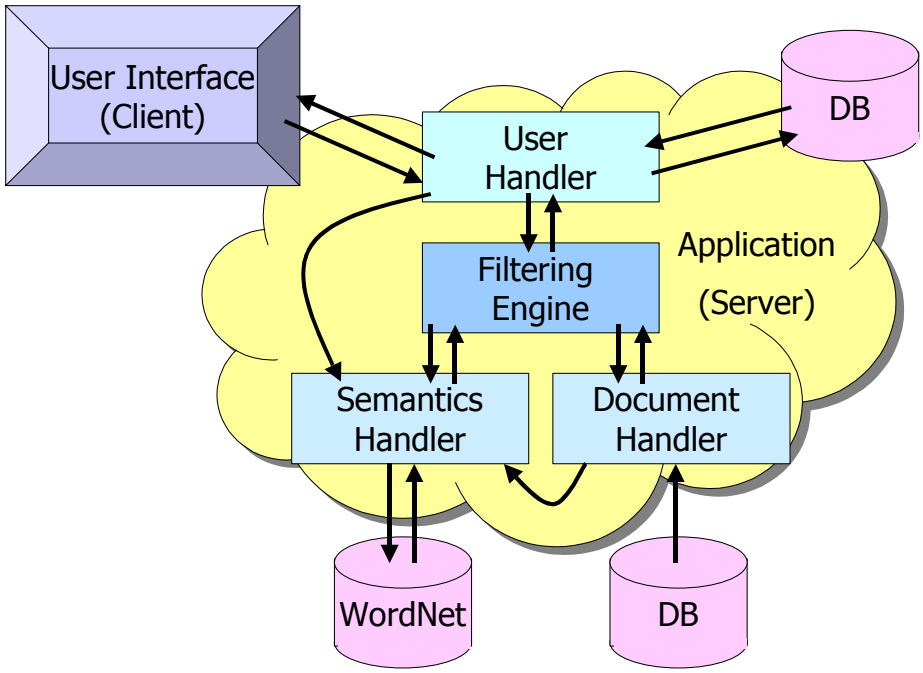
Στη συνέχεια παρουσιάζονται αναλυτικά τα τέσσερα μέρη του συστήματος.



Σχήμα 12. Γενική εικόνα συστήματος

5.1 Εφαρμογή (Server)

Το βασικό κομμάτι του συστήματος αποτελεί η εφαρμογή που φαίνεται στο κέντρο στο Σχήμα 13. Αυτό είναι το συστατικό μέρος που διατηρεί όλη τη λογική και που επικοινωνεί με όλα τα υπόλοιπα μέρη. Ο server αποτελείται από τέσσερα διαφορετικά μέρη, καθένα από τα οποία έχει τη δική του συμβολή και λειτουργία στο σύστημα.



Σχήμα 13. Server του συστήματος

5.1.1 Χειριστής χρηστών (User Handler)

Στο πάνω μέρος του σχήματος φαίνεται ο χειριστής των χρηστών (user handler). Αυτό το συστατικό είναι υπεύθυνο για τη διαχείριση των χρηστών και των προφίλ τους στο σύστημα. Διατηρεί μία λίστα με τους χρήστες και μία λίστα με υπο – προφίλ για τον κάθε χρήστη. Ο χειριστής των χρηστών λαμβάνει αρχικά το προφίλ από τη διεπαφή χρήσης και υλοποιεί την απαραίτητη επεξεργασία στο προφίλ του χρήστη. Στη συνέχεια εισάγει το χρήστη με το προφίλ του στη λίστες που διατηρεί

και αποθηκεύει αυτά τα στοιχεία στη βάση δεδομένων (εάν αυτό βέβαια είναι επιθυμητό). Τέλος παραδίδει το προφίλ στη μηχανή φιλτραρίσματος (filtering engine). Ειδικά στην περίπτωση που χρησιμοποιείται η δομή των σχετικών όρων, ο χειριστής των χρηστών αποστέλλει τους όρους του προφίλ στο σημασιολογικό χειριστή (semantics handler). Όταν η μηχανή φιλτραρίσματος ανακοινώσει στο χειριστή των χρηστών τα κείμενα που ταιριάζουν σε κάθε χρήστη, αυτός είναι υπεύθυνος να ενημερώσει το χρήστη για αυτά τα κείμενα.

5.1.2 Χειριστής κειμένων (Document Handler)

Ο χειριστής των κειμένων (document handler) είναι αυτός που διαχειρίζεται τα κείμενα στο σύστημα. Διατηρεί μία λίστα με τα κείμενα που υπάρχουν στο σύστημα και μία λίστα με τους βαθμούς των όρων που έχουν εμφανιστεί στα κείμενα. Ανακτά από τη βάση δεδομένων τα κείμενα, υλοποιεί την επεξεργασία των κειμένων, τα εισάγει στη λίστα του και τέλος τα παραδίδει στη μηχανή φιλτραρίσματος. Ειδικά στην περίπτωση που χρησιμοποιείται η δομή των σχετικών όρων, ο χειριστής των κειμένων αποστέλλει τους όρους του κειμένου στο σημασιολογικό χειριστή (semantics handler).

5.1.3 Σημασιολογικός χειριστής (Semantics Handler)

Ο σημασιολογικός χειριστής (semantics handler) είναι υπεύθυνος για την επικοινωνία με το θησαυρό. Διατηρεί τη δομή των σχετικών όρων. Δέχεται όρους από τον χειριστή των χρηστών και το χειριστή των κειμένων στην περίπτωση που χρησιμοποιείται η δομή των σχετικών όρων και από τη μηχανή φιλτραρίσματος στην αντίθετη περίπτωση. Για κάθε όρο αποστέλλει μία αίτηση στο θησαυρό. Στη συνέχεια λαμβάνει την απάντηση του θησαυρού και την αποθηκεύει στη δομή των σχετικών όρων (εάν αυτό είναι επιθυμητό). Τέλος, ο σημασιολογικός χειριστής αποστέλλει τους ανάλογους σε κάθε περίπτωση σχετικούς όρους στη μηχανή φιλτραρίσματος, μετά βέβαια από αίτηση της μηχανής.

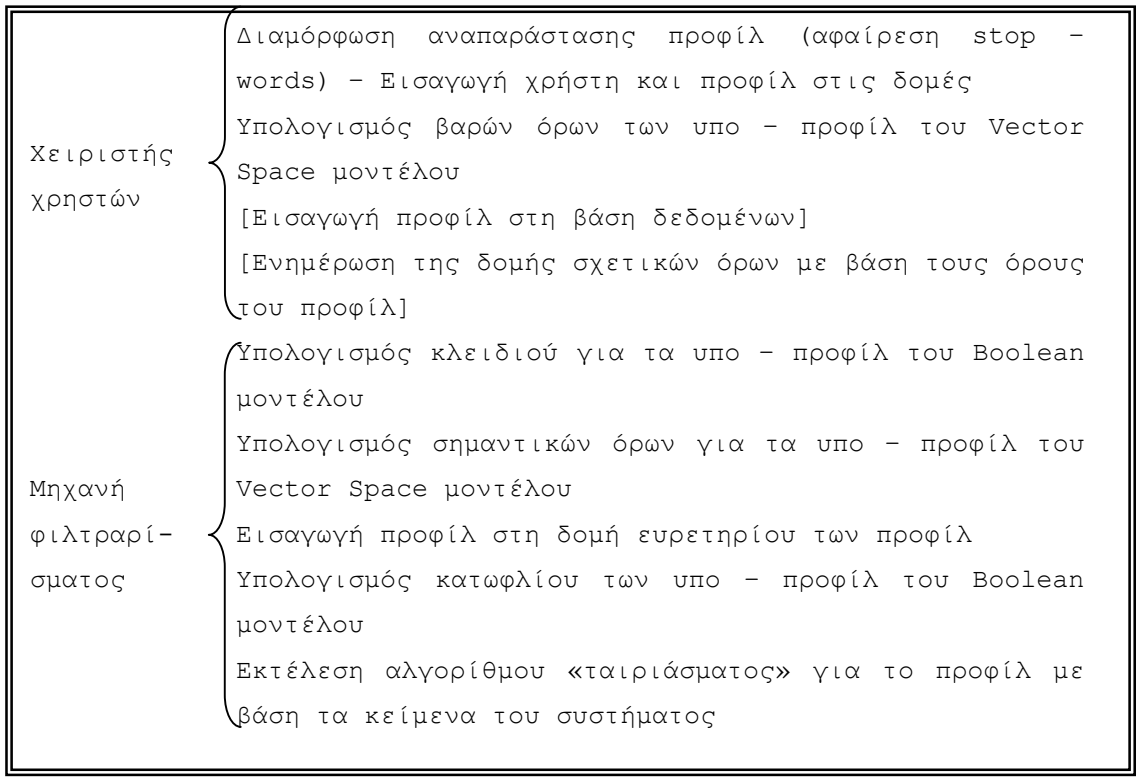
5.1.4 Μηχανή φιλτραρίσματος (Filtering Engine)

Η μηχανή φιλτραρίσματος είναι υπεύθυνη για τη διαδικασία του «ταιριάσματος» (matching). Διατηρεί το ευρετήριο των προφίλ των χρηστών. Κάθε φορά που ο χειριστής χρηστών αποστέλλει ένα προφίλ στη μηχανή, η μηχανή εκτελεί τον αλγόριθμο που περιγράφηκε παραπάνω και αποφασίζει ποια από τα κείμενα του συστήματος ταιριάζουν στο προφίλ. Παράλληλα κάθε φορά που ο χειριστής κειμένων αποστέλλει ένα κείμενο στη μηχανή, η μηχανή εκτελεί τον ίδιο αλγόριθμο και αποφασίζει ποια από τα προφίλ που υπάρχουν στο σύστημα ταιριάζουν στο κείμενο. Τέλος ο μηχανή αναζήτησης ενημερώνει τον χειριστή των χρηστών σχετικά με τα κείμενα που ταιριάζουν στο κάθε προφίλ.

5.1.5 Λειτουργία εφαρμογής

Στην έναρξη της λειτουργίας του συστήματος, ο server δεν περιέχει κανένα κείμενο και κανένα προφίλ, και επομένως δε διατηρεί καμία δομή. Όλες οι δομές δημιουργούνται δυναμικά με την εισαγωγή προφίλ και κειμένων στο σύστημά μας.

Καθώς εμφανίζεται ένα προφίλ χρήστη στο σύστημα, εκτελείται μία σειρά λειτουργιών τόσο από τον χειριστή των χρηστών όσο και από τη μηχανή φιλτραρίσματος (Σχήμα 14). Ο χειριστής των χρηστών λαμβάνει το προφίλ από τη διεπαφή χρήσης, διαμορφώνει την αναπαράσταση του κάθε υπο – προφίλ με χρήση του μοντέλου Boolean ή Vector Space ανάλογα, αφαιρώντας πρώτα τα stop – words και εισάγει το χρήστη και το προφίλ του στις δομές. Οι όροι που παρουσιάζονται σε κάθε κατηγορία ενδιαφερόντων του χρήστη (υπο – προφίλ) συνδέονται με AND και οι διαφορετικές κατηγορίες συνδέονται μεταξύ τους με OR. Εάν εμφανίζεται η άρνηση κάποιου όρου, χρησιμοποιείται ο τελεστής AND NOT, ενώ η σειρά ανάμεσα σε λέξεις αναπαριστάται με τη χρήστη του τελεστή εγγύτητας. Στην περίπτωση που κάποιο υπο - προφίλ δίνεται με τη μορφή του Vector Space μοντέλου υπολογίζονται τα βάρη των όρων του, σύμφωνα με τον τύπο που έχει προαναφερθεί. Εάν είναι επιθυμητό, το προφίλ και ο χρήστης εισάγονται στη βάση δεδομένων. Επιπλέον, πάλι εάν είναι επιθυμητό, οι όροι του αποστέλλονται στον σημασιολογικό χειριστή, ο οποίος με τη σειρά του ρωτάει το θησαυρό και εισάγει την αντίστοιχη πληροφορία στη δομή των σχετικών όρων.



Σχήμα 14. Λειτουργίες κατά την εμφάνιση προφίλ στο σύστημα

Στη συνέχεια το προφίλ, παραδίνεται στη μηχανή φιλτραρίσματος. Κάθε υπο – προφίλ επεξεργάζεται από τη μηχανή ξεχωριστά. Εάν κάποιο υπο – προφίλ ανήκει στο Vector – Space μοντέλο, πρέπει να υπολογισθεί το μη σημαντικό υποδιάνυσμά του, δεδομένου του κατωφλίου του. Οι όροι του υπο - προφίλ που δεν ανήκουν σε αυτό το μη σημαντικό υποδιάνυσμα είναι αυτοί οι οποίοι θα φιλοξενήσουν το υπο - προφίλ στις λίστες τους στη δομή ευρετηρίου των προφίλ. Εάν από την άλλη πλευρά κάποιο υπο - προφίλ ανήκει στο μοντέλο Boolean, πρέπει να πραγματοποιηθεί ο υπολογισμός του κλειδιού του. Με βάση το βαθμό κάθε λέξης του υπο - προφίλ στο σύστημα, εντοπίζεται η λέξη που είναι θετική και έχει το μικρότερο βαθμό. Το υπο - προφίλ μπαίνει στη λίστα αυτής της λέξης στη δομή ευρετηρίου, η οποία λέξη ονομάζεται κλειδί για το υπο - προφίλ.

Εφόσον η παραπάνω διαδικασίες πραγματοποιηθούν για όλα τα υπο – προφίλ του χρήστη, η μηχανή φιλτραρίσματος εισάγει το προφίλ στη δομή ευρετηρίου. Τα postings του προφίλ εισάγονται στις λίστες των κλειδιών/ σημαντικών όρων αντίστοιχα, και περιέχουν πέρα από το προσδιοριστικό του κάθε υπο - προφίλ, το πλήθος των λέξεων του υπο - προφίλ και όλες τις λέξεις με τα βάρη τους (εκτός από

το κλειδί). Εάν μία λέξη είναι θετική έχει βάρος 1, ενώ εάν είναι αρνητική έχει βάρος –1.

Για να ξεκινήσει η διαδικασία ταιριάσματος, πρέπει να υπολογισθούν και οι καταχωρίσεις του προφίλ στους πίνακες Threshold και Score. Στην περίπτωση του Vector Space μοντέλου, για κάθε υπο - προφίλ εισάγεται στον πίνακα Threshold μία καταχώρηση με την τιμή του κατωφλίου σχετικότητας (relevance threshold) το οποίο έχει καθορίσει ο χρήστης για αυτό το υπο - προφίλ. Για κάθε υπο - προφίλ του Boolean μοντέλου υπολογίζεται το πλήθος των θετικών του όρων. Αυτή η τιμή εισάγεται στον πίνακα Threshold. Ο πίνακας Score αρχικοποιείται με μηδέν.

Τέλος, εκτελείται ο αλγόριθμος «ταιριάσματος» (matching) για όλα τα κείμενα που υπάρχουν στο σύστημα έναντι του νέου προφίλ. Με τη γνωστή μεθοδολογία η μηχανή φιλτραρίσματος επεξεργάζεται τα νέα υπο - προφίλ ως προς τους όρους του κάθε κειμένου και συμπληρώνει διαδοχικά τον πίνακα Score. Τελικά τα υπο - προφίλ που προκύπτει ότι ταιριάζουν σε κάθε κείμενο είναι αυτά που έχουν μεγαλύτερη ή ίση τιμή στην καταχώρησή τους στον πίνακα Score από ότι στην καταχώρησή τους στον πίνακα Threshold. Εάν τα υπο - προφίλ αυτά περιέχουν κάποιο τελεστή εγγύτητας, πραγματοποιείται στη συνέχεια ο έλεγχος εάν το αντίστοιχο σε κάθε περίπτωση κείμενο ικανοποιεί το συγκεκριμένο τελεστή εγγύτητας. Τελικά τα κείμενα αποστέλλονται στο χειριστή χρηστών και στη συνέχεια στους χρήστες στους οποίους αντιστοιχούν τα υπο - προφίλ που ταιριάζουν μετά από όλη την παραπάνω επεξεργασία.

Η παραπάνω διαδικασία αναφέρεται στην περίπτωση που εμφανίζεται ένα νέο προφίλ ενός νέου χρήστη. Όπως όμως θα φανεί και στη συνέχεια, το σύστημα δίνει τη δυνατότητα στο χρήστη να μεταβάλλει το προφίλ του ή και να το διαγράψει εντελώς. Στην περίπτωση της διαγραφής όλου του προφίλ, το προφίλ αφαιρείται τόσο από τις λίστες των προφίλ του χειριστή των χρηστών, όσο και από τη βάση δεδομένων. Στην περίπτωση της μεταβολής του προφίλ, η διαδικασία που αναφέρεται παραπάνω πραγματοποιείται μόνο για τα νέα ή αλλαγμένα υπο - προφίλ του χρήστη.

Όταν εμφανίζεται ένα κείμενο στο σύστημά μας εκτελούνται αρκετές λειτουργίες τόσο από το χειριστή των κειμένων, όσο και από τη μηχανή φιλτραρίσματος (Σχήμα 15**)**. Αρχικά, ο χειριστής των κειμένων ανακτά το κείμενο από τη βάση δεδομένων, διαμορφώνει τόσο την Boolean όσο και τη διανυσματική αναπαράστασή του και το εισάγει στις δομές του. Στο Boolean μοντέλο, το κείμενο

αναπαρίσταται ως ένα διάνυσμα (w_1, w_2, …, w_n), όπου οι όροι w_i είναι οι λέξεις – κλειδιά που εμφανίζονται σε αυτό. Στο Vector Space μοντέλο, το κείμενο αποτελεί ένα διάνυσμα ζευγών (όρος, βάρος), οπότε τα βάρη των όρων του κειμένου υπολογίζονται με βάση το γνωστό τύπο. Στη συνέχεια, οι όροι του κειμένου χρησιμοποιούνται για την ενημέρωση της λίστας που διατηρείται από το χειριστή των κειμένων με τους βαθμούς όλων των όρων. Έπειτα, εάν είναι επιθυμητό, οι όροι του κειμένου αποστέλλονται στον σημασιολογικό χειριστή, ο οποίος με τη σειρά του ρωτάει το θησαυρό και εισάγει την αντίστοιχη πληροφορία στη δομή των σχετικών όρων.

Χειριστής κειμένων	Ανάκτηση κειμένου από τη βάση δεδομένων – Διαμόρφωση αναπαράστασης κειμένου – Εισαγωγή κειμένου στις δομές
	Ενημέρωση λίστας με βαθμούς όρων
	Υπολογισμός βαρών όρων για την αναπαράσταση του κειμένου με βάση το Vector Space μοντέλο
	[Ενημέρωση της δομής σχετικών όρων με βάση τους όρους του κειμένου]
Μηχανή φιλτραρίσματος	Εκτέλεση αλγορίθμου «ταιριάσματος» για το κείμενο ως προς όλα τα προφίλ του συστήματος

Σχήμα 15. Λειτουργίες κατά την εμφάνιση κειμένου στο σύστημα

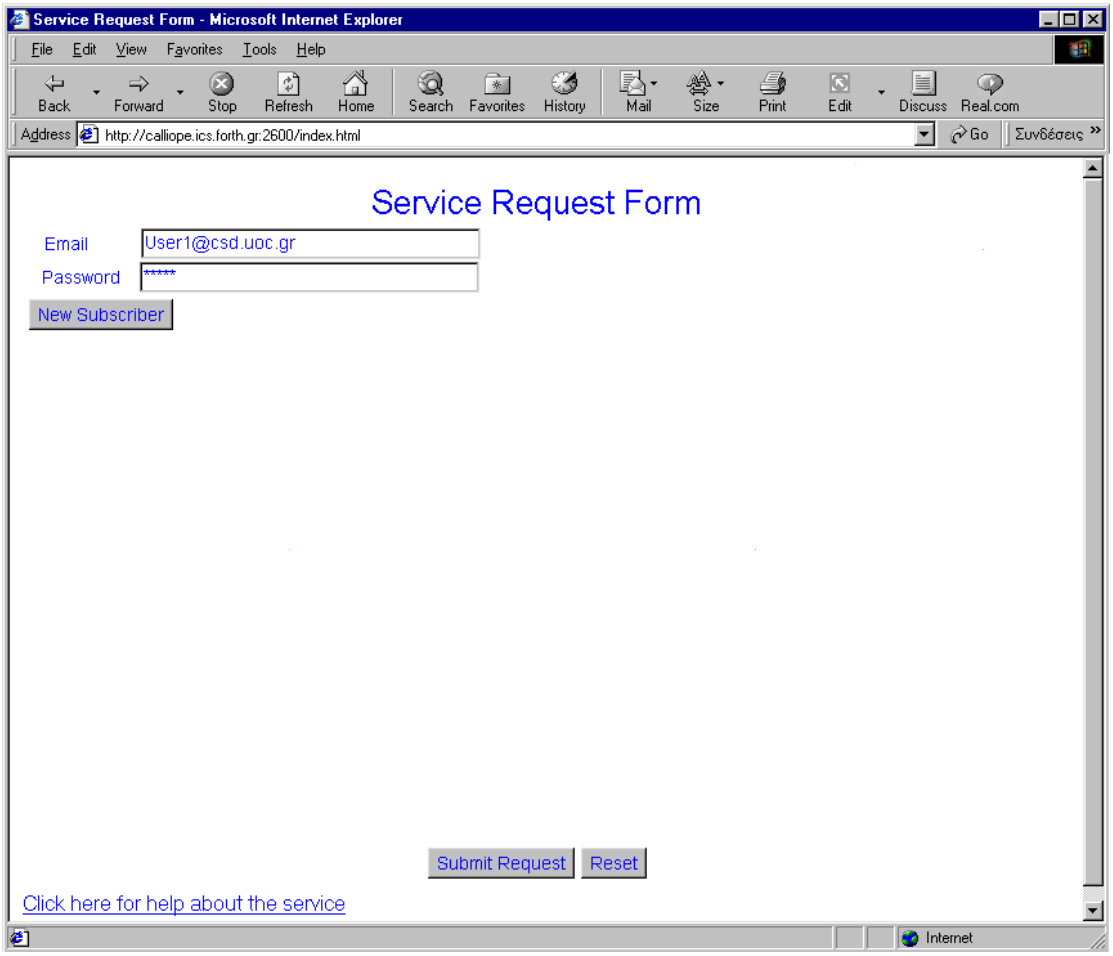
Τέλος, ο χειριστής των κειμένων αποστέλλει το κείμενο στη μηχανή φιλτραρίσματος, η οποία πραγματοποιεί τον έλεγχο ταιριάσματος του κειμένου με τα προφίλ που υπάρχουν τη δεδομένη χρονική στιγμή στο σύστημα. Με τη γνωστή μεθοδολογία η μηχανή φιλτραρίσματος επεξεργάζεται τα υπο - προφίλ ως προς τους όρους του κειμένου και συμπληρώνει διαδοχικά τον πίνακα Score. Τελικά τα υπο - προφίλ που προκύπτει ότι ταιριάζουν στο κείμενο είναι αυτά που έχουν μεγαλύτερη ή ίση τιμή στην καταχώρησή τους στον πίνακα Score από ότι στην καταχώρησή τους στον πίνακα Threshold. Εάν τα υπο - προφίλ αυτά περιέχουν κάποιο τελεστή εγγύτητας, πραγματοποιείται στη συνέχεια ο έλεγχος εάν το κείμενο ικανοποιεί το συγκεκριμένο τελεστή εγγύτητας. Τελικά το κείμενο αποστέλλεται στο χειριστή χρηστών και στη συνέχεια στους χρήστες στους οποίους αντιστοιχούν τα υπο - προφίλ που ταιριάζουν μετά από όλη την παραπάνω επεξεργασία.

5.2 Διεπαφή χρήσης (Client)

Η επικοινωνία του χρήστη με το σύστημα πραγματοποιείται μέσω ενός Web Browser. Μέσω της συγκεκριμένης διεπαφής χρήσης, ο χρήστης μπορεί να εγγραφεί στην υπηρεσία, να διαχειριστεί το προφίλ του (εισαγωγή, μεταβολή, διαγραφή προφίλ) και να δει τα κείμενα που ταιριάζουν σε αυτό το προφίλ.

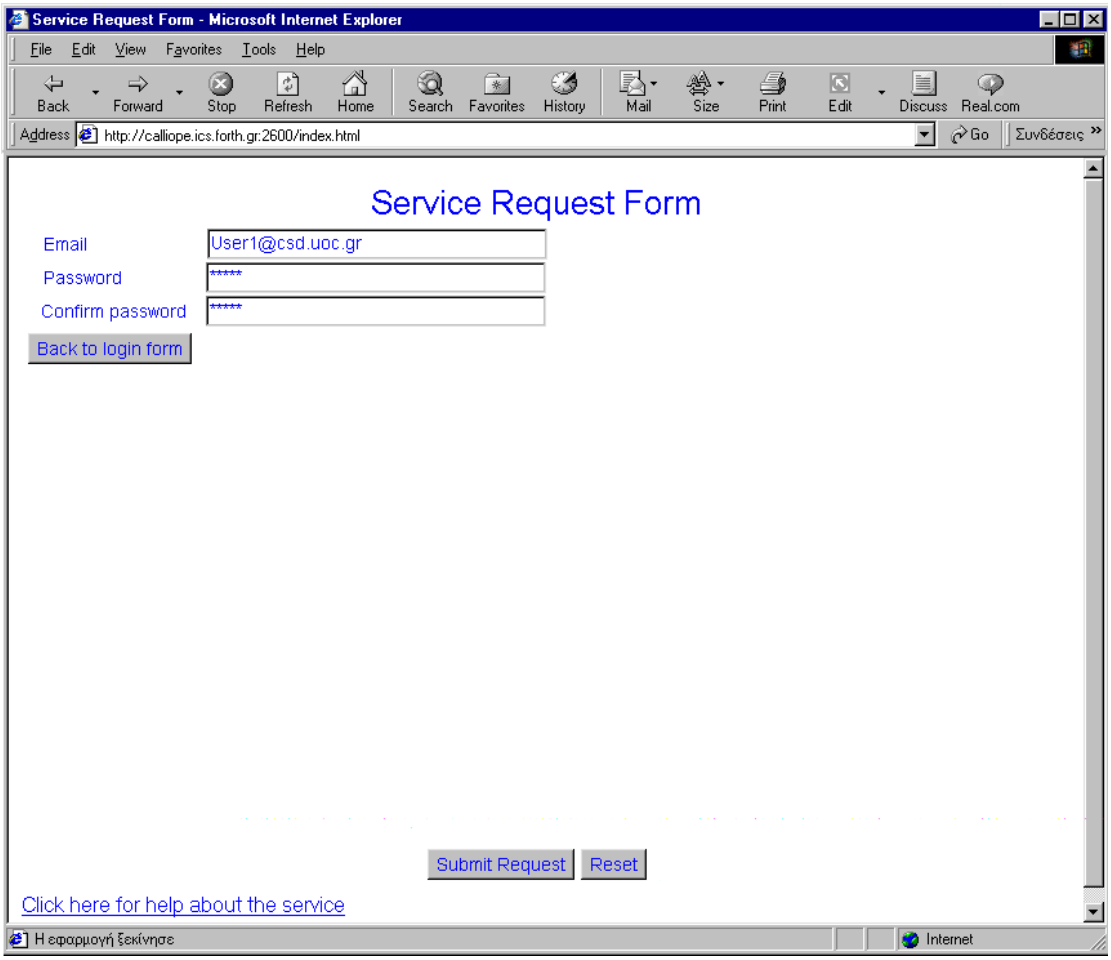
5.2.1 Ταυτοποίηση/ Εγγραφή χρήστη

Αρχικά το σύστημα ζητάει την ταυτοποίηση του χρήστη, όπως φαίνεται και στην φόρμα στο Σχήμα 16. Ο χρήστης θέτει το email και το password του και στη συνέχεια επιλέγει το κουμπί "Submit Request".



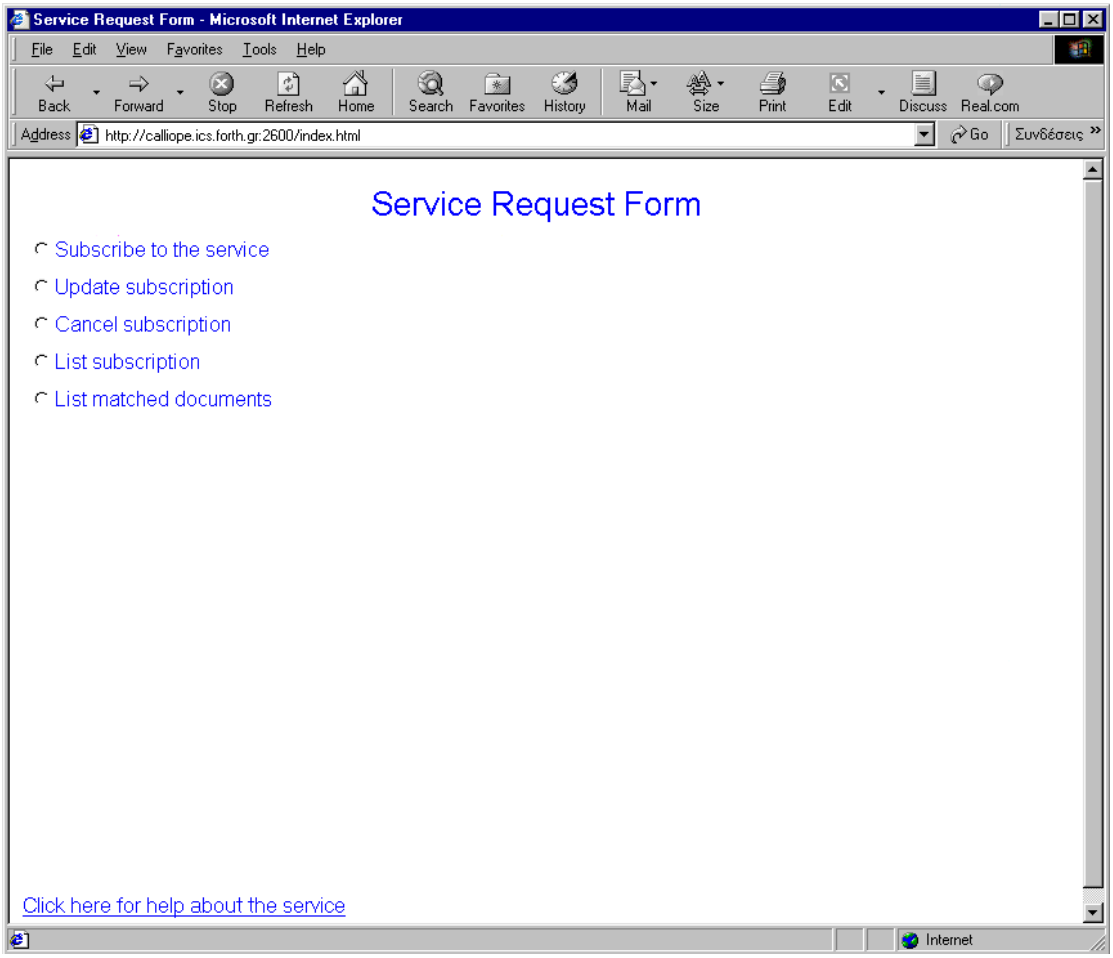
Σχήμα 16. Εισαγωγή «παλιού» χρήστη στο σύστημα.

Εάν ο χρήστης δεν έχει εγγραφεί στο σύστημα, επιλέγει το κουμπί "New Subscriber". Τότε εμφανίζεται η φόρμα στο Σχήμα 17, στην οποία ο χρήστης συμπληρώνει τα πεδία και ακολούθως επιλέγει το κουμπί "Submit Request". Το σύστημα αποδέχεται την εγγραφή ενός νέου χρήστη, εφόσον δεν υπάρχει κάποιος άλλος χρήστης με το ίδιο e-mail.



Σχήμα 17. Εγγραφή «νέου» χρήστη σύστημα

Εφόσον η ταυτοποίηση του χρήστη ολοκληρωθεί, παρουσιάζεται σε αυτόν η φόρμα στο Σχήμα 18. Σε αυτή τη φόρμα υπάρχει ένα μενού με όλες τις δυνατές ενέργειες που μπορούν να πραγματοποιηθούν, δηλαδή η εφαρμογή ενός προφίλ ("Subscribe to the service"), η μεταβολή του προφίλ ("Update subscription"), η διαγραφή του προφίλ ("Cancel subscription"), η εμφάνιση του προφίλ ("List subscription") και τέλος η παρουσίαση των κειμένων που ταιριάζουν στο προφίλ του χρήστη ("List matched documents").

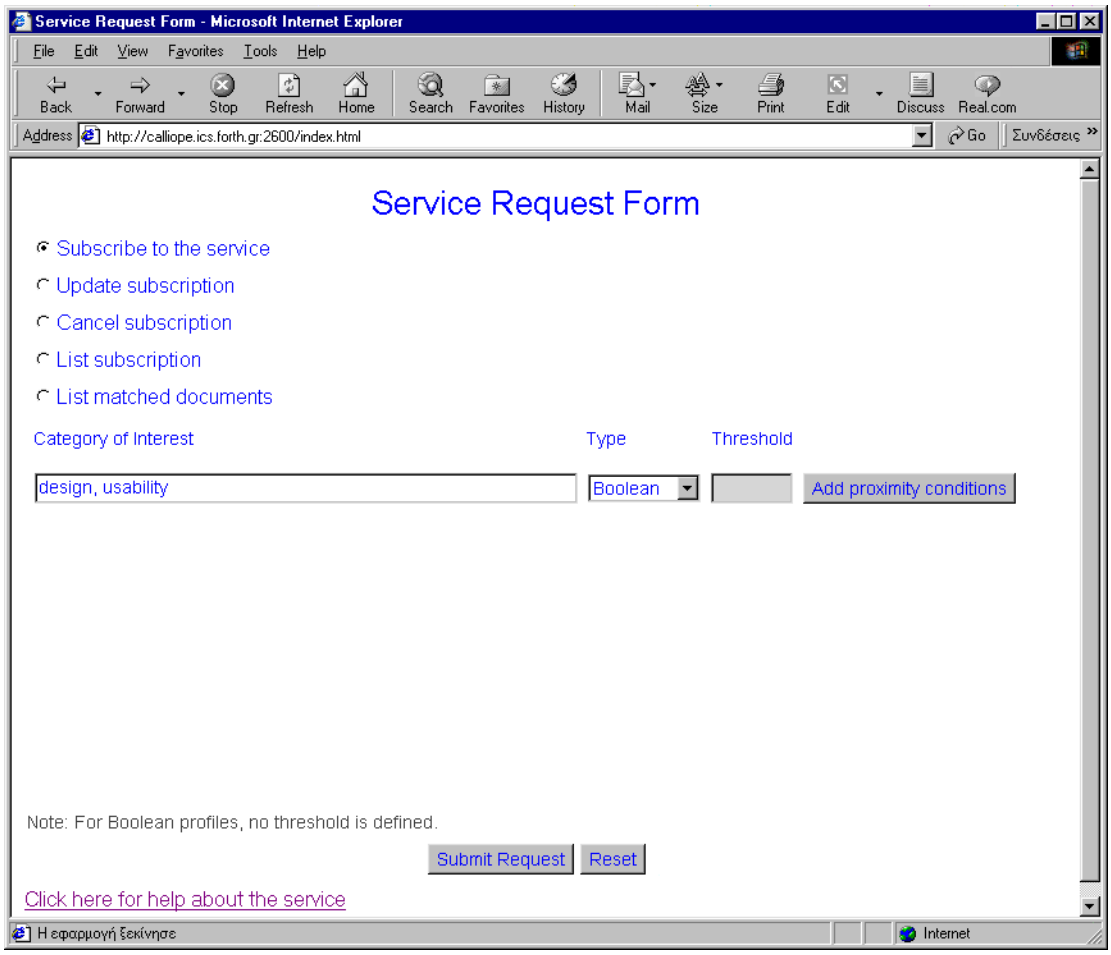


Σχήμα 18. Μενού επιλογών

5.2.2 Εισαγωγή προφίλ χρήστη

Επιλέγοντας το "Subscribe to the service" , εμφανίζεται η φόρμα στο Σχήμα 19, στην οποία ο χρήστης μπορεί να θέσει το προφίλ του. Σε κάθε στιγμιότυπο της φόρμας, ο χρήστης μπορεί να υποβάλλει και ένα καινούργιο υπο – προφίλ. Πιο συγκεκριμένα στο πεδίο της κατηγορίας ενδιαφερόντων εισάγει τους όρους για τους οποίους ενδιαφέρεται, χωρισμένους με κόμμα. Στη συνέχεια επιλέγει το είδος του συγκεκριμένου υπο – προφίλ, το οποίο μπορεί να είναι είτε Boolean είτε Weighted (με βάρη). Στην περίπτωση που το υπο – προφίλ είναι τύπου Weighted, ο χρήστης οφείλει να ορίσει και κάποιο κατώφλι σχετικότητας (relevance threshold), το οποίο πρέπει να ανήκει στο διάστημα [0, 1].

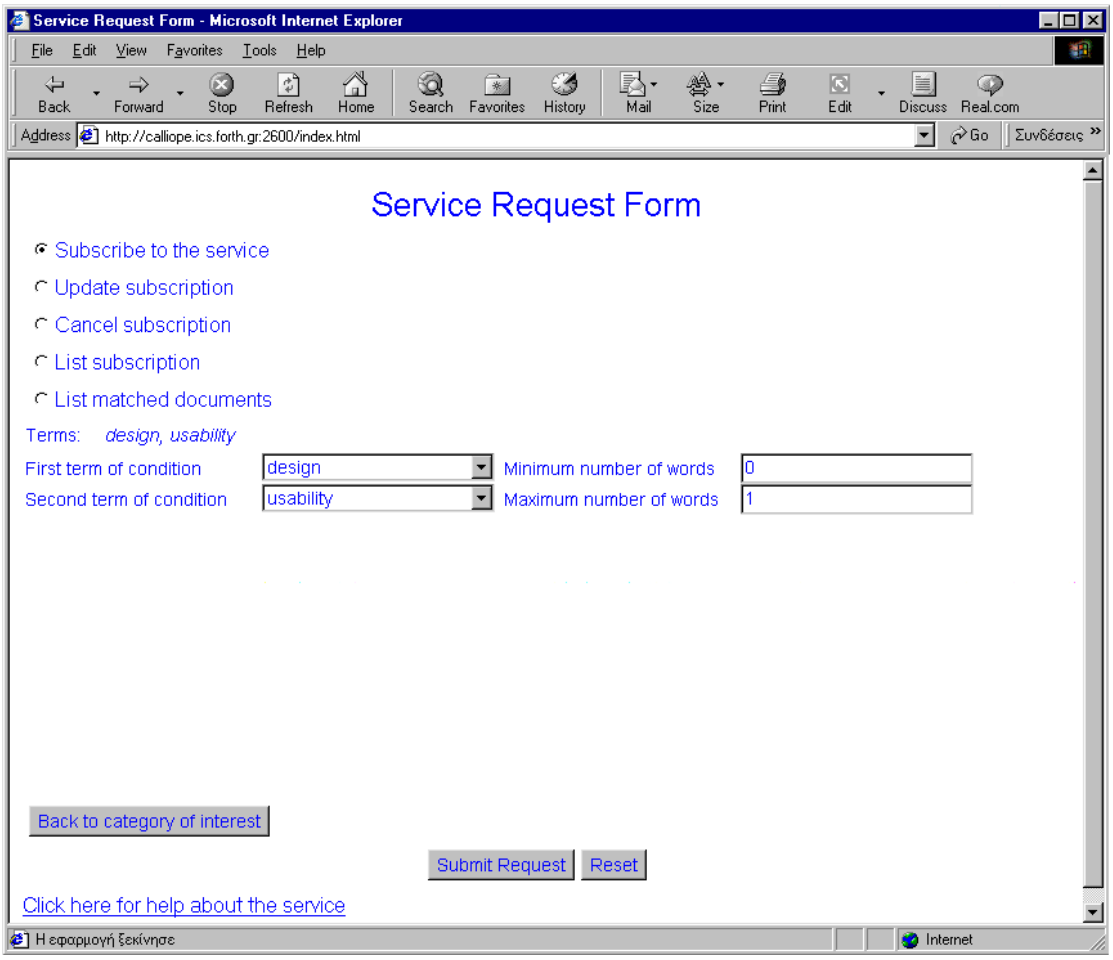
Συγκεκριμένα, στο Σχήμα 19 παρατηρούμε την εισαγωγή του υπο - προφίλ "design, usability". Καθώς ο τύπος του είναι Boolean, το πεδίο του κατωφλίου (threshold) είναι απενεργοποιημένο.



Σχήμα 19. Εισαγωγή Boolean υπο - προφίλ

Στην περίπτωση που το υπο – προφίλ είναι τύπου Boolean, ο χρήστης έχει τη δυνατότητα να ορίσει και κάποια συνθήκη εγγύτητας. Αυτό επιτυγχάνεται επιλέγοντας το κουμπί "Add proximity conditions", το οποίο έχει σαν αποτέλεσμα την εμφάνιση της φόρμας στο Σχήμα 20. Σε αυτή τη φόρμα ο χρήστης μπορεί να ορίσει μία σειρά συνθηκών εγγύτητας ανάμεσα στους όρους του υπο – προφίλ του, δίνοντας τιμές στα τέσσερα πεδία που εμφανίζονται και επιλέγοντας στη συνέχεια το κουμπί "Submit Request". Οι όροι που λαμβάνουν μέρος στη συνθήκη εγγύτητας επιλέγονται από μία λίστα με όλους τους όρους του υπο - προφίλ. Η συνθήκη εγγύτητας γίνεται αποδεκτή από το σύστημα, εφόσον οι δύο όροι είναι διαφορετικοί και ο αριθμός "min number of words" είναι μικρότερος από τον αριθμό "max number

of words". Επίσης δεν πρέπει να υπάρχει άλλη συνθήκη εγγύτητας στο ίδιο υπο – προφίλ με τους ίδιους αντίστοιχους όρους.



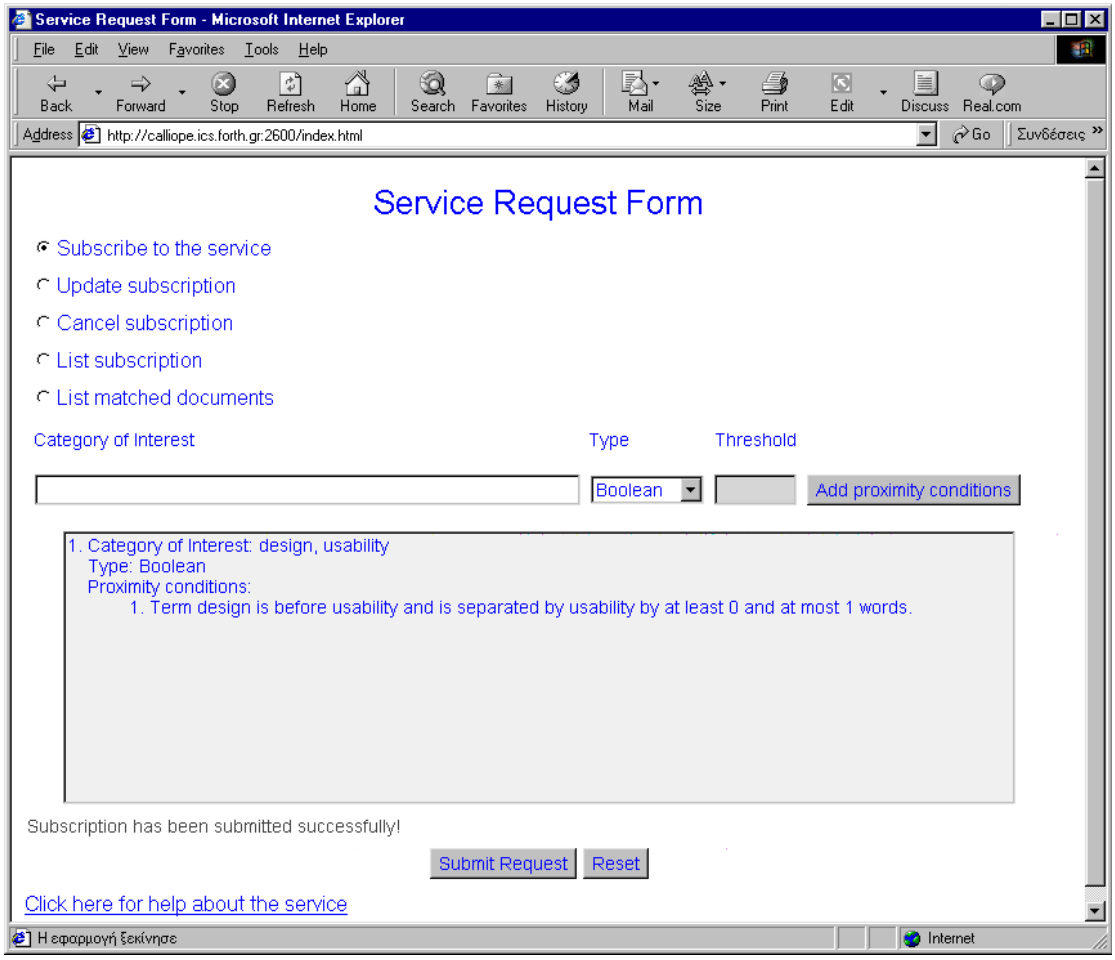
Σχήμα 20. Εισαγωγή συνθήκης εγγύτητας σε υπο - προφίλ

Μετά την ολοκλήρωση των συνθηκών εγγύτητας ο χρήστης μπορεί να επιστρέψει στη φόρμα στο Σχήμα 21 επιλέγοντας το κουμπί "Back to category of interest". Τότε μπορεί να αποστείλει το υπο - προφίλ στο σύστημα επιλέγοντας το κουμπί "Submit Request".

Το σύστημα κάνει αποδεκτό το υπο - προφίλ, μόνο εφόσον αυτό ικανοποιεί κάποιες συνθήκες. Αρχικά στην περίπτωση που το υπο – προφίλ ανήκει στο Vector Space μοντέλο, το κατώφλι σχετικότητας πρέπει να έχει τιμή μέσα στα αποδεκτά όρια, δηλαδή μέσα στο διάστημα [0, 1]. Επίσης το υπο – προφίλ δεν επιτρέπεται να έχει κάποιο αρνητικό όρο, δηλαδή κάποιο όρο, ο οποίος ξεκινά με τη λέξη "not". Στην περίπτωση που το υπο – προφίλ είναι τύπου Boolean, δε μπορεί να περιέχει μόνο αρνητικούς όρους, δηλαδή μόνο όρους που αρχίζουν με τη λέξη "not". Τέλος, το

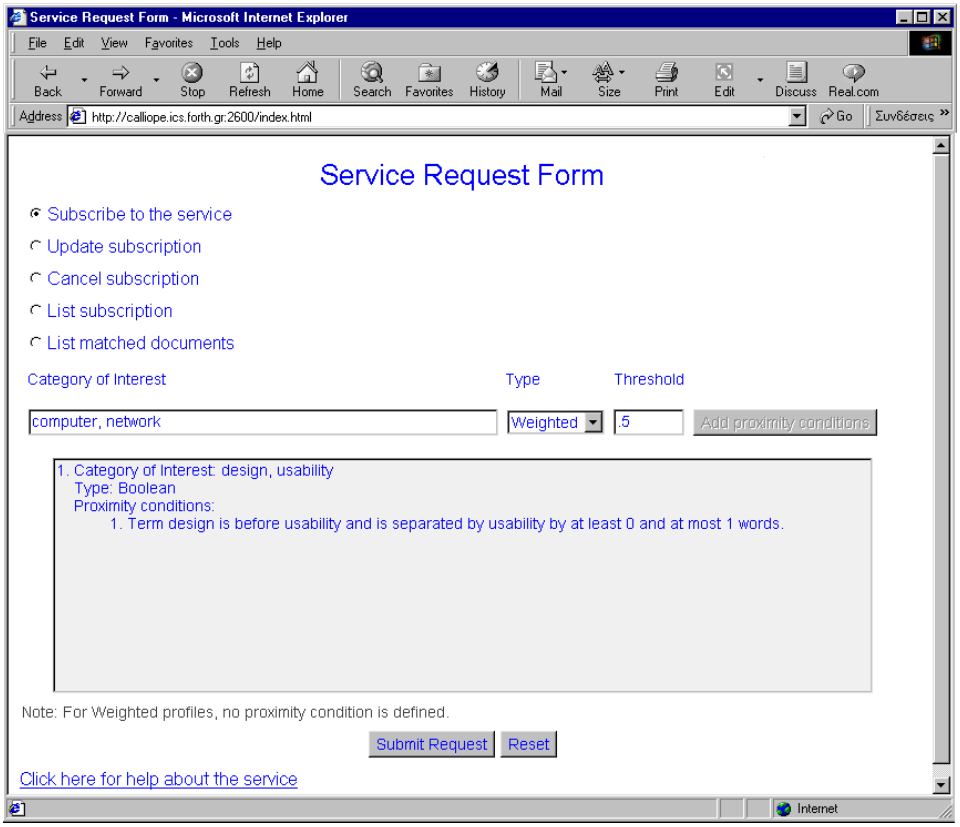
σύστημα δεν επιτρέπει την εισαγωγή δύο προφίλ με τους ίδιους ακριβώς όρους, παρά μόνο στην περίπτωση που τα δύο αυτά προφίλ έχουν διαφορετικό τύπο.

Μετά την ολοκλήρωση του υπο – προφίλ και την επιλογή του κουμπιού "Submit Request", εμφανίζεται η φόρμα στο Σχήμα 21. Σε αυτήν ο χρήστης μπορεί να εισάγει ένα καινούργιο υπο – προφίλ, ενώ παράλληλα μπορεί να δει όλα τα υπο – προφίλ τα οποία έχει ήδη υποβάλλει στο σύστημα.

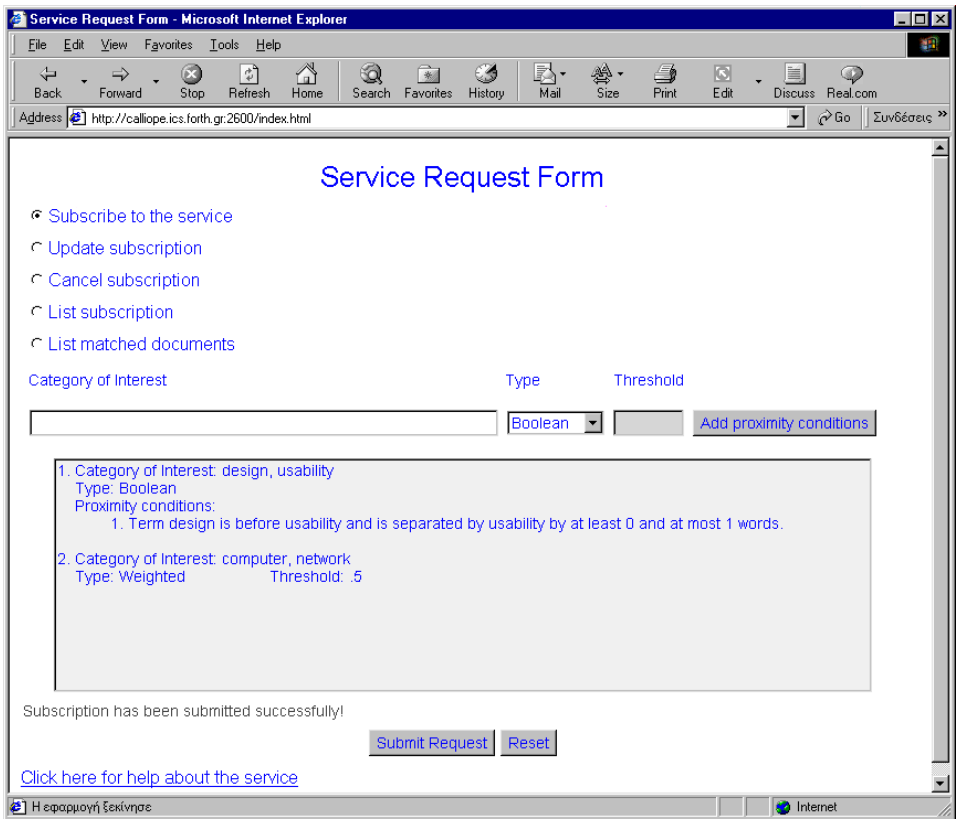


Σχήμα 21. Αποστολή προφίλ στο σύστημα

Στη φόρμα στο Σχήμα 22, παρατηρούμε την εισαγωγή μίας δεύτερης κατηγορίας ενδιαφερόντων από το χρήστη, η οποία είναι τύπου Vector Space. Παρατηρούμε ότι επειδή το συγκεκριμένο υπο – προφίλ έχει τύπο "Weighted", το κουμπί "Add proximity conditions" είναι απενεργοποιημένο, καθώς δεν επιτρέπεται ο ορισμός συνθήκης εγγύτητας για τα υπο – προφίλ του συγκεκριμένου τύπου. Όταν ο χρήστης αποθηκεύσει και αυτήν την κατηγορία ενδιαφερόντων εμφανίζεται σε αυτόν η φόρμα στο Σχήμα 23.



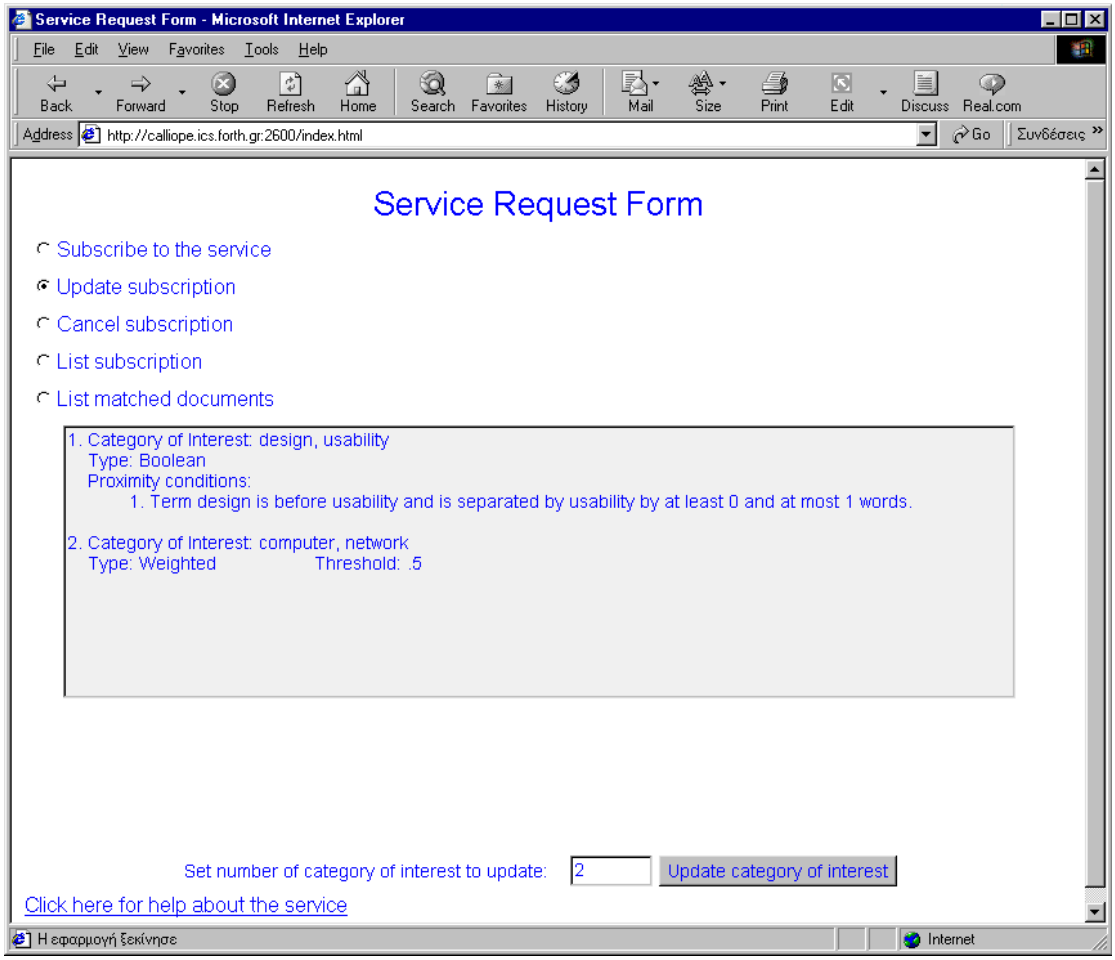
Σχήμα 22. Εισαγωγή Vector Space υπο - προφίλ



Σχήμα 23. Αποστολή υπο - προφίλ στο σύστημα

5.2.3 Μεταβολή προφίλ χρήστη

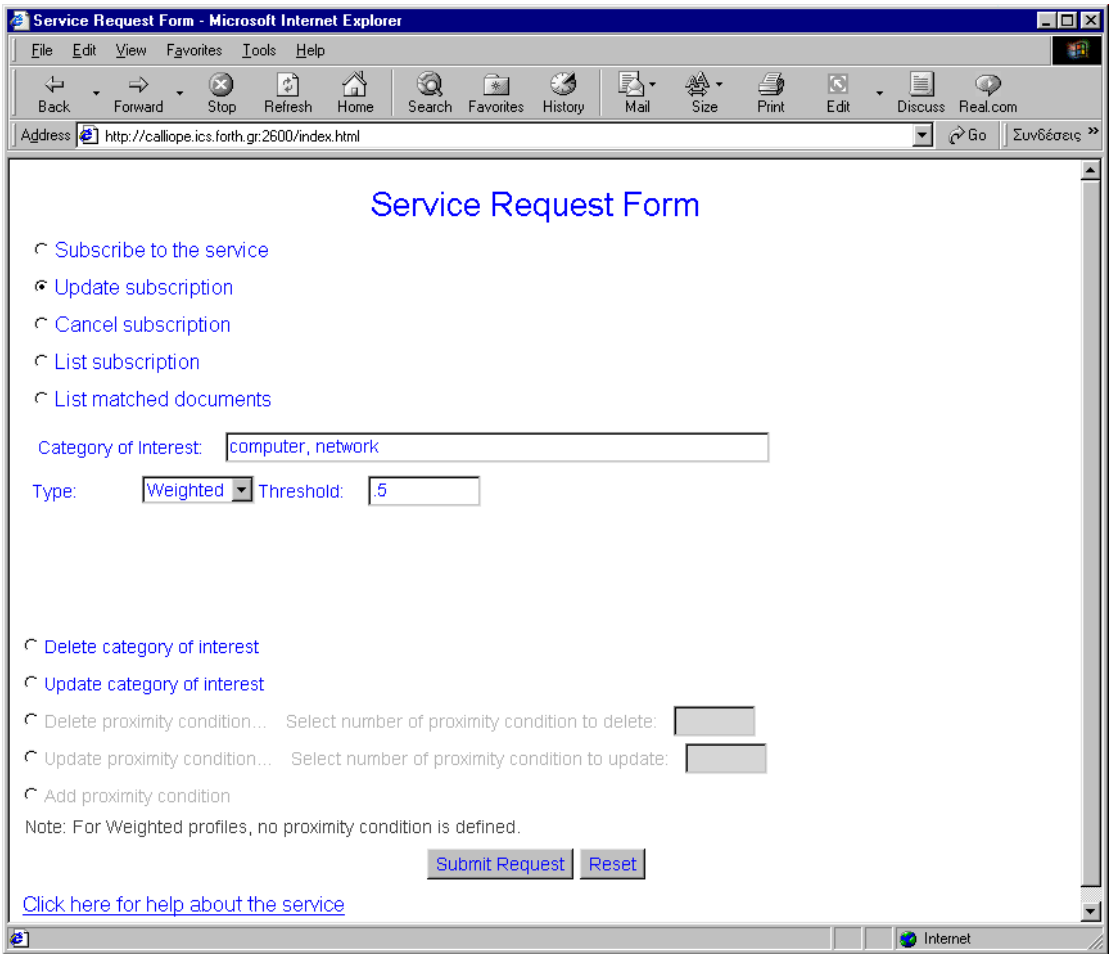
Η δεύτερη επιλογή του μενού "Update subscription" δίνει τη δυνατότητα στο χρήστη να μεταβάλλει το προφίλ του. Επιλέγοντας τη συγκεκριμένη επιλογή, εμφανίζεται η φόρμα στο Σχήμα 24. Σε αυτή τη φόρμα ο χρήστης βλέπει όλο το προφίλ του.



Σχήμα 24. Μεταβολή προφίλ

Για να μεταβάλλει κάποια συγκεκριμένη κατηγορία ενδιαφερόντων του, πρέπει να εισάγει τον αύξων αριθμό της συγκεκριμένης κατηγορίας στο πεδίο που φαίνεται στο κάτω μέρος της φόρμας και να πιέσει το κουμπί "Update category of interest". Εάν η τιμή που εισαχθεί στο πεδίο δεν είναι έγκυρη εμφανίζεται το αντίστοιχο μήνυμα λάθους.

Στην αντίθετη περίπτωση, παρουσιάζεται η φόρμα στο Σχήμα 25, στην οποία ο χρήστης μπορεί να κάνει όποιες αλλαγές επιθυμεί στο υπο – προφίλ του, να επιλέξει στη συνέχεια το είδος της μεταβολής από το μενού στο κάτω μέρος και να πιέσει το κουμπί "Submit Request".



Σχήμα 25. Μεταβολή Vector Space υπο – προφίλ

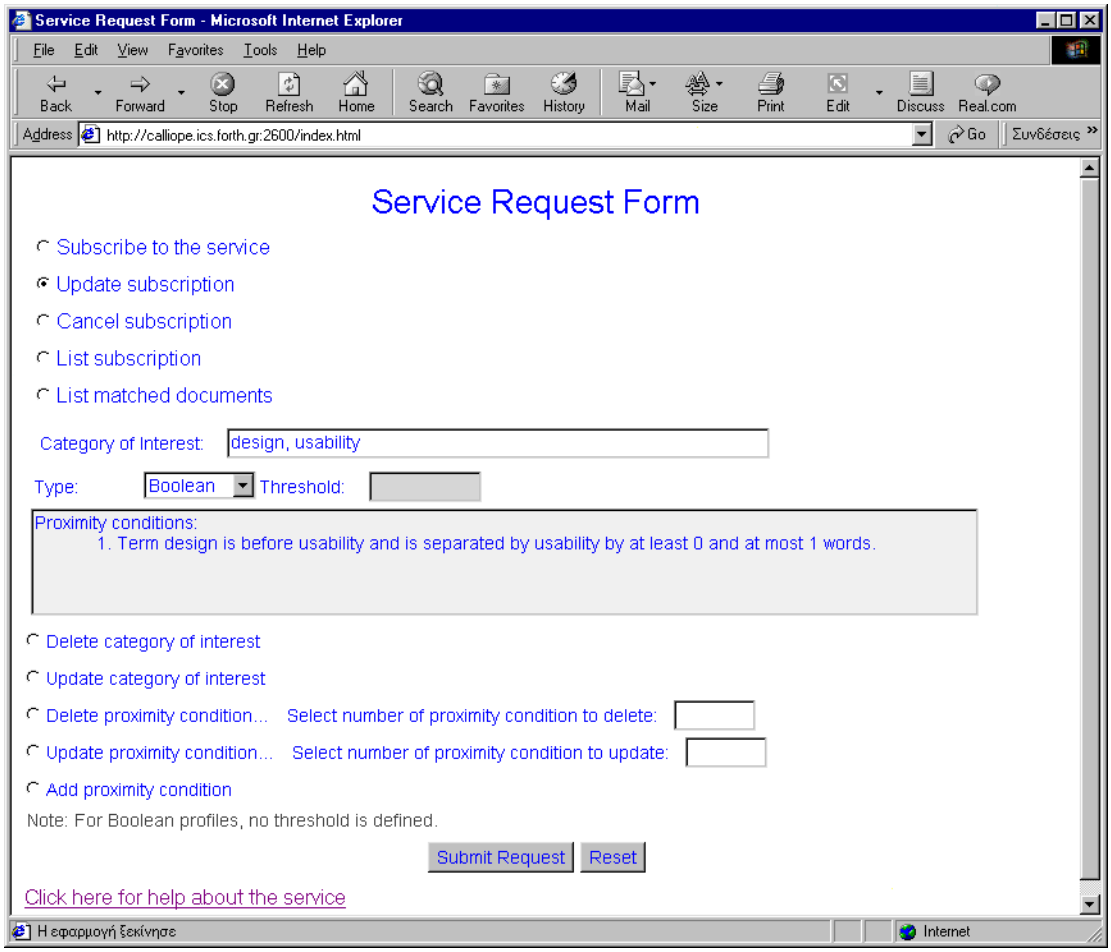
Όπως παρατηρούμε σε αυτή τη φόρμα, στο πάνω μέρος εμφανίζονται τα στοιχεία του συγκεκριμένου υπο – προφίλ. Καθώς το συγκεκριμένο υπο – προφίλ είναι τύπου Vector Space, ο χρήστης μπορεί μόνο να το διαγράψει και να μεταβάλλει είτε τους όρους, είτε τον τύπο του, είτε το κατώφλι σχετικότητας.

Για να το διαγράψει, διαλέγει την επιλογή του μενού "Delete category of interest" και στη συνέχεια επιλέγει το κουμπί "Submit Request". Σε αυτήν την περίπτωση το υπο – προφίλ διαγράφεται εντελώς από το προφίλ του χρήστη και ο χρήστης μεταφέρεται στη φόρμα στο Σχήμα 24.

Από την άλλη πλευρά, ο χρήστης μπορεί να μεταβάλλει οποιοδήποτε από τα πεδία του προφίλ, ακολούθως να διαλέξει την επιλογή "Update category of interest"

και στη συνέχεια να επιλέξει το κουμπί "Submit Request". Τότε ο χρήστης παραμένει στο ίδιο περιβάλλον, με τη μόνη διαφορά ότι το προφίλ έχει πλέον την νέα αλλαγμένη μορφή.

Εάν ο χρήστης επιλέξει από τη φόρμα στο Σχήμα 24 να μεταβάλλει το πρώτο προφίλ, θα εμφανιστεί η φόρμα στο Σχήμα 26. Καθώς το υπο - προφίλ είναι τύπου Boolean, ο χρήστης μπορεί να διαγράψει και να μεταβάλλει το υπο - προφίλ (ενέργειες που υλοποιούνται με τον ίδιο τρόπο όπως και στο υπο – προφίλ μοντέλου Vector Space) και επιπλέον να κάνει κάποιες μεταβολές στις συνθήκες εγγύτητας του υπο – προφίλ.

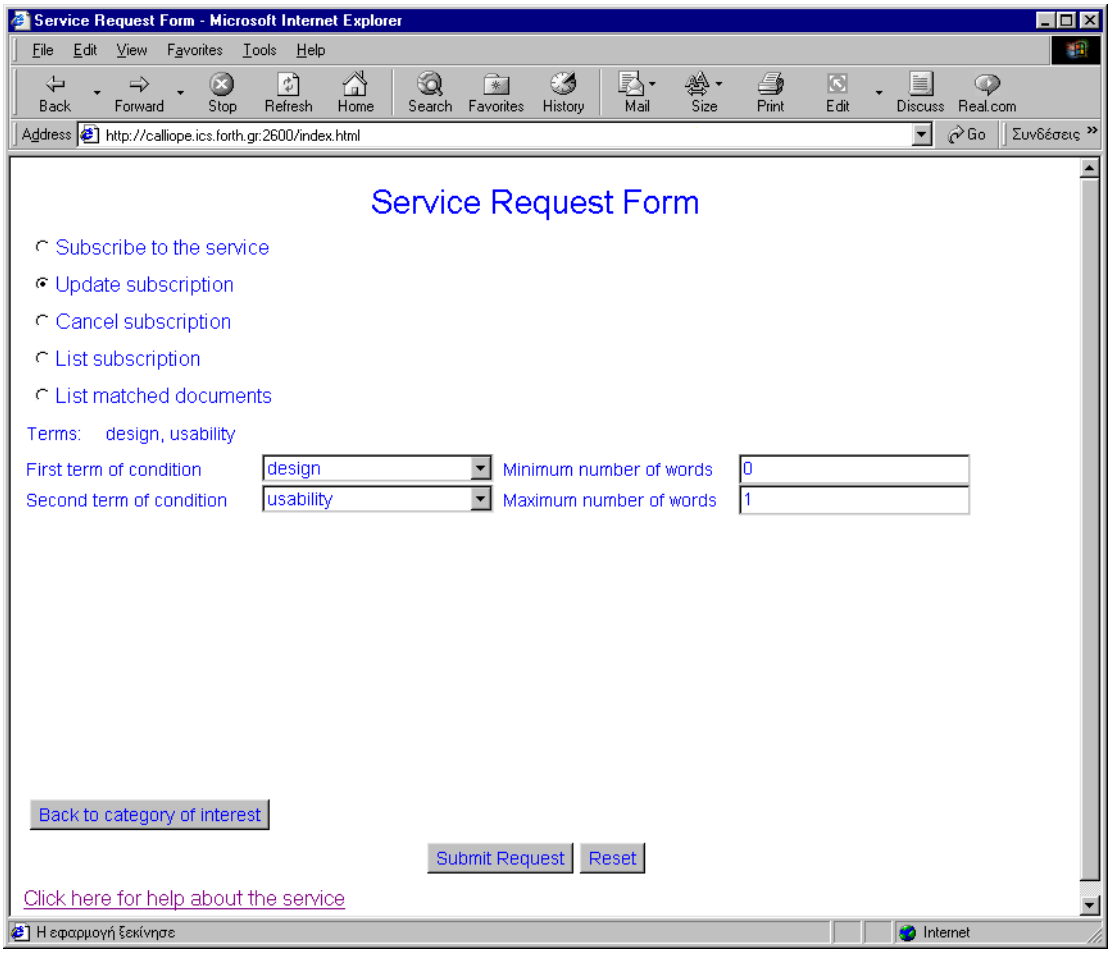


Σχήμα 26. Μεταβολή Boolean υπο – προφίλ

Αρχικά μπορεί να διαγράψει κάποια συνθήκη εγγύτητας, διαλέγοντας την επιλογή "Delete proximity condition", εισάγοντας τον αύξων αριθμό της συνθήκης στο πεδίο που φαίνεται δίπλα και επιλέγοντας τέλος το κουμπί "Submit Request". Η αίτηση του χρήστη γίνεται αποδεκτή μόνο στην περίπτωση που ο αύξων αριθμός

είναι έγκυρος. Τότε ο χρήστης παραμένει στο ίδιο περιβάλλον με τη μόνη διαφορά ότι το προφίλ δεν έχει πλέον τη διαγραφείσα συνθήκη εγγύτητας.

Παράλληλα, ο χρήστης μπορεί να μεταβάλλει κάποια συνθήκη εγγύτητας, διαλέγοντας την επιλογή "Update proximity condition", εισάγοντας τον αύξων αριθμό της συνθήκης στο πεδίο που φαίνεται δίπλα και επιλέγοντας τέλος το κουμπί "Submit Request". Σε αυτήν την περίπτωση, εμφανίζεται η φόρμα στο Σχήμα 27, στην οποία μπορεί να κάνει τις αντίστοιχες αλλαγές και να τις αποθηκεύσει επιλέγοντας το κουμπί "Submit Request". Η μεταβολή της συνθήκης εγγύτητας γίνεται αποδεκτή από το σύστημα μόνο εφόσον ικανοποιούνται οι συνθήκες που αναφέρθηκαν και στην εισαγωγή τέτοιας συνθήκης.

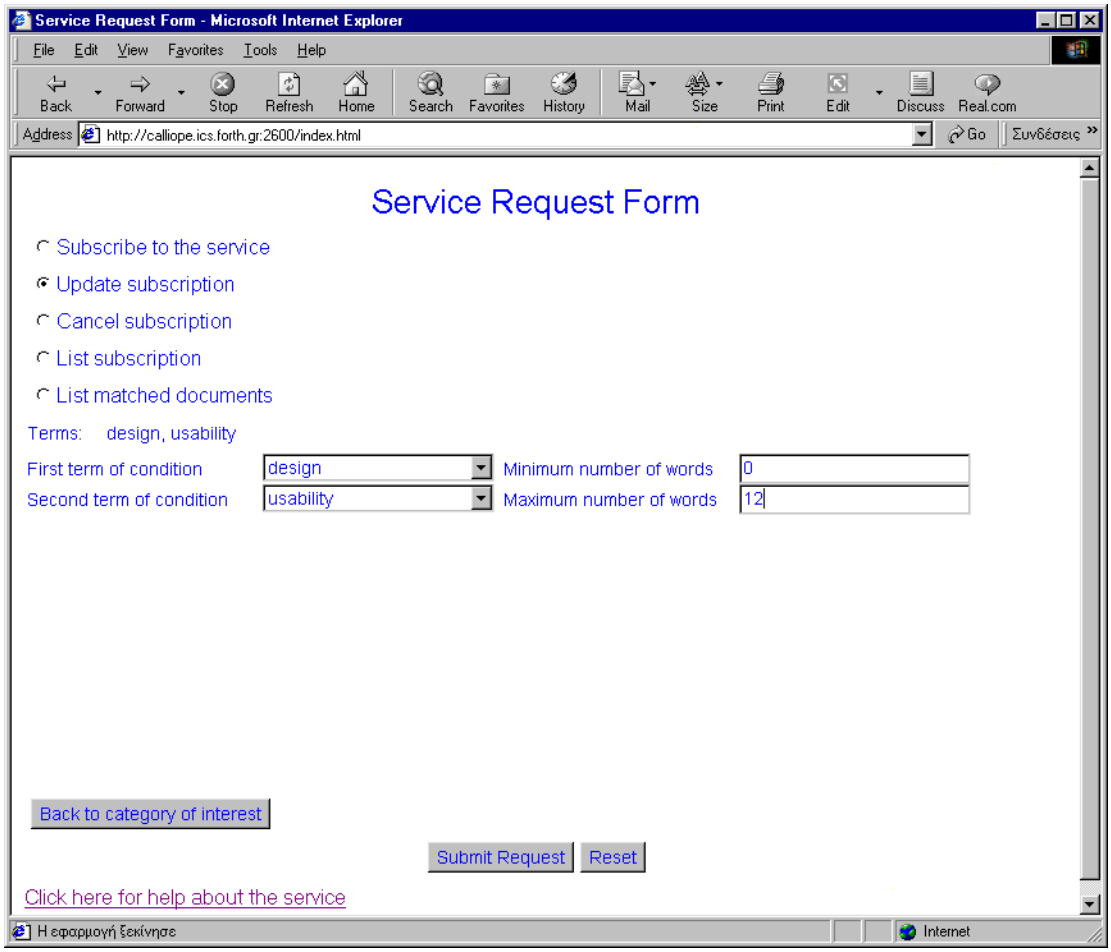


Σχήμα 27. Μεταβολή συνθήκης εγγύτητας σε υπο – προφίλ

Όταν ο χρήστης ολοκληρώσει την μεταβολή της συνθήκης εγγύτητας μπορεί να επιλέξει το κουμπί "Back to category of interest" και να επιστρέψει στη φόρμα στο

Σχήμα 26, στην οποία πλέον η συνθήκη εγγύτητας εμφανίζεται με τη νέα της αλλαγμένη μορφή.

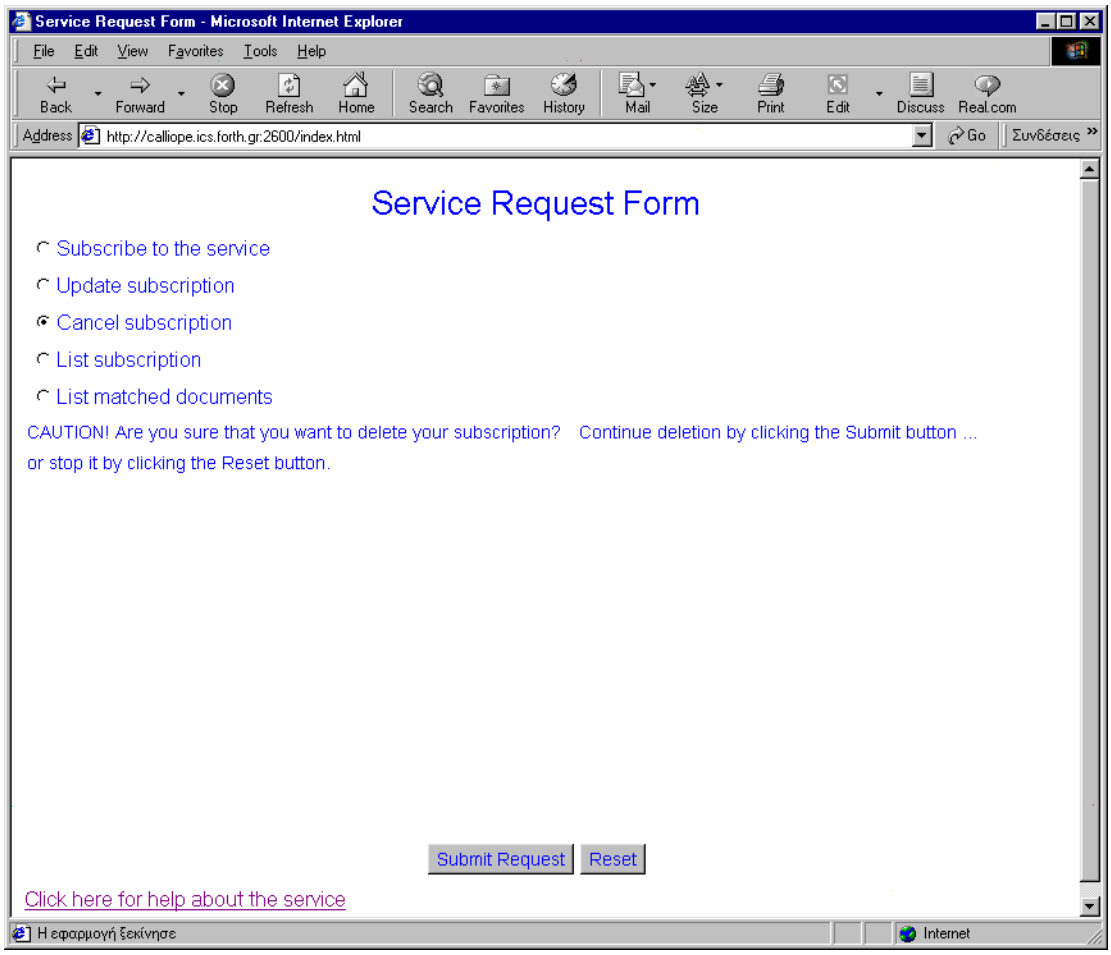
Τέλος, ο χρήστης μπορεί να προσθέσει κάποια καινούργια συνθήκη εγγύτητας στο συγκεκριμένο υπο – προφίλ. Για να υλοποιήσει αυτή τη λειτουργία, διαλέγει την επιλογή "Add proximity condition" και επιλέγει το κουμπί "Submit Request". Τότε εμφανίζεται η φόρμα στο Σχήμα 28, στην οποία ο χρήστης εισάγει τιμές σε όλα τα πεδία και αποθηκεύει τη συνθήκη επιλέγοντας το κουμπί "Submit Request". Προφανώς και σε αυτήν την περίπτωση η νέα συνθήκη εγγύτητας γίνεται αποδεκτή μόνο όταν ικανοποιεί τις προαναφερθείσες συνθήκες.



Σχήμα 28. Προσθήκη συνθήκης εγγύτητας σε υπο – προφίλ

5.2.4 Διαγραφή προφίλ χρήστη

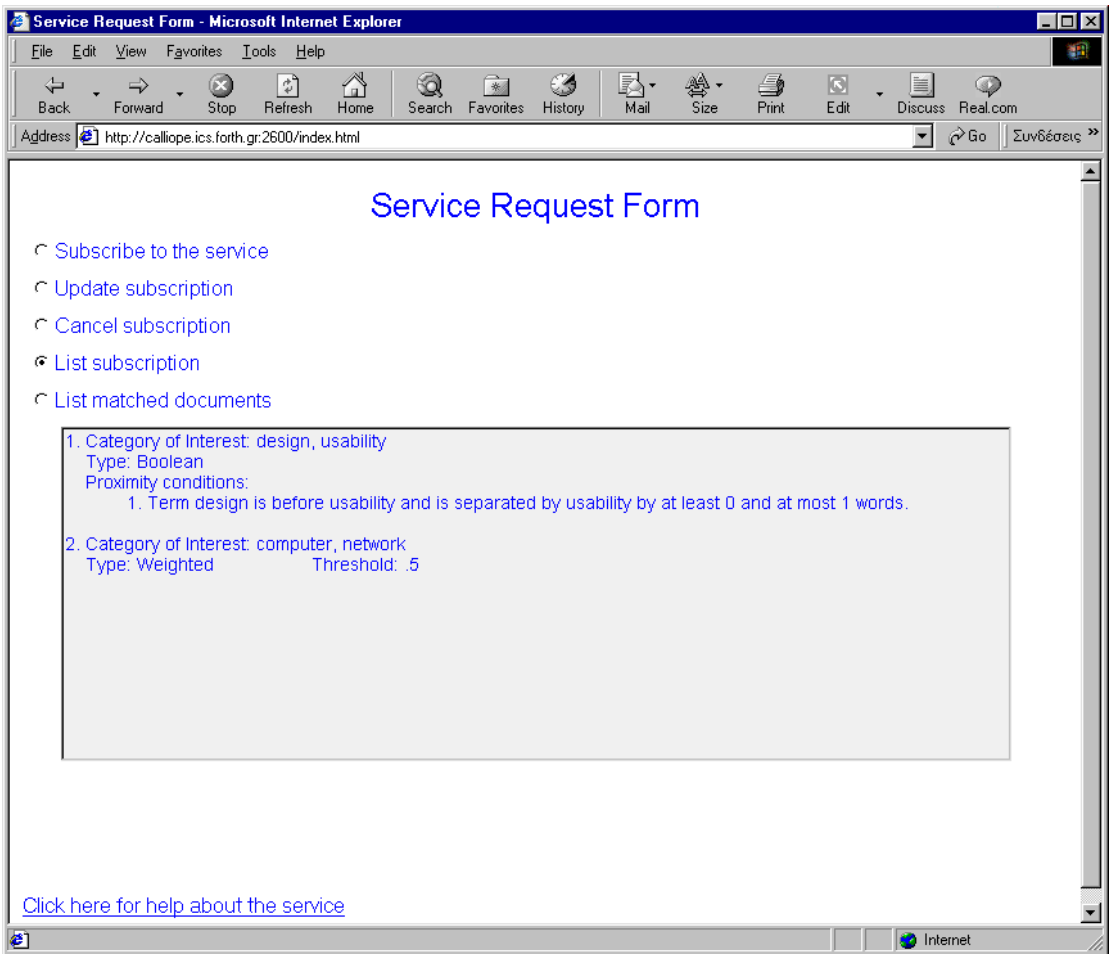
Το σύστημα δίνει τη δυνατότητα στο χρήστη να διαγράψει όλο το προφίλ του, δηλαδή το σύνολο όλων των υπο – προφίλ του. Αυτό γίνεται εάν διαλέξει την τρίτη επιλογή του μενού "Cancel Subscription", οπότε και εμφανίζεται η φόρμα στο Σχήμα 29. Τότε ο χρήστης μπορεί να πιέσει το κουμπί "Submit Request", με αποτέλεσμα όλο το προφίλ του να διαγραφεί. Διαφορετικά, μπορεί να επιλέξει το κουμπί "Reset", το οποίο ουσιαστικά δεν προκαλεί καμία μεταβολή στο προφίλ του.



Σχήμα 29. Διαγραφή προφίλ από το σύστημα

5.2.5 Εμφάνιση προφίλ χρήστη

Με την επιλογή τέταρτη επιλογή του μενού "List subscription", εμφανίζεται η φόρμα στο Σχήμα 30, στην οποία ο χρήστης μπορεί να δει το προφίλ του. Στη συγκεκριμένη φόρμα, δε μπορεί να υλοποιηθεί καμία ενέργεια από το χρήστη. Για αυτό το λόγο απουσιάζουν και τα δύο κουμπιά "Submit Request" και "Reset".



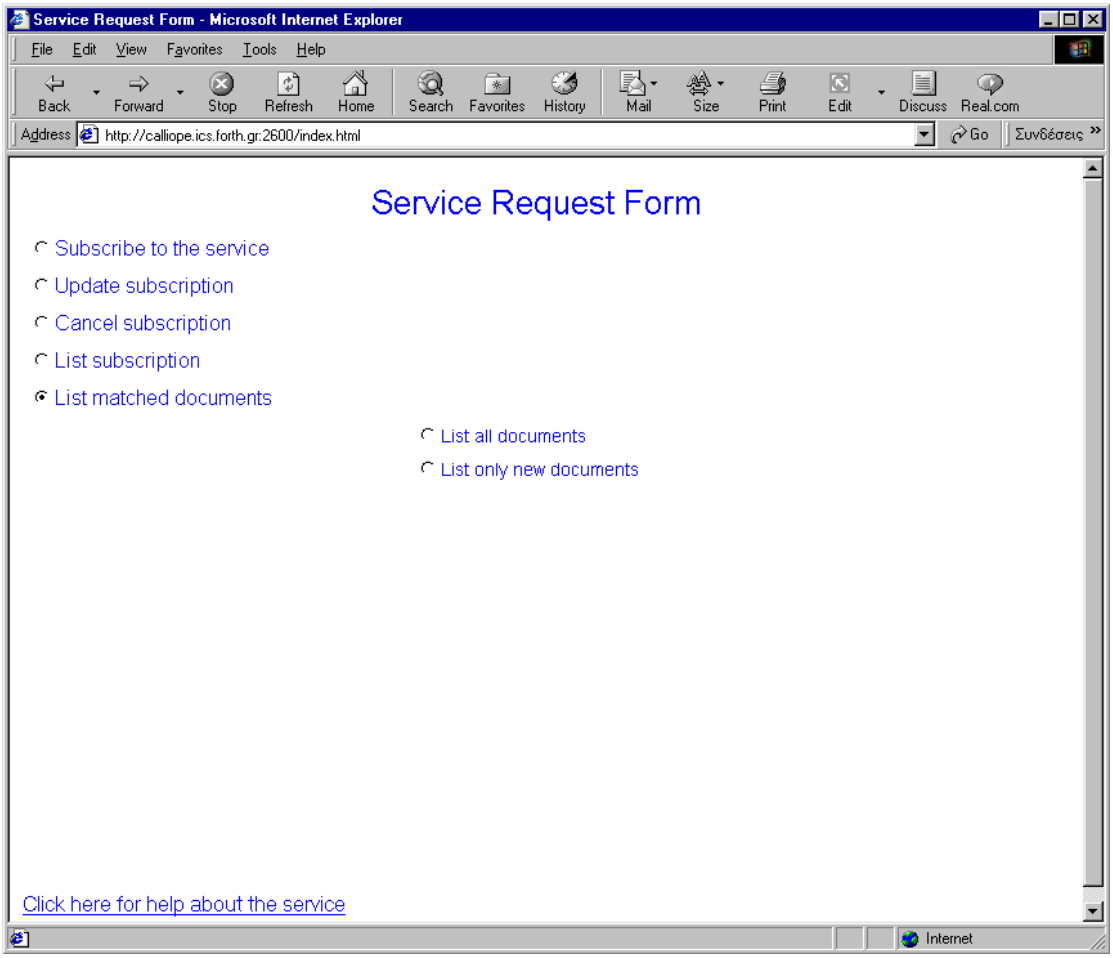
Σχήμα 30. Εμφάνιση προφίλ χρήστη

5.2.6 Εμφάνιση κειμένων που ταιριάζουν στο προφίλ χρήστη

Μία από τις βασικότερες λειτουργικότητες που προσφέρει η υπηρεσία παρέχεται από την τελευταία επιλογή του μενού, την "List matched documents", η

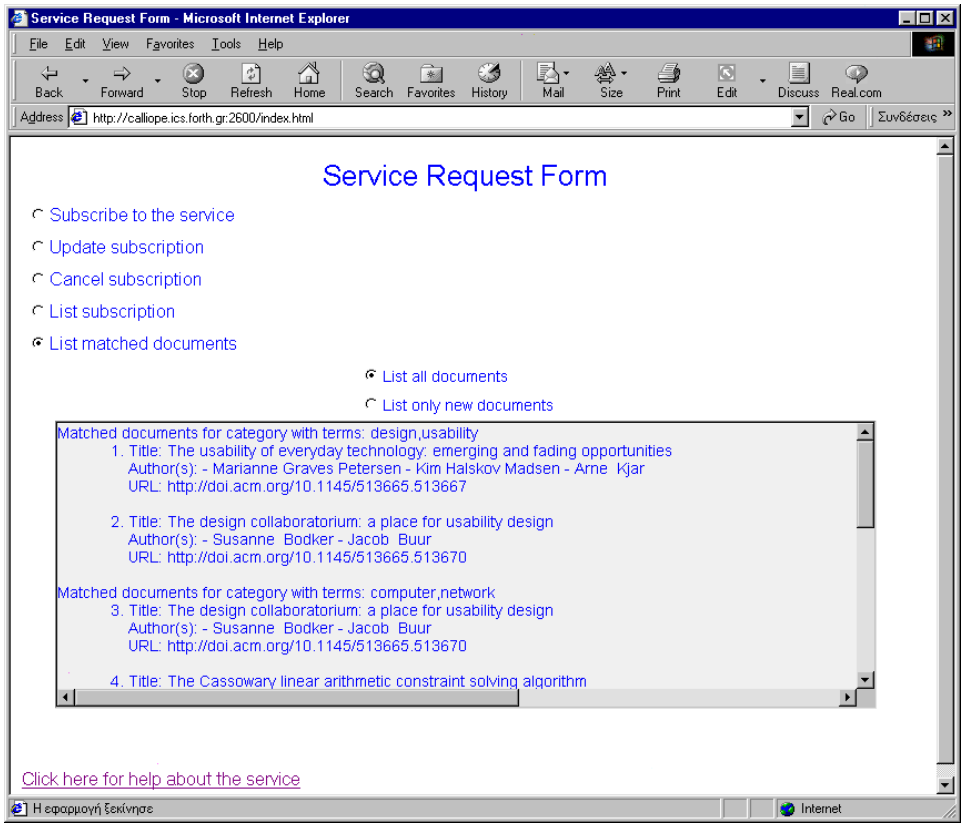
οποία δίνει τη δυνατότητα στο χρήστη να δει τα κείμενα που ταιριάζουν στα διάφορα υπο – προφίλ του, κατηγοριοποιημένα ανά υπο – προφίλ.

Όταν ο χρήστης διαλέξει τη συγκεκριμένη επιλογή, εμφανίζεται η φόρμα στο Σχήμα 31. Παρατηρούμε ότι ο χρήστης μπορεί να επιλέξει να δει είτε όλα τα κείμενα που ταιριάζουν στο προφίλ του, είτε μόνο τα νέα, δηλαδή αυτά που δεν είχαν γίνει match με το προφίλ του την τελευταία φορά που αυτός χρησιμοποίησε το σύστημα.

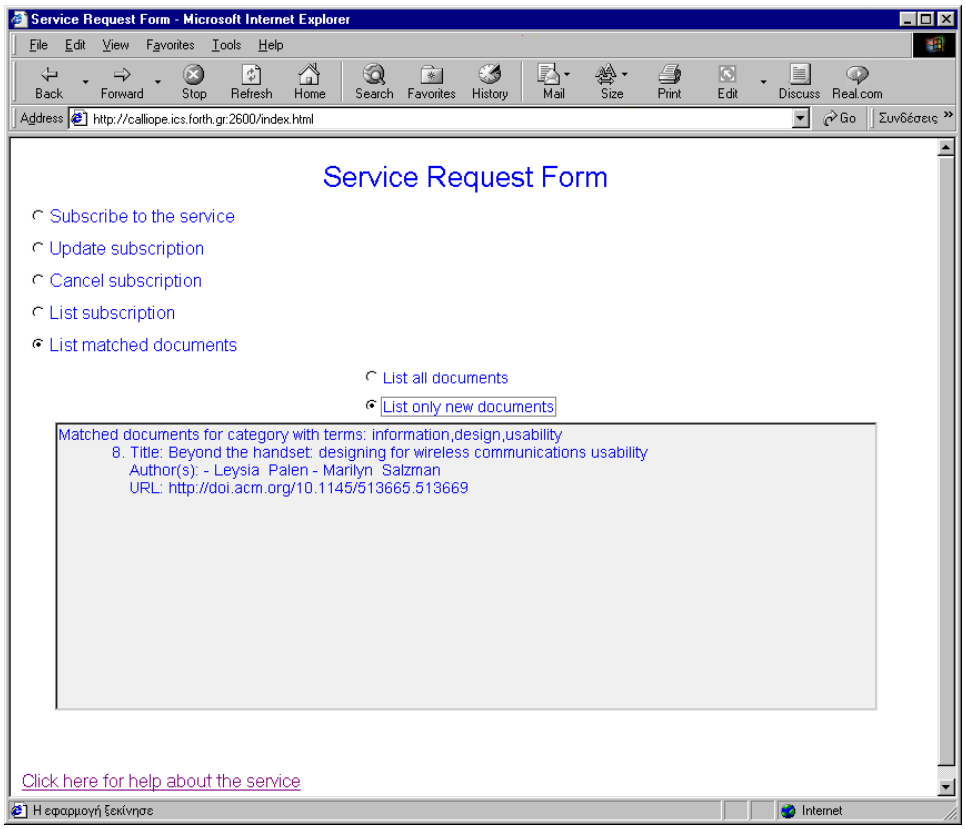


Σχήμα 31. Εμφάνιση κειμένων που ταιριάζουν στο προφίλ χρήστη

Διαλέγοντας την επιλογή "List all documents", εμφανίζεται η φόρμα στο Σχήμα 32, στην οποία παρουσιάζονται όλα τα κείμενα που ταιριάζουν στο προφίλ του χρήστη. Διαλέγοντας την επιλογή "List only new documents", εμφανίζεται η φόρμα στο Σχήμα 33, στην οποία παρουσιάζονται μόνο τα νέα κείμενα που ταιριάζουν στο προφίλ του χρήστη. Αυτά τα κείμενα καλούνται «νέα», καθώς δεν είχαν αποσταλεί στο χρήστη κατά την τελευταία του επικοινωνία με το σύστημα. Και στις δύο περιπτώσεις, όπως προαναφέρθηκε, τα κείμενα είναι κατηγοριοποιημένα ανά υπο – προφίλ.



Σχήμα 32. Εμφάνιση όλων των κειμένων που ταιριάζουν στο προφίλ χρήστη



Σχήμα 33. Εμφάνιση των νέων κειμένων που ταιριάζουν στο προφίλ χρήστη

5.2.7 Επικοινωνία διεπαφής χρήσης με την εφαρμογή

Όταν ο χρήστης ταυτοποιείται από το σύστημα, φορτώνεται αυτόματα το προφίλ του στη διεπαφή χρήσης, εφόσον βέβαια ο χρήστης διατηρεί κάποιο προφίλ. Παράλληλα φορτώνονται και τα μεταδεδομένα των κειμένων που ταιριάζουν στο προφίλ του χρήστη.

Το προφίλ του χρήστη αποστέλλεται στην εφαρμογή (server) και πιο συγκεκριμένα στον χειριστή χρηστών, καθώς αυτός εγκαταλείψει τη διεπαφή χρήσης (δηλαδή είτε όταν αυτός κλείσει τον Web Browser είτε όταν μεταφερθεί σε κάποια άλλη ιστοσελίδα). Ακολούθως πραγματοποιείται όλη η διαδικασία για την επεξεργασία του προφίλ. Έτσι την επόμενη φορά που ο χρήστης επικοινωνήσει με την υπηρεσία, μπορεί να δει όλα τα παλιά και νέα κείμενα που ταιριάζουν στο προφίλ του.

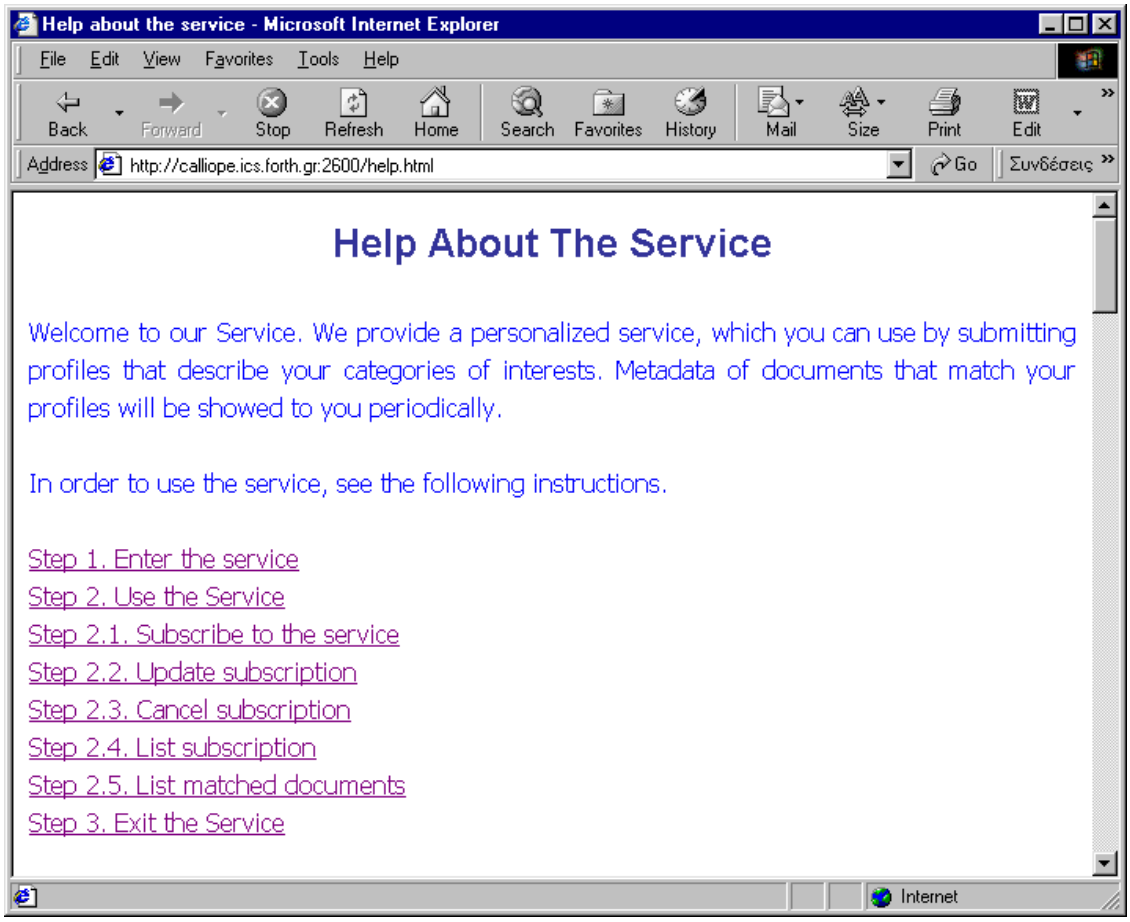
5.2.8 Παρατηρήσεις στη διεπαφή χρήσης του συστήματος

Η διεπαφή χρήσης διαθέτει κάποια χαρακτηριστικά τα οποία την καθιστούν πολύ εύχρηστη και φιλική απέναντι στο χρήστη.

Αρχικά κάθε φορά που ο χρήστης επιθυμεί να αποθηκεύσει κάποια ενέργειά του, επιλέγει το κουμπί "Submit Request". Από την άλλη πλευρά κάθε φορά που επιθυμεί να ακυρώσει κάποια ενέργειά του επιλέγει το κουμπί "Reset". Αλλά και γενικότερα η διεπαφή στηρίζεται σε μία κοινή λογική για την πραγματοποίηση των διαφόρων ενεργειών.

Επιπλέον σε κάθε περίπτωση, το σύστημα προβάλλει τα κατάλληλα μηνύματα λάθους και ειδοποίησης, γεγονός που από τη μία πλευρά οδηγεί το χρήστη στη σωστή χρήση του συστήματος και από την άλλη τον ενημερώνει για την κατάσταση των ενεργειών του. Αξίζει μάλιστα να παρατηρήσουμε ότι η διεπαφή χρήσης προσφέρει «βοήθεια» (Σχήμα 34), την οποία μπορεί να δει ο χρήστης εάν επιλέξει το σύνδεσμο "Click here for help about the service".

Παράλληλα η διεπαφή χρήσης δίνει τη δυνατότητα στο χρήστη να διαχειριστεί με όποιο τρόπο επιθυμεί το προφίλ του, να υλοποιήσει οποιαδήποτε μεταβολή. Τέλος, ο διαχωρισμός των κειμένων σε «όλα» και «νέα» είναι αρκετά βοηθητικός, καθώς προστατεύει το χρήστη από την υπερφόρτωση πληροφορίας.



Σχήμα 34. Βοήθεια για τη χρήση της διεπαφής χρήσης

5.3 Βάση Δεδομένων

Τόσο για την αποθήκευση των κειμένων όσο και για την αποθήκευση των χρηστών του συστήματος και των προφίλ τους χρησιμοποιήθηκε μία βάση δεδομένων. Η βάση δεδομένων αποτελείται από τέσσερις πίνακες για τους χρήστες και τα προφίλ τους και τρεις πίνακες για τα κείμενα. Η αναλυτική περιγραφή των του σχήματος της βάσης και των πινάκων της παρουσιάζεται στο Παράρτημα Α.

Επικοινωνία με τη βάση δεδομένων πραγματοποιείται μόνο από το χειριστή των χρηστών και το χειριστή των κειμένων. Ο χειριστής των κειμένων ανακτά κείμενα από τη βάση με τυχαίο τρόπο, τα οποία όπως προαναφέρθηκε επεξεργάζεται και στη συνέχεια στέλνει στη μηχανή φιλτραρίσματος. Ο χειριστής των χρηστών και λαμβάνει και στέλνει στοιχεία στη βάση. Είναι υπεύθυνος για την εγγραφή και ενημέρωση των στοιχείων των χρηστών και των προφίλ τους στη βάση. Επίσης εάν

αυτό είναι επιθυμητό, φροντίζει να φορτώσει τα προφίλ της βάσης στις δομές του συστήματος.

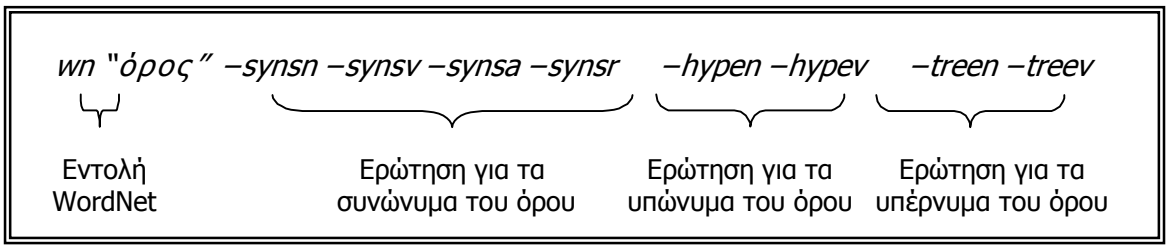
Το σύστημα είναι παραμετρικοποιημένο ως προς τη χρήση της βάσης δεδομένων για τα προφίλ και τους χρήστες. Δηλαδή μπορεί να πραγματοποιηθεί η επιλογή εάν οι χρήστες με τα προφίλ τους θα αποθηκεύονται στη βάση δεδομένων ή απλά θα διατηρούνται στις δομές της κύριας μνήμης. Βεβαίως, στη δεύτερη περίπτωση όλη η πληροφορία σχετικά με τα προφίλ και τους χρήστες χάνεται όταν η λειτουργία του συστήματος σταματήσει.

Παράλληλα στο σύστημα υπάρχουν δύο επιλογές σχετικά με τα προφίλ και τους χρήστες που θα φορτωθούν στο σύστημα από τη βάση δεδομένων κατά την έναρξη της λειτουργίας του συστήματος. Στην πρώτη περίπτωση, το σύστημα φορτώνει όλα τα συγκεκριμένα στοιχεία στις δομές του. Στη δεύτερη περίπτωση, το σύστημα φορτώνει μόνο τα προφίλ των χρηστών, οι οποίοι έχουν επικοινωνήσει με το σύστημα την τελευταία βδομάδα. Βέβαια, εάν κάποιος χρήστης, ο οποίος υπάρχει στη βάση δεδομένων, αλλά όχι στις δομές της κύριας μνήμης, επικοινωνήσει με το σύστημα, τα στοιχεία του φορτώνονται αυτόματα στις δομές.

5.4 Θησαυρός

Για την εφαρμογή της σημασιολογίας στο σύστημα χρησιμοποιήθηκε ένας θησαυρός ευρέως διαδεδομένος, το WordNet [39], το οποίο αποτελεί μία ηλεκτρονική λεξιλογική βάση δεδομένων. Ο σχεδιασμός του έχει στηριχτεί στην υπολογιστική θεωρία της ανθρώπινης μνήμης. Μέσα στο WordNet, αγγλικά ουσιαστικά, ρήματα, επίθετα και επιρρήματα είναι οργανωμένο σε σύνολα συνωνύμων, καθένα από τα οποία αναπαριστά μία έννοια. Διάφοροι σύνδεσμοι ενώνουν τα σύνολα συνωνύμων, διαμορφώνοντας ανάμεσά τους σημασιολογικές σχέσεις, όπως είναι τα υπώνυμα και τα υπερώνυμα.

Από όλες τις σημασιολογικές συσχετίσεις που παρέχει το WordNet, χρησιμοποιήθηκαν μόνο τα συνώνυμα, τα υπώνυμα και τα υπερώνυμα. Κάθε φορά που ο σημασιολογικός χειριστής αναζητά τους σχετικούς όρους μίας λέξης, θέτει την παρακάτω ερώτηση που φαίνεται στο Σχήμα 35 στο WordNet. Η απάντηση αποθηκεύεται σε ένα αρχείο στο λειτουργικό σύστημα, το οποίο στη συνέχεια διαβάζεται από τον σημασιολογικό χειριστή και αξιοποιείται κατάλληλα.



Σχήμα 35. Ερώτηση συστήματος προς το WordNet

Η επιλογή του συγκεκριμένου λεξικού στηρίχτηκε σε τρία κυρίως στοιχεία. Αρχικά το WordNet περιέχει μία ποικιλία όρων και σημασιολογικών σχέσεων ανάμεσά τους, οι οποίες μάλιστα καλύπτουν τις απαιτήσεις του συστήματος. Επιπλέον παρέχει μία διεπαφή "command line", γεγονός που καθιστά εύκολη και εφικτή την επικοινωνία με την εφαρμογή. Τέλος θεωρείται από τους ερευνητές μία από τις πιο σημαντικές πηγές που μπορούν να χρησιμοποιηθούν για τον υπολογισμό linguistics, για την ανάλυση κειμένου και για άλλες σχετικές περιοχές έρευνας.

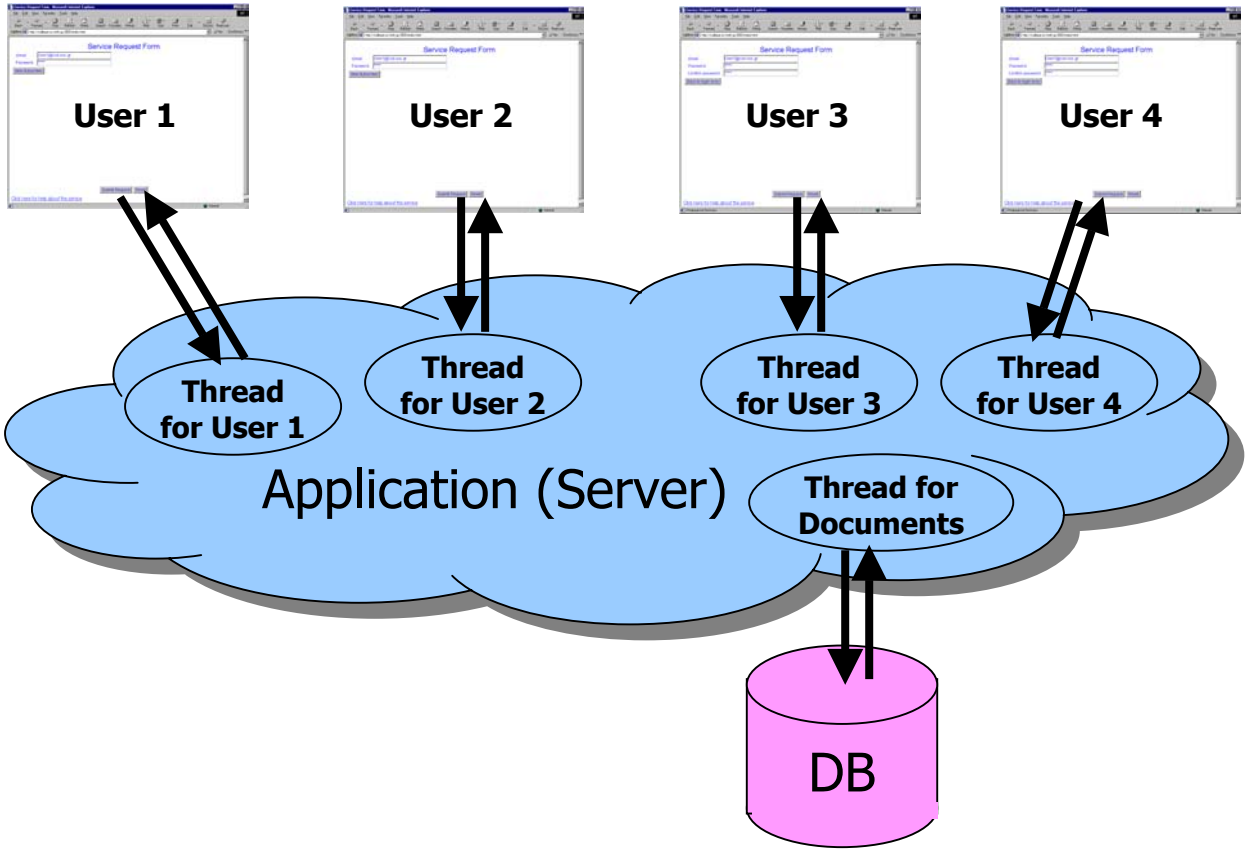
Το σύστημα είναι παραμετρικοποιημένο ως προς τη χρήση του θησαυρού. Δηλαδή μπορεί να λειτουργήσει είτε με είτε χωρίς το θησαυρό, γεγονός το οποίο μας βοηθάει στην αξιολόγηση της εισαγωγής σημασιολογίας στην υπηρεσία μας.

Κεφάλαιο V

Υλοποίηση συστήματος

Η ανάπτυξη και υλοποίηση του συστήματος πραγματοποιήθηκε στη γλώσσα προγραμματισμού Java [40], με χρήση του "The Java Developer Kit" [41] και με τη βοήθεια του εργαλείου Microsoft Visual J++ 6.00 [42]. Όλα τα μέρη του συστήματος, με εξαίρεση το θησαυρό WordNet, υλοποιήθηκαν εξ' αρχής και εξ' ολοκλήρου με χρήση της γλώσσας Java.

Το σύστημα έχει client – server αρχιτεκτονική. Ο server είναι η εφαρμογή, η οποία εξυπηρετεί όλες τις αιτήσεις των πελατών (clients) και η οποία υλοποιεί όλες τις λειτουργίες του συστήματος. Ο client είναι η διεπαφή χρήσης, η οποία στέλνει αιτήσεις και αποδέχεται απαντήσεις από το server.



Σχήμα 36. Πολυνηματική εφαρμογή πολλών χρηστών

7.1 Εφαρμογή (Server)

Ο server αποτελεί μία πολυνηματική (multithreaded) εφαρμογή πολλών χρηστών (multi-user), όπως παρουσιάζεται και στο Σχήμα 36. Η εφαρμογή σχεδιάστηκε και υλοποιήθηκε ως πολυνηματική, καθώς οφείλει να εκτελεί πολλές διαφορετικές λειτουργίες ταυτόχρονα και ως εφαρμογή πολλών χρηστών, καθώς πρέπει να εξυπηρετεί ταυτόχρονα πολλούς χρήστες.

Για κάθε πελάτη η εφαρμογή γεννά ένα νήμα (thread), το οποίο τον εξυπηρετεί. Αυτό το νήμα είναι υπεύθυνο για την εκτέλεση όλης της επεξεργασίας και όλων των λειτουργιών για το συγκεκριμένο χρήστη και τα προφίλ του. Παράλληλα, η εφαρμογή δημιουργεί ένα άλλο νήμα, το οποίο είναι υπεύθυνο για τα κείμενα. Αυτό το thread φροντίζει για την εκτέλεση όλων των λειτουργιών που σχετίζονται με τα κείμενα του συστήματος.

Η σχεδίαση και υλοποίηση της εφαρμογής είναι απόλυτα οντοκεντρική, γεγονός που αποτέλεσε και απαίτηση λόγω της χρήσης της συγκεκριμένης γλώσσας προγραμματισμού. Μία συνοπτική περιγραφή των βασικών αντικειμένων που υλοποιήθηκαν παρουσιάζεται στο Παράρτημα Β.

7.2 Διεπαφή χρήσης (Client)

Η διεπαφή χρήσης δημιουργήθηκε με βάση την τεχνολογία "Java Applet" [43]. Κάθε χρήστης που χρησιμοποιεί την υπηρεσία φορτώνει το applet στον Web browser του, το οποίο υλοποιεί την επικοινωνία με την εφαρμογή. Πιο συγκεκριμένα, για κάθε χρήστη ο server «ανοίγει» ένα socket [44] μέσω του οποίου το applet στέλνει δεδομένα στο server και αντίστροφα. Τα δεδομένα που στέλνονται και προς τις δύο κατευθύνσεις ακολουθούν ένα πρωτόκολλο επικοινωνίας, με αποτέλεσμα πάντα να στέλνονται και να λαμβάνονται τα αναμενόμενα δεδομένα.

Ο μόνος περιορισμός που θέτει η χρήση της τεχνολογίας "Java Applet" είναι η ύπαρξη ενός HTTP server στο ίδιο μηχάνημα όπου υπάρχει ο server και από το οποίο «σηκώνεται» το applet. Για το λόγο αυτό επιλέχθηκε ο HTTP server httpd, ο οποίος ικανοποιεί τις απαιτήσεις του συστήματός μας.

7.3 Βάση δεδομένων

Η βάση δεδομένων που χρησιμοποιήθηκε είναι η ORACLE8*i* [45], καθώς χαρακτηρίζεται ιδιαίτερα αξιόπιστη για την αποθήκευση μεγάλου όγκου δεδομένων. Για την επικοινωνία της βάσης δεδομένων με την εφαρμογή χρησιμοποιήθηκε η τεχνολογία JDBC [46]. Με χρήση της συγκεκριμένης τεχνολογίας είναι δυνατή η διαχείριση των δεδομένων μίας βάσης δεδομένων μέσα από οποιαδήποτε εφαρμογή σε γλώσσα προγραμματισμού Java. Έτσι, η εφαρμογή μας μπορεί δυναμικά να δημιουργεί πίνακες στη βάση δεδομένων, να εισάγει τιμές σε αυτούς, να μεταβάλλει τις τιμές των πεδίων των πινάκων, να θέτει επερωτήσεις στη βάση και να ανακτά τα αποτελέσματα.

7.4 Παραμετροποίηση συστήματος

Όπως έχει αναφερθεί σε πολλά σημεία της παρούσας εργασίας το σύστημα είναι παραμετρικοποιημένο. Σε αυτό το σημείο θα αναφέρουμε συγκεντρωτικά τις δυνατότητες κατάστασης εκτέλεσης του συστήματος που υπάρχουν.

Το σύστημα λαμβάνει συνολικά πέντε παραμέτρους. Η πρώτη παράμετρος αναφέρεται στο πλήθος των κειμένων που θα φορτωθούν από τη βάση δεδομένων των κειμένων. Τα κείμενα αυτά επιλέγονται με ένα ειδικό μηχανισμό τυχαία από τη βάση δεδομένων.

Η δεύτερη παράμετρος αναφέρεται στη μεθοδολογία που θα χρησιμοποιηθεί για την ανάκτηση των σχετικών όρων. Εάν η συγκεκριμένη παράμετρος λάβει την τιμή 1, τότε χρησιμοποιείται η δομή σχετικών όρων, στην οποία εισάγονται στοιχεία κάθε φορά που εμφανίζεται ένα κείμενο ή προφίλ. Σε αυτήν την περίπτωση, κατά τη διάρκεια του ταιριάσματος, το σύστημα ανακτά τους σχετικούς όρους από τη δομή. Εάν αυτή η παράμετρος λάβει την τιμή 2, τότε δε χρησιμοποιείται η εν λόγω δομή, αλλά κατά τη διάρκεια του ταιριάσματος το σύστημα ρωτά απευθείας το θησαυρό και χρησιμοποιεί άμεσα την απάντησή του, αποθηκεύοντάς την προσωρινά.

Η δεύτερη παράμετρος ορίζει εάν το σύστημα θα χρησιμοποιήσει το θησαυρό (τιμή παραμέτρου ίση με 1) ή όχι (τιμή παραμέτρου ίση με 2), εάν δηλαδή με άλλα λόγια ληφθεί υπόψη η σημασιολογία των όρων των προφίλ και των κειμένων.

Οι δύο τελευταίοι παράμετροι αναφέρονται στη βάσης δεδομένων των προφίλ. Η πρώτη ορίζει ή όχι τη χρήση της συγκεκριμένης βάση δεδομένων, τόσο την ανάκτηση πληροφορίας από αυτή, όσο και την αποθήκευση πληροφορίας σε αυτήν. Η βάση δεδομένων χρησιμοποιείται όταν η συγκεκριμένη παράμετρος έχει τιμή 1, ενώ δε χρησιμοποιείται για την τιμή 2. Τέλος, η δεύτερη παράμετρος ορίζει εάν θα φορτωθούν προφίλ από τη βάση δεδομένων του συστήματος. Εάν έχει την τιμή 0, δε φορτώνεται κανένα προφίλ από τη βάση στις δομές της κύριας μνήμης., ενώ εάν έχει την τιμή 1, φορτώνονται όλα τα προφίλ της βάσης στο σύστημα. Εάν έχει την τιμή 2, τότε φορτώνονται τα "most recently used" προφίλ, δηλαδή αυτά που έχουν προσπελαθεί από τους χρήστες τους στη διάρκεια της τελευταίας βδομάδας.

Η παραμετρικοποίηση του συστήματος πραγματοποιήθηκε ουσιαστικά για την διεξαγωγή πειραμάτων, τα οποία παρουσιάζονται στη συνέχεια.

Κεφάλαιο VI

Πειράματα

8.1 Δεδομένα πειραμάτων

Για την πραγματοποίηση πειραμάτων στο σύστημα επιλέχθηκαν τα κείμενα των ACM Transactions, τα οποία διέθεταν τα απαραίτητα μεταδεδομένα. Με τη χρήση ενός script, οι σελίδες των κειμένων ανακτήθηκαν, επεξεργάστηκαν κατάλληλα και τα μεταδεδομένα αποθηκεύτηκαν στη βάση δεδομένων. Τα κείμενα που δεν είχαν λέξεις – κλειδιά απορρίφθηκαν, με αποτέλεσμα τελικά να παραμείνουν περίπου 1650 κείμενα στη βάση δεδομένων.

Η εισαγωγή των προφίλ έγινε χειροκίνητα (manually) από τη διεπαφή χρήσης. Κάποια από τα προφίλ που εφαρμόστηκαν στο σύστημα φαίνονται στον παρακάτω πίνακα:

Κατηγορία Ενδιαφερόντων	**Τύπος**	**Κατώφλι**
relational database	Boolean	
database, query	Boolean	
query, relational database	Boolean	
distributed, relational database, query	Boolean	
parallel, relational database, query, distributed	Boolean	
query, distributed, database	Boolean	
computer	Boolean	
computer, design	Boolean	
computer, design, usability	Boolean	
computer, design, not usability	Boolean	
computer $<_{[0,1]}$ design	Boolean	
computer $<_{[0,100]}$ design	Boolean	
relational database	Weighted	0.1
relational database, query	Weighted	0.1
relational database, query, distributed	Weighted	0.1

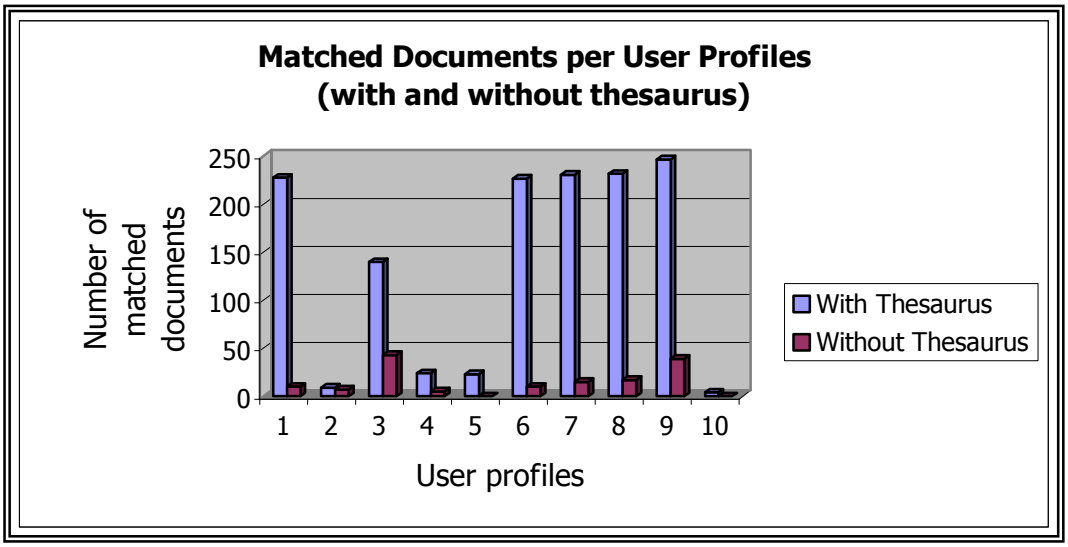
relational database, query, distributed, parallel	Weighted	0.1
relational database	Weighted	0.5
relational database, query	Weighted	0.5
relational database, query, distributed	Weighted	0.5
relational database, query, distributed, parallel	Weighted	0.5
relational database	Weighted	0.9
relational database, query	Weighted	0.9
relational database, query, distributed	Weighted	0.9
relational database, query, distributed, parallel	Weighted	0.9
design, usability	Boolean	
designing, usability	Boolean	
knowledge, usability	Boolean	
Plan, usability	Boolean	
layout, usability	Boolean	
computer, network	Boolean	
information, design $<_{[0,12]}$ usability	Boolean	
information, design, usability	Boolean	
design $<_{[0,12]}$ usability	Boolean	

Παρατηρούμε ότι οι παραπάνω κατηγορίες ενδιαφερόντων καλύπτουν όλες τις δυνατές μορφές που μπορεί να λάβει ένα προφίλ. Στο συγκεκριμένο σύνολο προφίλ εμφανίζονται τόσο προφίλ του μοντέλου Boolean (απλά, με τελεστή άρνησης, με συνθήκη εγγύτητας), όσο και προφίλ του μοντέλου Vector Space (με διαφορετικές τιμές κατωφλίου). Με αυτόν τον τρόπο μπορούμε να ισχυριστούμε ότι τα πειραματικά μας αποτελέσματα είναι αρκετά αντιπροσωπευτικά όσον αφορά τουλάχιστον τα προφίλ του συστήματος.

8.2 Πειραματικά αποτελέσματα

Τα πειράματα που πραγματοποιήθηκαν πάνω στο σύστημα έχουν μία σειρά από στόχους. Πρωταρχικός σκοπός είναι να αποδειχθεί ότι το συγκεκριμένο σύστημα όντως ικανοποιεί την έννοια της σημασιολογίας, δηλαδή ότι τελικά ο χρήστης λαμβάνει κείμενα, τα οποία περιέχουν όχι μόνο τους όρους του προφίλ του αλλά και

όρους συνώνυμα ή υπώνυμα των όρων αυτών. Το παρακάτω γράφημα καθώς και ο πίνακας με τα αποτελέσματα αποδεικνύουν ότι ο συγκεκριμένος στόχος υλοποιήθηκε.



Σχήμα 37. Πλήθος κειμένων που ταιριάζουν στα προφίλ με και χωρίς τη χρήση θησαυρού

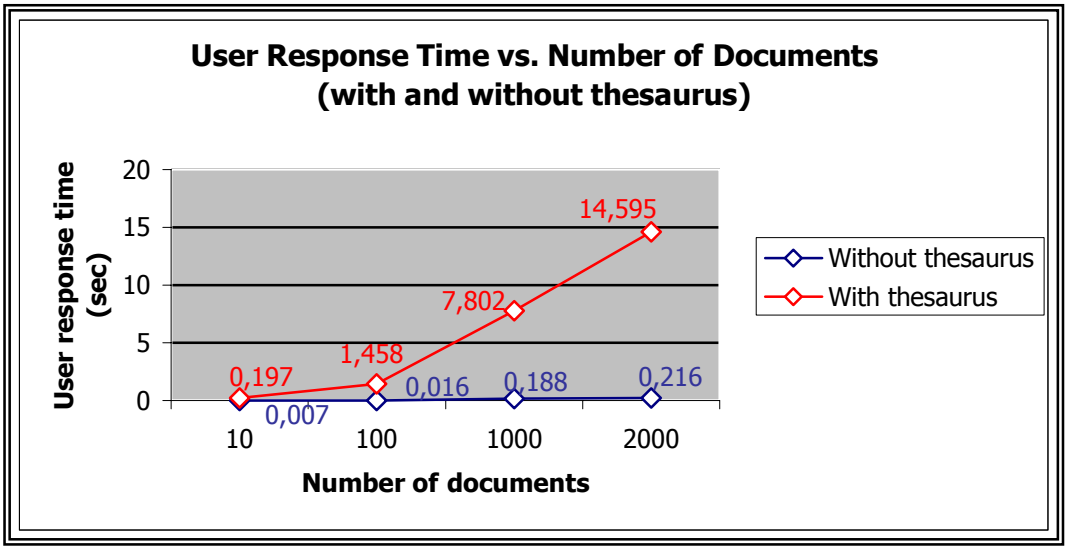
Id	Profiles	Type - [Threshold]	Matched Documents	
			With Thesaurus	Without Thesaurus
1	relational database	BM	228	10
2	database, query	BM	9	7
3	computer	BM	140	43
4	computer, design	BM	24	5
5	computer, design, not usability	BM	23	0
6	Relational database	VSM - 0.1	227	10
7	Relational database, query	VSM - 0.1	231	15
8	Relational database, query, distributed	VSM - 0.1	232	17
9	Relational database, query, distributed, parallel	VSM - 0.1	247	39
10	plan, usability	BM	4	0

Το συγκεκριμένο πείραμα περιλαμβάνει δύο καταστάσεις λειτουργίας του συστήματος. Στην πρώτη κατάσταση υπάρχει επικοινωνία με το θησαυρό WordNet, δηλαδή με άλλα λόγια λαμβάνεται υπόψη η σημασιολογική σχετικότητα των όρων. Χρησιμοποιούνται 1000 κείμενα και τα 10 προφίλ χρηστών που φαίνονται παραπάνω. Το πλήθος των κειμένων που ταιριάζουν σε καθένα από τα προφίλ φαίνεται στην τρίτη στήλη του παραπάνω πίνακα ("With Thesaurus") και απεικονίζεται στο

γράφημα με τις μπλε μπάρες (μία για κάθε προφίλ). Στη δεύτερη κατάσταση λειτουργίας τους συστήματος δε λαμβάνεται υπόψη η σημασιολογία των όρων των προφίλ και των κειμένων. Σε αυτήν την περίπτωση χρησιμοποιήθηκαν τα ίδια 1000 κείμενα και 10 προφίλ. Το πλήθος των κειμένων που ταιριάζουν τα 10 προφίλ παρουσιάζεται στην τέταρτη στήλη του παραπάνω πίνακα ("Without Thesaurus") και απεικονίζεται με τις κόκκινες μπάρες στο γράφημα αντίστοιχα.

Με βάση τις συγκεκριμένες μετρήσεις, παρατηρούμε ότι στην περίπτωση που χρησιμοποιείται ο θησαυρός WordNet, τα κείμενα που επιστρέφονται στον χρήστη είναι πολύ περισσότερα από την περίπτωση που ελέγχεται απλά η ταυτοσημότητα ανάμεσα στους όρους του προφίλ και του κειμένου (χωρίς δηλαδή τη χρήση του θησαυρού). Αυτό συμβαίνει καθώς στην πρώτη περίπτωση ο χρήστης λαμβάνει και κείμενα τα οποία περιέχουν και σημασιολογικά σχετικούς όρους των όρων του προφίλ του. Επομένως ο βασικός στόχος του συστήματος επιτεύχθηκε.

Στο επόμενο γράφημα (Σχήμα 38) παρουσιάζονται οι χρόνοι απόκρισης του συστήματος ανά προφίλ στις δύο παραπάνω καταστάσεις λειτουργίας. Η μπλε γραμμή δείχνει τον μέσο χρόνο απόκρισης του συστήματος ανά προφίλ όταν δεν υπάρχει επικοινωνία με το θησαυρό, στις περιπτώσεις που έχουμε 10, 100, 1000, 2000 κείμενα στο σύστημά μας. Η κόκκινη γραμμή, παρουσιάζει τον αντίστοιχο χρόνο για τους ίδιους αριθμούς κειμένων, με τη μόνη διαφορά ότι το σύστημα επικοινωνεί με το θησαυρό.

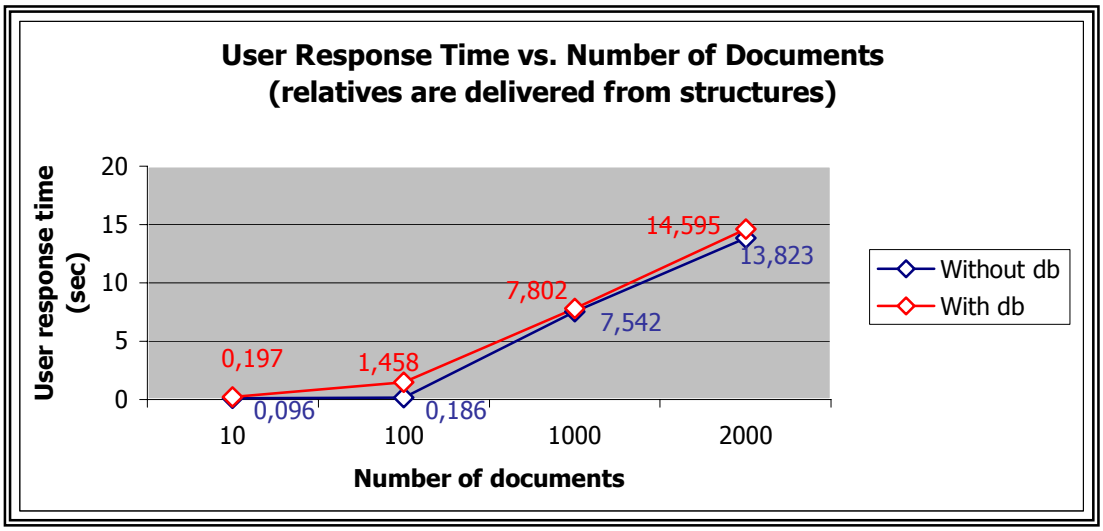


Σχήμα 38. Απόκριση συστήματος προς τον αριθμό των κειμένων με και χωρίς τη χρήση θησαυρού

Όπως παρατηρούμε, η χρήση του θησαυρού παρουσιάζει αρκετά μεγάλη καθυστέρηση στην απόδοση του συστήματος. Χωρίς το θησαυρό, ο χρήστης λαμβάνει απάντηση από το σύστημα σε μικρότερο χρόνο από την περίπτωση που το σύστημα χρησιμοποιεί το θησαυρό. Αυτό το γεγονός είναι πολύ λογικό, καθώς η επικοινωνία της εφαρμογής με το θησαυρό είναι πολύ αργή σε σχέση με την υπόλοιπη λειτουργία του συστήματος.

Ένα δεύτερο πείραμα ελέγχει την απόδοση του συστήματος, όπως αυτή γίνεται αντιληπτή από την πλευρά του χρήστη (δηλαδή το user response time), στις δύο διαφορετικές μεθοδολογίες επικοινωνίας με το θησαυρό. Για τη διεξαγωγή του πειράματος χρησιμοποιούνται 30 προφίλ χρηστών και 10, 100, 1000, 2000 κείμενα διαδοχικά. Το συγκεκριμένο πείραμα περιλαμβάνει τρεις καταστάσεις λειτουργίας του συστήματος.

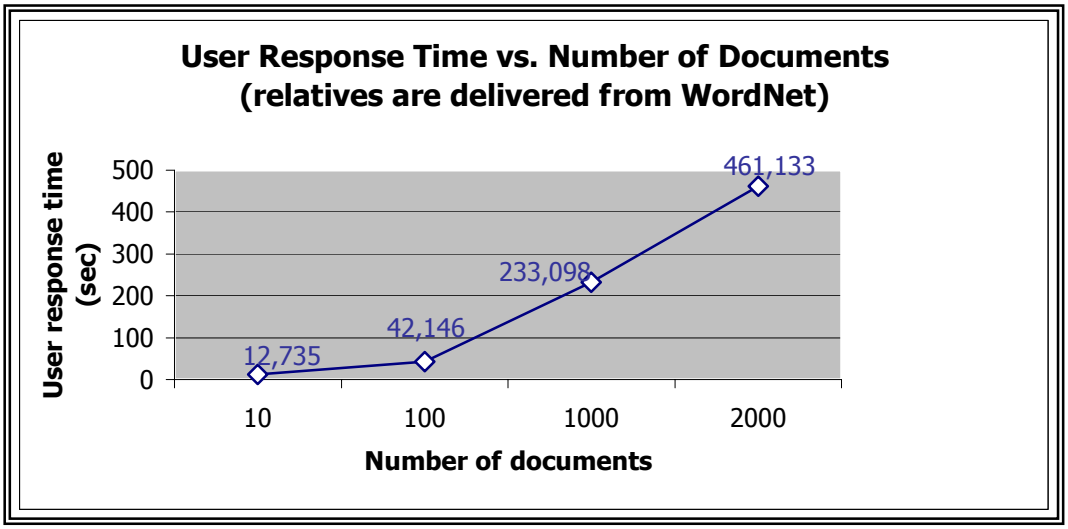
Στην πρώτη κατάσταση, χρησιμοποιείται η δομή ευρύτερων όρων, άρα η επικοινωνία με το θησαυρό πραγματοποιείται κάθε φορά που εμφανίζεται ένα νέο κείμενο ή προφίλ στο σύστημα. Ωστόσο, τα προφίλ των χρηστών δεν αποθηκεύονται στη βάση δεδομένων, αλλά διατηρούνται απλά στις δομές της κύριας μνήμης. Ο μέσος όρος χρόνου απόκρισης ανά προφίλ για τη συγκεκριμένη κατάσταση λειτουργίας αναπαρίσταται με την κόκκινη γραμμή στο γράφημα στο Σχήμα 39, για 10, 100, 1000, 2000 κείμενα αντίστοιχα.



Σχήμα 39. Απόκριση συστήματος προς τον αριθμό των κειμένων με και χωρίς τη βάσης δεδομένων για τα προφίλ. Οι σχετικοί όροι κατά τη σύγκριση ανακτώνται από τη δομή σχετικών όρων

Η δεύτερη κατάσταση λειτουργίας είναι όμοια με την πρώτη με τη μόνη διαφορά ότι χρησιμοποιείται η βάση δεδομένων για την αποθήκευση των στοιχείων των χρηστών και των προφίλ τους. Ο μέσος χρόνος απόκρισης του συστήματος ανά προφίλ χρήστη συμβολίζεται με την μπλε γραμμή του ίδιου γραφήματος (Σχήμα 39). Παρατηρούμε ότι η χρήση της βάσης δεδομένων κοστίζει μία πολύ μικρή καθυστέρηση στην απόδοση του συστήματος.

Στην τρίτη κατάσταση λειτουργίας του συστήματος η επικοινωνία με το θησαυρό πραγματοποιείται κατά τη διάρκεια του ταιριάσματος και δε χρησιμοποιείται η δομή των σχετικών όρων. Παράλληλα δε χρησιμοποιείται η βάση δεδομένων για την αποθήκευση των προφίλ των χρηστών. Οι μέσοι όροι χρόνων απόκρισης ανά προφίλ για 10, 100, 1000 και 2000 κείμενα εμφανίζονται στο γράφημα στο Σχήμα 40.



Σχήμα 40. Απόκριση συστήματος προς τον αριθμό των κειμένων. Οι σχετικοί όροι κατά τη σύγκριση ανακτώνται απευθείας από το θησαυρό WordNet και δεν αποθηκεύονται στη δομή σχετικών όρων.

Παρατηρούμε ότι στην περίπτωση που χρησιμοποιείται η δομή σχετικών όρων, ο χρόνος που απαιτείται για να δώσει το σύστημα την απόκριση στο χρήστη είναι αισθητά μικρότερος από την περίπτωση που το σύστημα ρωτάει το θησαυρό κατά τη διάρκεια του ταιριάσματος. Πρέπει βέβαια να σημειωθεί ότι η ταχύτητα στην απόκριση του συστήματος έχει το κόστος της κατανάλωσης μεγαλύτερου χώρου αποθήκευσης. Στην περίπτωση που χρησιμοποιούνται οι δομές, δεσμεύεται χώρος της μνήμης για την αποθήκευσή τους.

Οι χρόνοι απόκρισης του συστήματος ανά προφίλ που σχεδιάζονται στα δύο τελευταία γραφήματα συνοψίζονται και στον παρακάτω πίνακα. Κάθε γραμμή του πίνακα αντιστοιχεί σε μία κατάσταση λειτουργίας του συστήματος, ενώ κάθε στήλη αναφέρει το πλήθος των κειμένων που χρησιμοποιήθηκαν σε κάθε εκτέλεση. Οι τιμές του πίνακα αναφέρονται στο μέσο χρόνο (σε seconds) που απαιτείται για την επεξεργασία ενός προφίλ χρήστη, δηλαδή στο μέσο χρόνο απόκρισης του συστήματος ανά προφίλ.

	Number of Documents			
Mode of operation	**10**	**100**	**1000**	**2000**
Relatives from structures - Without db	0.096	0.186	7.542	13.823
Relatives from structures - With db	0.197	1.458	7.802	14.595
Relatives from WordNet	12.735	42.146	233.098	461.133

8.3 Συμπεράσματα από τα πειραματικά αποτελέσματα

Τα πειραματικά αποτελέσματα που παρουσιάστηκαν παραπάνω παρέχουν τη δυνατότητα εξαγωγής χρήσιμων συμπερασμάτων για το σύστημα.

Αρχικά, με τη χρήση της σημασιολογικής συσχέτισης ανάμεσα στους όρους των προφίλ και των κειμένων αυξάνεται η ποιότητα της υπηρεσίας που παρέχεται από το σύστημα. Ο χρήστης λαμβάνει περισσότερα κείμενα σχετικά με το προφίλ του, πολλά από τα οποία δε θα είχε λάβει διαφορετικά. Πιο συγκεκριμένα λαμβάνει κείμενα τα οποία περιέχουν τους όρους του προφίλ του, αλλά και κείμενα που περιέχουν σχετικούς όρους (συνώνυμα ή υπώνυμα) των όρων του προφίλ του.

Για να ελεγχθεί η σημασιολογική συσχέτιση των όρων απαιτείται η επικοινωνία του συστήματος με κάποιον θησαυρό. Η συγκεκριμένη επικοινωνία κοστίζει κάποια καθυστέρηση στο χρόνο απόκρισης του συστήματος. Αυτό το γεγονός ωστόσο δεν αποτελεί τροχοπέδη για την υλοποίηση ενός τέτοιου συστήματος, λόγω της φύσης της υπηρεσίας που παρέχεται. Σε ένα σύστημα επιλεκτικής διασποράς πληροφορίας, ο χρήστης δεν περιμένει άμεση απόκριση από το σύστημα, όπως γίνεται για παράδειγμα στις κλασσικές μηχανές αναζήτησης. Αντίθετα αποστέλλει το προφίλ του και κάποια στιγμή ενημερώνεται για τα κείμενα που

ταιριάζουν με αυτό. Επομένως η συγκεκριμένη καθυστέρηση, η οποία μάλιστα κυμαίνεται σε λογικά χρονικά πλαίσια, δεν αποτελεί πρόβλημα.

Αύξηση στο χρόνο απόκρισης του συστήματος προκαλεί και η χρήση της βάσης δεδομένων για την αποθήκευση των στοιχείων του χρήστη και των προφίλ του. Επειδή η συγκεκριμένη μείωση είναι μικρή, προτείνεται η χρήση της συγκεκριμένης βάσης, έτσι ώστε τα προφίλ των χρηστών να μη χάνονται σε περίπτωση διακοπής της λειτουργίας του συστήματος.

Τέλος, προτείνεται η χρήση της δομής σχετικών όρων και η επικοινωνία του συστήματος με το θησαυρό τη στιγμή που εμφανίζεται κάποιο προφίλ ή κείμενο και όχι τη στιγμή που πραγματοποιείται η σύγκριση. Αυτή η απόφαση προκύπτει από τη μεγάλη απόκλιση των χρόνων απόκρισης στις δύο παραπάνω περιπτώσεις. Πρέπει να σημειωθεί μάλιστα ότι όταν χρησιμοποιείται η δομή των σχετικών όρων, ο θησαυρός δεν ερωτάται για όρους οι οποίοι ήδη υπάρχουν στη δομή. Επομένως ο χρόνος απόκρισης του συστήματος μειώνεται ακόμη περισσότερο στην περίπτωση που παρουσιάζονται επικαλυπτόμενα προφίλ στο σύστημα, δηλαδή προφίλ με ίδιους όρους. Βέβαια με τη συγκεκριμένη μεθοδολογία υπάρχει σπατάλη χώρου αποθήκευσης. Οπότε σε κάθε περίπτωση μπορεί να γίνει η επιλογή ανάμεσα στην εξοικονόμηση χώρου ή χρόνου ("Trade – off: time vs. space").

Με βάση λοιπόν τα πειραματικά αποτελέσματα, καταλήγουμε στο συμπέρασμα ότι το σύστημα έχει δύο βασικά πλεονεκτήματα. Αρχικά προσφέρει ακρίβεια όσον αναφορά τα σχετικά κείμενα που αποστέλλει στους χρήστες και επομένως καλή ποιότητα υπηρεσίας. Παράλληλα παρουσιάζει καλούς χρόνους απόκρισης, δεδομένης πάντα της φύσης της υπηρεσίας που παρέχεται.

Συμπεράσματα

Στόχος της παρούσας εργασίας αποτελεί η δημιουργία ενός συστήματος επιλεκτικής διασποράς πληροφορίας το οποίο λαμβάνει υπόψη τη σημασιολογία των όρων που εμφανίζονται στα κείμενα και στα προφίλ. Το σύστημα που υλοποιήθηκε υλοποιεί το συγκεκριμένο στόχο. Ειδικά μετά την διεξαγωγή των πειραμάτων, αποδείχθηκε ότι τα κείμενα που επιστρέφονται σε κάθε περίπτωση στο χρήστη είναι περισσότερα από αυτά που θα του επιστρέφονταν χωρίς τη χρήση του θησαυρού. Σημαντικό στοιχείο αποτελεί το γεγονός ότι αυτά τα κείμενα δεν οδηγούν σε υπερφόρτωση πληροφορίας του χρήστη, καθώς είναι απολύτως σχετικά με το προφίλ του.

Παράλληλα το σύστημα παρουσιάζει αρκετά καλή απόδοση, καθώς ακόμα και με μεγάλο αριθμό κειμένων και προφίλ, επιστρέφει σε μικρό χρόνο απάντηση στο χρήστη (αναφερόμαστε προφανώς στην περίπτωση που χρησιμοποιείται η δομή των σχετικών όρων, η οποία είναι και η προτεινόμενη μεθοδολογία). Μάλιστα η καθυστέρηση που παρουσιάζεται, οφείλεται κυρίως στην επικοινωνία με το θησαυρό WordNet. Πιθανότατα με τη χρήση κάποιου άλλου θησαυρού, ο χρόνος απόκρισης του συστήματος να ήταν αισθητά μικρότερος.

Ένα σημαντικό στοιχείο του συστήματος αποτελεί και η κοινή προσέγγιση που υλοποιήθηκε για τα δύο μοντέλα αναπαράστασης προφίλ, δηλαδή για το Boolean μοντέλο και το Vector Space μοντέλο. Αυτό επιτυγχάνεται με τη χρήση μίας κοινής δομής ευρετηρίασης για τα δύο μοντέλα, αλλά και μίας κοινής μεθοδολογίας για την εύρεση των κειμένων που ταιριάζουν στα προφίλ. Παράλληλα, ενώ ο χρήστης μπορεί να αποστείλει στο σύστημα υπο – προφίλ και των δύο μορφών, δεν ασχολείται με τις λεπτομέρειες του κάθε μοντέλου. Το σύστημα είναι αυτό που φροντίζει για την κατάλληλη αναπαράσταση των υπο – προφίλ, αλλά και των κειμένων σε κάθε περίπτωση.

Τέλος, αξίζει να σημειωθεί ότι τα περισσότερα συστήματα που αυτή τη στιγμή υπάρχουν στο χώρο της ανάκτησης και της διασποράς πληροφορίας δε λαμβάνουν υπόψη τη σημασιολογία των όρων. Ελέγχουν απλά την λεκτική ταυτοσημότητα ανάμεσα στους όρους των προφίλ και των κειμένων, με αποτέλεσμα από τη μία πλευρά να παρουσιάζουν στους χρήστες κείμενα που δεν ανταποκρίνονται στα

ενδιαφέροντά τους, και από την άλλη πλευρά να αποκρύβουν σχετικά κείμενα από αυτούς.

Η αξιολόγηση και σύγκριση συστημάτων επιλεκτικής διασποράς πληροφορίας πραγματοποιείται συνήθως με βάση δύο μετρικές, την ακρίβεια (precision) και την ανάκληση (recall) [38]. Η ακρίβεια ορίζεται ως το ποσοστό των κειμένων τα οποία επιστρέφονται στο χρήστη και τα οποία είναι πραγματικά σχετικά με το προφίλ του. Η ανάκληση ισούται με το λόγο του πλήθους των σχετικών κειμένων που επιστρέφονται στο χρήστη προς το συνολικό πλήθος των σχετικών κειμένων που υπάρχουν στη συλλογή.

Με βάση τις συγκεκριμένες μετρικές, το σύστημά μας παρουσιάζεται ιδιαίτερα αποτελεσματικό, καθώς επιστρέφει στο χρήστη όλα τα κείμενα που περιέχουν τους όρους του προφίλ του ή όρους σχετικούς με αυτούς. Στην περίπτωση ειδικά που το προφίλ του χρήστη περιλαμβάνει ικανοποιητικό πλήθος όρων, τα κείμενα που αυτός λαμβάνει είναι σε μεγάλο ποσοστό σχετικά με τα ενδιαφέροντά του, δηλαδή το σύστημα έχει αρκετά υψηλή τιμή στην ακρίβεια. Παράλληλα πρέπει να τονιστεί ότι το σύστημα διαχειρίζεται τους σύνθετους όρους, δηλαδή τις φράσεις, ως μία οντότητα, γεγονός το οποίο επίσης οδηγεί στην αύξηση της ακρίβειας της υπηρεσίας που παρέχει.

Αντίθετα το SIFT, το οποίο αποτελεί το πιο συγγενές σύστημα, δεν είναι τόσο αποτελεσματικό. Πιο συγκεκριμένα το SIFT προσεγγίζει την υλοποίηση του συστήματός μας χωρίς τη χρήση του θησαυρού. Όπως φάνηκε και από το πρώτο πείραμα που περιγράφηκε, το πλήθος των σχετικών κειμένων που αποστέλλονται στους χρήστες είναι μεγαλύτερο όταν χρησιμοποιείται ο θησαυρός. Επομένως οι τιμές των δύο μετρικών είναι καλύτερες και το σύστημά μας παρουσιάζεται τελικά πιο αποτελεσματικό από το SIFT.

Καταλήγοντας, πρέπει να τονιστεί ότι η υλοποίηση ενός συστήματος που χρησιμοποιεί τη σημασιολογία των όρων αποτελεσματικά αποτελεί ένα βασικό γεγονός που πρέπει να ληφθεί υπόψη για τη δημιουργία και άλλων τέτοιων συστημάτων.

Μελλοντική Εργασία

Ο τρόπος σχεδίασης και ανάπτυξης του συστήματος θέτει εύκολη την επέκτασή του και την προσθήκη επιπλέον λειτουργικότητας σε αυτό. Αυτό οφείλεται κυρίως στο γεγονός της κατάτμησης του συστήματος σε κομμάτια που λειτουργούν ανεξάρτητα, ενώ παράλληλα επικοινωνούν μεταξύ τους.

Μία μέθοδος που μπορεί να χρησιμοποιηθεί για την αύξηση της απόδοσης του συστήματος είναι η ανάδραση σχετικότητας (relevance feedback) [47]. Η συγκεκριμένη τεχνική απαιτεί από το χρήστη να «βαθμολογήσει» τα κείμενα που του αποστέλλονται, με βάση το πόσο σχετικά τα θεωρεί ο ίδιος με το προφίλ του. Η ανάδραση αυτή από το χρήστη μπορεί να χρησιμοποιηθεί αποτελεσματικά και να βελτιώσει την ακρίβεια του συστήματος.

Με βάση τη συγκεκριμένη μεθοδολογία, το διάνυσμα ενός προφίλ μπορεί να επαναδιαμορφωθεί με την πρόσθεση σε αυτό σχετικών διανυσμάτων κειμένων και με την αφαίρεση από αυτό μη σχετικών διανυσμάτων κειμένων, όπως αυτά κρίνονται από το χρήστη. Για παράδειγμα μία φόρμουλα η οποία μπορεί να εφαρμοστεί είναι η Ide Regular [29], σύμφωνα με την οποία:

$$P^{(i+1)} = P^{(i)} + \sum_{D\_\sigma\chi\varepsilon\tau\iota\kappa\acute{o}} D - \sum_{D\_\mu\eta\_\sigma\chi\varepsilon\tau\iota\kappa\acute{o}} D$$

όπου $P^{(i)}$ είναι το διάνυσμα του προφίλ μετά την i-στή επανάληψη feedback.

Το βασικό στοιχείο της συγκεκριμένης μεθόδου είναι η ένωση των διανυσμάτων των κειμένων με τα αρχικά διανύσματα των προφίλ. Αυτή η ένωση επαναπροσδιορίζει τα βάρη των όρων του προφίλ. Πιο συγκεκριμένα, προσθέτει τα βάρη των όρων στα σχετικά κείμενα και αφαιρεί τα βάρη των όρων στα μη – σχετικά κείμενα. Αποτέλεσμα της συγκεκριμένης διαδικασίας είναι η επέκταση των όρων των προφίλ, καθώς προσθέτονται όροι οι οποίοι δεν είναι στις αρχικές τους αναπαραστάσεις, αλλά υπάρχουν στις αναπαραστάσεις των σχετικών και μη σχετικών κειμένων. Οι επεκτάσεις στηρίζονται σε θετικά και αρνητικά βάρη, γεγονός που εξαρτάται από το εάν οι όροι προέρχονται από σχετικά ή μη σχετικά κείμενα αντίστοιχα. Ωστόσο δεν προστίθενται νέοι όροι με αρνητικά βάρη. Η συμβολή των μη σχετικών κειμένων είναι η αλλαγή των βαρών των νέων όρων που προστίθενται από τα σχετικά κείμενα.

Η μεθοδολογία της ανάδρασης σχετικότητας, όπως περιγράφεται παραπάνω, φαίνεται να προσφέρει μία μεγάλη βελτίωση στην απόδοση του συστήματος. Ειδικά, λαμβάνοντας υπόψη το γεγονός ότι οι χρήστες συχνά δε μπορούν να προσδιορίσουν ακριβώς τα ενδιαφέροντά τους, η συγκεκριμένη τεχνική φαίνεται να αντιμετωπίζει το συγκεκριμένο πρόβλημα και να βοηθάει τελικά το χρήστη στη διαμόρφωση ενός αποτελεσματικού προφίλ.

Αναφορές

[1] T. W. Yan, and H. Garcia – Molina. Distributed selective dissemination of information. In Proceedings of the 3rd International Conference on Parallel and Distributed Information Systems (PDIS), pages 89 – 98, 1994.

[2] M. Franklin (Special Issue Editor). Special issue on data dissemination. Data Engineering Bulletin, 19(3): 3 – 54, 1996.

[3] M. Franklin, and S. Zdonik. Dissemination-based information systems. Data Engineering Bulletin, 19(3): 20 – 30, 1996.

[4] M. Franklin, and S. Zdonic. A Framework for Scalable Dissemination – Based Systems. In Proceedings of the ACM OOPSLA Conference, Atlanta, pages 94 – 105, October, 1997.

[5] K. Fujimoto, and H Sato. Semantic Word – matching for Knowledge Acquisition from Text Containing Daily used Words: A Multiagent – based Approach. In Proceedings of the 1st International Conference on Advances in Intelligent Systems: Theory and Application (AISTA – 2000), 2000.

[6] E. Krol. The Whole Internet User's Guide & Catalog. O'Reilly & Associates, Sebastopol, California, 1992.

[7] D. Gifford, R. Baldwin, S. Berlin, and J. Lucassen. An Architecture for Large Scale Information Systems. In Proceedings of the Symposium on Operating System Principles, pages 161 – 170, ACM, December, 1985.

[8] S. Acharya, M. Franklin, and S. Zdonik. Disseminating updates on broadcast disks. In Proceedings of Very Large Data Bases (VLDB), pages 354 – 365, 1996.

[9] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast Disks: Data

Management for Asymmetric Communication Environments. In Proceedings of the ACM SIGMOD Conference, San Jose, May, 1995.

[10] S. Acharya, M. Franklin, and S. Zdonik. Dissemination-based Data Delivery Using Broadcast Disks. In Proceedings of the IEEE Personal Communications, 2(6): 50 – 60, December, 1995.

[11] S. Acharya, M. Franklin, and S. Zdonik. Balancing Push and Pull for Data Broadcast. In Proceedings of the ACM SIGMOD Conference, Arizona, pages 183 – 194, May, 1997.

[12] D. Aksoy, and M. Franklin. Scheduling for Large – Scale On – Demand Data Broadcasting. In Proceedings of the IEEE INFO-COM Conference, San Francisco, pages 651 – 659, March, 1998.

[13] M. Altinel, and M. Franklin. Efficient Filtering of XML Documents for Selective Dissemination of Information. In Proceedings of the 26th VLDB Conference, Cairo, Egypt, pages 53 – 64, September, 2000.

[14] J. Clark, and S. DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendation, http://www.w3.org/TR/xpath, November, 1999.

[15] B. Oki, M. Pfluegl, A. Siegel, and D. Skeen. The Information Bus – an architecture for extensible distributed Systems. Operating Systems Review, 27(5): 58-68, 1993.

[16] A. Campailla, S. Chaki, E. Clarke, S. Jha, and H. Veith. Efficient Filtering in Publish-Subscribe Systems using Binary Decision Diagrams. In Proceedings of the 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, pages 443 – 452, May, 2001.

[17] F. Fabret, H. A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha. Filtering algorithms and implementation for very fast publish/ subscribe systems. In Proceedings of the ACM SIGMOD – 2001, 2001.

[18] J. Pereira, F. Fabret, F. Llibrat, and D. Shasha. Efficient matching for web – based publish/ subscribe systems. In Proceedings of the COOPIS – 2000, 2000.

[19] T. W. Yan, and H. Garcia – Molina. SIFT – A Tool for Wide – Area Information Dissemination. In Proceedings of the 1995 USENIX Technical Conference, pages 177 – 86, 1995.

[20] T. W. Yan, and H. Garcia – Molina. The SIFT information dissemination system. ACM Transactions on Database Systems, 24(4): 529 – 565, 1999.

[21] M. Franklin, and S. Zdonik. "Data in Your Face": Push Technology in Perspective. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 516 – 519, 1998.

[22] P. Deolasee, A. KatKar, A. Panchbudhe, K. Ramamritham, and P. Shenoy. Adaptive Push – Pull: Disseminating Dynamic Web Data. In Proceedings of the 16th International WWW Conference, Hong Kong, pages 265 – 274, May, 2001.

[23] C. - C. K. Chang, H. Garcia-Molina, and A. Paepcke. Predicate Rewriting for Translating Boolean Queries in a Heterogeneous Information System. ACM Transactions on Information Systems, 17(1): 1 - 39, January, 1999.

[24] C. - C. K. Chang, H. Garcia-Molina, and A. Paepcke. Boolean query mapping across heterogeneous information sources. IEEE Transactions on Knowledge and Data Engineering 8(4): 515 – 521, 1996.

[25] C. - C. K. Chang, H. Garcia-Molina, and A. Paepcke. Boolean query mapping across heterogeneous information sources (extended version). Technical Report SIDL-WP-1996-0044, Stanford University Accessible at http://www-diglib.stanford.edu, September, 1996.

[26] M. Koubarakis, C. Tryfonopoulos, P. Raftopoulou, and T. Koutris. Data Models

and Languages for Agent-Based Textual Information Dissemination. To be presented at the 6th International Workshop on Cooperative Information Agents (CIA2002), Madrid, Spain, September 18 -20, 2002.

[27] M. Koubarakis. Boolean Queries with Proximity Operators for Information Dissemination. In Proceedings of the workshop on Foundations of Models and Languages for Information Integration (FMII-2001), Viterbo, Italy, 16 – 18 September, 2001.

[28] K. Goda, T. Tamura, M. Kitsuregawa, A. Chowdhury, and O. Frieder. Query Optimization for Vector Space Problems. In Proceedings of the 24th annual international ACM SIGIR Conference on R&D in Information Retrieval, New Orleans, Louisiana, USA, September, 2001.

[29] M. W. Berry, Z. Drmac, and E. R. Jessup. Matrices, Vector Spaces, and Information Retrieval. Society for Industrial and Applied Mathematics (SIAM REVIEW), Vol. 41, No. 2, pages 335–362, November, 1999.

[30] C. L. Viles, and J. C. French. Dissemination of Collection Wide Information in a Distributed Information Retrieval System. In Proceedings of the 18th Annual International ACM – SIGIR Conference on R&D in Information Retrieval, Seattle, pages 12 – 20, July, 1995.

[31] J. Callan. Document Filtering with Inference Networks. In Proceedings of the 19th International ACM SIGIR Conference on R&D in Information Retrieval (SIGIR – 1996) Zurich, Switzerland, pages 262 – 269, 1996.

[32] U. Cetintemel, M. J. Franklin, and C. L. Giles. Self-Adaptive User Profiles for Large-Scale Data Delivery. In Proceedings of the 16th International Conference on Data Engineering (ICDE), San Diego, CL, USA, pages 622 – 633, February, 2000.

[33] R M. Losee. Comparing Boolean and Probabilistic Information Retrieval Systems Across Queries and Disciplines. Journal of the American Society for

Information Science 48(2): 143–156, 1997.

[34] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman. On Supporting Containment Queries in Relational Database Management Systems. In Proceedings of the ACM SIGMOD Conference on Management of Data, Santa Barbara, California, USA, pages 425 – 436, May, 2001.

[35] T. W. Yan and H. Garcia – Molina. Index structures for selective dissemination of information under the Boolean model. ACM Transactions on Database Systems, 19(2): 332 – 364, 1994.

[36] T. W. Yan and H. Garcia – Molina. Index Structures for Selective Dissemination of Information. Technical Report, STAN-CS-92-1454, Stanford University, 1992.

[37] T. W. Yan and H. Garcia – Molina. Index Structures for Information Filtering Under the Vector Space Model. In Proc. International Conference on Data Engineering, pages 337 – 47, 1994.

[38] D. W. Oard, and G. Marchionini. A Conceptual Framework for Text Filtering. Technical Report, CS-TR-3643, University of Maryland, May, 1996.

[39] WordNet Home Page (http://www.cogsci.princeton.edu/~wn/)

[40] Java Home Page (http://java.sun.com)

[41] The Java Developers Kit Home Page (http://java.sun.com/products/jdk/1.1/index.html)

[42] Visual Studio Home Page (http://msdn.microsoft.com/vstudio/)

[43] Java.applet Package (http://java.sun.com/products/jdk/1.1/docs/api/Package-java.applet.html)

[44] Java.net Package (http://java.sun.com/products/jdk/1.1/docs/api/Package-java.net.html)

[45] Oracle home page (http://www.oracle.com)

[46] JDBC Technology (http://java.sun.com/products/jdbc/index.html)

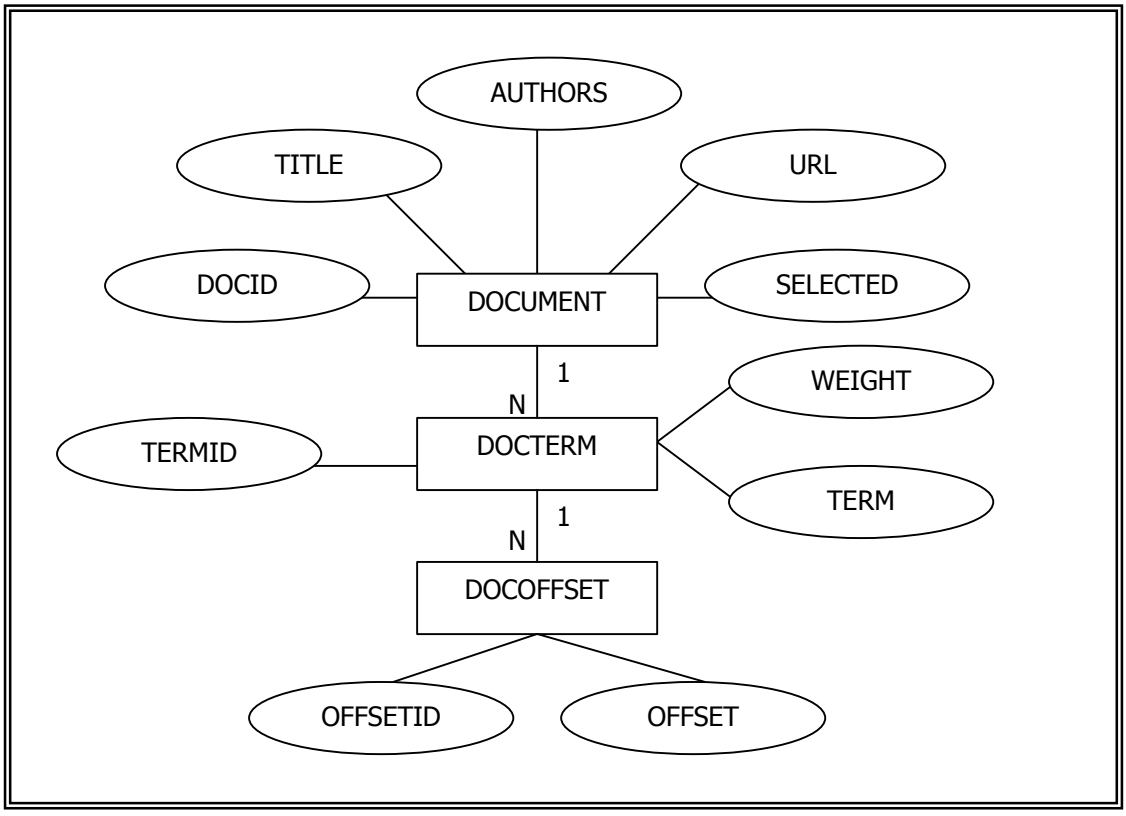
[47] D. Harman. Relevance Feedback Revisited. In Proceedings of the 15th International ACM SIGIR Conference on R&D in Information Retrieval (SIGIR – 1992), Denmark, pages 1 – 10, June, 1992.

Παράρτημα Α.

Περιγραφή βάσης δεδομένων

Α.1 Πίνακες κειμένων

Το διάγραμμα οντοτήτων συσχετίσεων (E – R Diagram) του σχήματος της βάσης δεδομένων για τα κείμενα του συστήματος φαίνεται στο Σχήμα 41. Από αυτό το διάγραμμα προκύπτουν τρεις πίνακες, οι οποίοι διατηρούν τα στοιχεία των κειμένων.



Σχήμα 41. Διάγραμμα E-R του σχήματος βάσης δεδομένων των κειμένων

Ο πίνακας DOCUMENT, του οποίου τα πεδία φαίνονται στο Σχήμα 42, χρησιμοποιείται για την αποθήκευση των βασικών στοιχείων των κειμένων και έχει σαν πρωτεύον κλειδί το πεδίο DOCID.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	DOCID	Number	Προσδιοριστικό (Id) του κειμένου
2	TITLE	Varchar2(200)	Τίτλος του κειμένου
3	AUTHORS	Varchar2(200)	Συγγραφέας του κειμένου
4	URL	Varchar2(200)	URL του κειμένου
5	SELECTED	Number	Βοηθητική μεταβλητή που φανερώνει εάν το κείμενο έχει φορτωθεί στη βάση (1) ή όχι (0).

Σχήμα 42. Πεδία πίνακα DOCUMENT

Ο πίνακας DOCTERM, του οποίου τα πεδία φαίνονται στο Σχήμα 43, χρησιμοποιείται για την αποθήκευση των όρων των κειμένων. Έχει σαν πρωτεύον κλειδί το συνδυασμό πεδίων DOCID, TERMID και ως ξένο κλειδί το πεδίο DOCID, το οποίο αναφέρεται στο πεδίο DOCUMENT.DOCID.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	TERMID	Number	Προσδιοριστικό (Id) του όρου
2	TERM	Varchar2(100)	Όνομα του όρου
3	WEIGHT	Long	Βάρος του όρου
4	DOCID	Number	Προσδιοριστικό του κειμένου, στο οποίο ανήκει ο όρος.

Σχήμα 43. Πεδία πίνακα DOCTERM

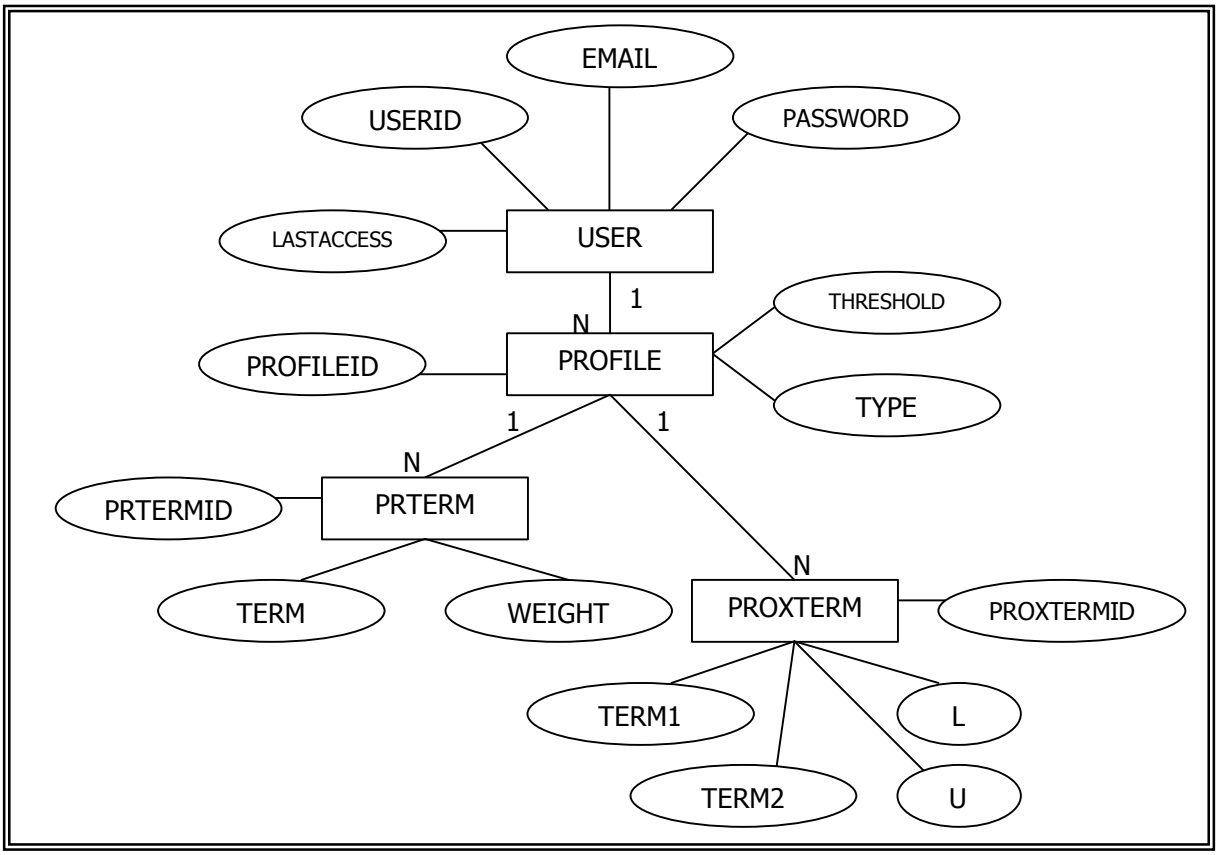
Τέλος καθώς οι όροι ενός κειμένου μπορεί να επαναλαμβάνονται στο κείμενο, ένας όρος μπορεί να έχει πολλά offsets. Οπότε διατηρούμε ένα πίνακα για τα offsets των όρων, τον DOCOFFSET, του οποίου τα πεδία περιγράφονται στο Σχήμα 44. Πρωτεύον κλειδί του πίνακα είναι ο συνδυασμός των πεδίων OFFSETID, TERMID, DOCID και ξένο κλειδί είναι το TERMID, DOCID το οποίο αναφέρεται στο πρωτεύον κλειδί του πίνακα DOCTERM.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	OFFSETID	Number	Προσδιοριστικό του offset
2	OFFSET	Number	Τιμή του offset
3	TERMID	Number	Προσδιοριστικό του όρου, στο οποίο ανήκει το offset.
4	DOCID	Number	Προσδιοριστικό του κειμένου, στο οποίο ανήκει ο όρος που ανήκει το offset.

Σχήμα 44. Πεδία πίνακα DOCOFFSET

Α.2 Πίνακες χρηστών και προφίλ

Το διάγραμμα οντοτήτων συσχετίσεων (E – R Diagram) του σχήματος της βάσης δεδομένων για τους χρήστες και τα προφίλ τους φαίνεται στο Σχήμα 45. Από αυτό το διάγραμμα προκύπτουν τέσσερις πίνακες, στους οποίους αποθηκεύονται όλα τα στοιχεία των χρηστών και των προφίλ του συστήματος.



Σχήμα 45. Διάγραμμα E-R του σχήματος βάσης δεδομένων των προφίλ

Αρχικά υπάρχει ο πίνακας των χρηστών USERS, ο οποίος αποθηκεύει τα απαραίτητα στοιχεία ενός χρήστη. Αυτός ο πίνακας φαίνεται στο Σχήμα 46. Πρωτεύον κλειδί του είναι το πεδίο USERID.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	USERID	Number	Προσδιοριστικό (Id) του χρήστη
2	EMAIL	Varchar2(50)	Ηλεκτρονική διεύθυνση (email) του χρήστη
3	PASSWORD	Varchar2(10)	Κωδικός πρόσβασης του χρήστη

Σχήμα 46. Πεδία πίνακα USERS

Στη βάση αποθηκεύονται τα υπο – προφίλ κάθε χρήστη στον πίνακα PROFILE, του οποίου τα πεδία φαίνονται στο Σχήμα 47. Ο πίνακας έχει σαν πρωτεύον κλειδί το συνδυασμό των πεδίων PROFILEID, USERID. Ως ξένο κλειδί παρουσιάζεται το πεδίο USERID το οποίο αναφέρεται στο πεδίο USER.USERID.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	PROFILEID	Number	Προσδιοριστικό (id) του υπο – προφίλ
2	THRESHOLD	Long	Κατώφλι σχετικότητας του υπο - προφίλ
3	TYPE	Number	Τύπος του υπο – προφίλ (ισούται με 1 εάν το υπο – προφίλ ανήκει στο μοντέλο Boolean και με 2 εάν το υπο – προφίλ ανήκει στο μοντέλο Vector Space)
4	USERID	Number	Προσδιοριστικό του χρήστη στον οποίο ανήκει το υπο – προφίλ

Σχήμα 47. Πεδία πίνακα PROFILE

Οι όροι του κάθε υπο - προφίλ αποθηκεύονται στον πίνακα PRTERM, ο οποίος φαίνεται στο Σχήμα 48. Πρωτεύον κλειδί του πίνακα είναι το σύνθετο πεδίο PRTERMID, PROFILEID, USERID, ενώ ξένο κλειδί είναι το σύνθετο πεδίο PROFILEID, USERID, το οποίο αναφέρεται στο πρωτεύον κλειδί του πίνακα PROFILE.

Τέλος, οι συνθήκες εγγύτητας του κάθε υπο - προφίλ αποθηκεύονται στον πίνακα PROXTERM, ο οποίος φαίνεται στο Σχήμα 49. Πρωτεύον κλειδί του πίνακα

είναι το σύνθετο πεδίο PROXID, PROFILEID, USERID, ενώ ξένο κλειδί είναι το σύνθετο πεδίο PROFILEID, USERID, το οποίο αναφέρεται στο πρωτεύον κλειδί του πίνακα PROFILE.

Α/Α	Όνομα	Τύπος	Περιγραφή
1	PRTERMID	Number	Προσδιοριστικό (id) του όρου
2	TERM	Varchar2(100)	Όρος
3	WEIGHT	Long	Βάρος του όρου
4	PROFILEID	Number	Προσδιοριστικό του υπο - προφίλ στο οποίο ανήκει ο όρος
5	USERID	Number	Προσδιοριστικό του χρήστη στον οποίο ανήκει το υπο – προφίλ που ανήκει ο όρος

Σχήμα 48. Πεδία πίνακα PRTERM

Α/Α	Όνομα	Τύπος	Περιγραφή
1	PROXID	Number	Προσδιοριστικό της συνθήκης εγγύτητας
2	TERM1	Varchar2(100)	Πρώτος όρος της συνθήκης
3	TERM2	Varchar2(100)	Δεύτερος όρος της συνθήκης
4	L	Number	Ελάχιστο πλήθος λέξεων που πρέπει να μεσολαβεί ανάμεσα στους δύο όρους
5	U	Number	Μέγιστο πλήθος λέξεων που πρέπει να μεσολαβεί ανάμεσα στους δύο όρους
6	PROFILEID	Number	Προσδιοριστικό του υπο - προφίλ στο οποίο ανήκει ο όρος
7	USERID	Number	Προσδιοριστικό του χρήστη στον οποίο ανήκει το υπο – προφίλ που ανήκει ο όρος

Σχήμα 49. Πεδία πίνακα PROXTERM

Παράρτημα Β.

Περιγραφή δομών εφαρμογής

Η ιδέα του οντοκεντρικού προγραμματισμού, η οποία ακολουθήθηκε κατά την ανάπτυξη και υλοποίηση του συστήματος, απαιτεί τη δημιουργία αντικειμένων με ιδιότητες και μεθόδους. Τα αντικείμενα αυτά χρησιμοποιήθηκαν για την υλοποίηση των διαφόρων δομών και περιγράφονται συνοπτικά στη συνέχεια.

Β.1 Δομές χειριστή χρηστών

Όπως προαναφέρθηκε, ο χειριστής των χρηστών διατηρεί μία λίστα με τους χρήστες και τα προφίλ του συστήματος. Για την υλοποίηση της συγκεκριμένης δομής υλοποιήθηκαν τέσσερα αντικείμενα, όπως φαίνεται και στο Σχήμα 50.

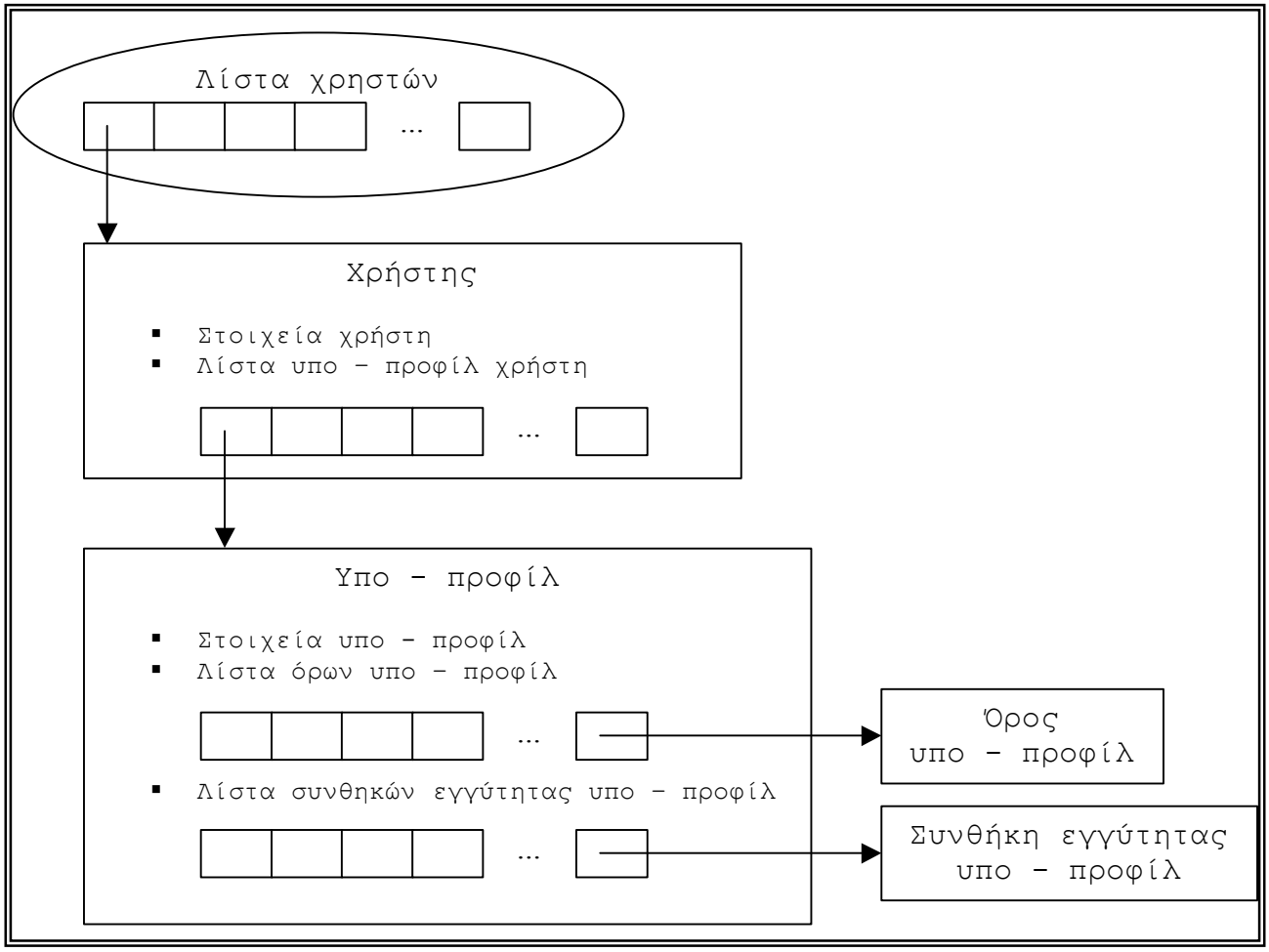
Ως πρώτο αντικείμενο εμφανίζεται η λίστα των χρηστών, το οποίο αποτελεί τη βασική δομή από την οποία μπορούμε να έχουμε πρόσβαση στους χρήστες και τα προφίλ τους.

Κάθε στοιχείο της συγκεκριμένης λίστας είναι ένα αντικείμενο τύπου «χρήστης». Τα στοιχεία του χρήστη που επιλέχθηκαν να αποθηκεύονται είναι πέρα από το προσδιοριστικό του (id), το email και ο κωδικός πρόσβασής του, καθώς και το πλήθος των υπο – προφίλ τα οποία αυτός έχει αποστείλει στο σύστημα. Κάθε χρήστης διατηρεί μία λίστα με τα υπο – προφίλ του.

Κάθε στοιχείο της λίστας των υπο – προφίλ του χρήστη είναι ένα αντικείμενο «υπο – προφίλ». Οι βασικές ιδιότητες του συγκεκριμένου αντικειμένου είναι το προσδιοριστικό του (id), ο τύπος του, το κατώφλι σχετικότητάς του, το σκορ ως προς τα διάφορα κείμενα και το πλήθος των όρων του, καθώς και των συνθηκών εγγύτητάς του. Όπως παρουσιάζεται και στο σχήμα, το υπο – προφίλ διατηρεί δύο λίστες, μία για τους όρους του και μία για τις συνθήκες εγγύτητας που περιλαμβάνει.

Μέλη της λίστας των όρων του υπο – προφίλ αποτελούν αντικείμενα τύπου «όρος υπο - προφίλ». Αυτά τα αντικείμενα δεν περιέχουν τίποτα περισσότερο από τον όρο και το βάρος του. Από την άλλη πλευρά μέλη της λίστας των συνθηκών εγγύτητας αποτελούν αντικείμενα τύπου «συνθήκη εγγύτητας». Τα συγκεκριμένα αντικείμενα περιέχουν τον πρώτο και το δεύτερο όρο που λαμβάνει μέρος στη

συνθήκη, καθώς και το μέγιστο και ελάχιστο πλήθος όρων που υπάρχουν ανάμεσα στους δύο όρους.



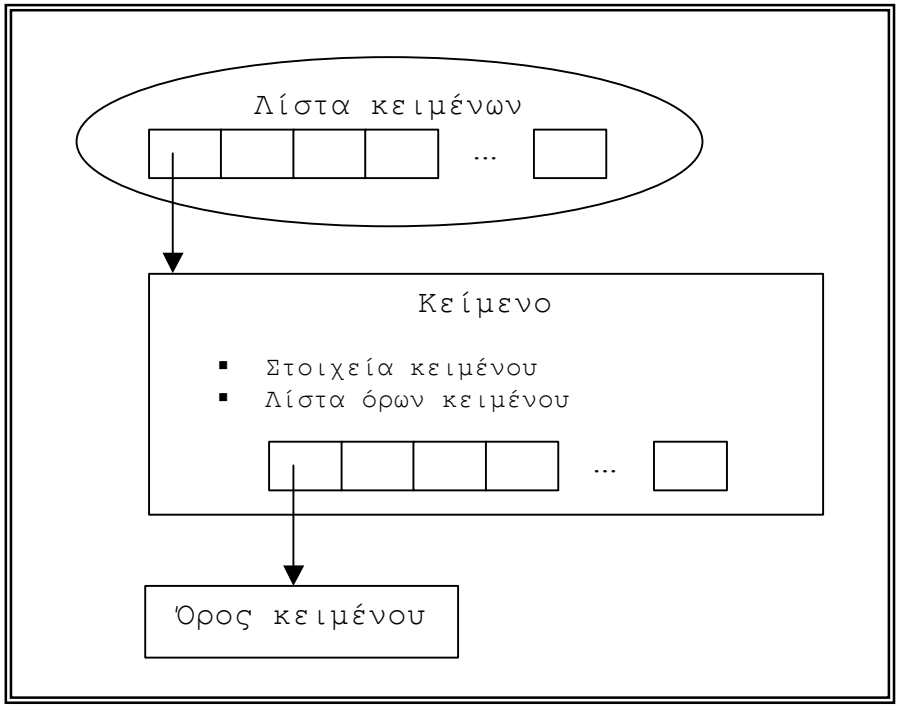
Σχήμα 50. Δομή για τους χρήστες και τα προφίλ τους

Επιπλέον, ο χειριστής χρηστών διατηρεί και μία δομή, που καλείται «λίστα stop – words» και περιέχει όλα τα stop –words. Αυτή η δομή χρησιμοποιείται για την αφαίρεση των συγκεκριμένων λέξεων από τα προφίλ των χρηστών. Είναι μία απλή λίστα με stop – words, δηλαδή με λέξεις οι οποίες θεωρούνται πολύ συνηθισμένες και δεν προσθέτουν κάποιο προσδιοριστικό στοιχεία για ένα προφίλ ή κείμενο. Η εισαγωγή στοιχείων στη δομή πραγματοποιείται κατά την εκκίνηση του συστήματος, όταν διαβάζεται το αρχείο "stopwords.txt" που περιέχει αυτές τις λέξεις.

Β.2 Δομές χειριστή κειμένων

Η λίστα των κειμένων που διατηρείται από το χειριστή των κειμένων παρουσιάζεται στο Σχήμα 51 και περιλαμβάνει τρία αντικείμενα.

Βασικό αντικείμενο είναι η λίστα των κειμένων, από την οποία το σύστημα μπορεί να προσπελάσει τα κείμενα. Κάθε στοιχείο της λίστας αποτελεί ένα αντικείμενο τύπου «κείμενο». Κυριότερες ιδιότητες του συγκεκριμένου αντικειμένου είναι το προσδιοριστικό του (id), ο τίτλος, οι συγγραφείς και το URL του κειμένου, καθώς και το πλήθος των όρων που αυτό έχει.

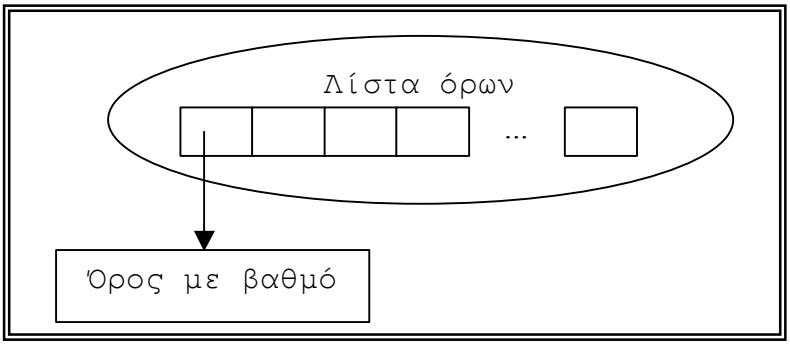


Σχήμα 51. Δομή για τα κείμενα

Όπως φαίνεται και στο σχήμα, κάθε κείμενο διατηρεί μία λίστα με τους όρους του, δηλαδή μία λίστα με αντικείμενα τύπου «όρος κειμένου». Αυτά τα αντικείμενα περιλαμβάνουν τον όρο, το βάρος του, καθώς και μία λίστα με τα offsets που έχει ο όρος μέσα στο κείμενο. Αξίζει να σημειώσουμε ότι επιλέχθηκε η δημιουργία λίστας για την αποθήκευση της τιμής των offsets, καθώς ένας όρος μπορεί να εμφανίζεται πολλές φορές σε ένα κείμενο, οπότε και να έχει περισσότερα του ενός offsets.

Παράλληλα ο χειριστής κειμένων διατηρεί μία δομή για τους όρους με τους βαθμούς τους, η οποία παρουσιάζεται στο Σχήμα 52. Δηλαδή υπάρχει ένα αντικείμενο

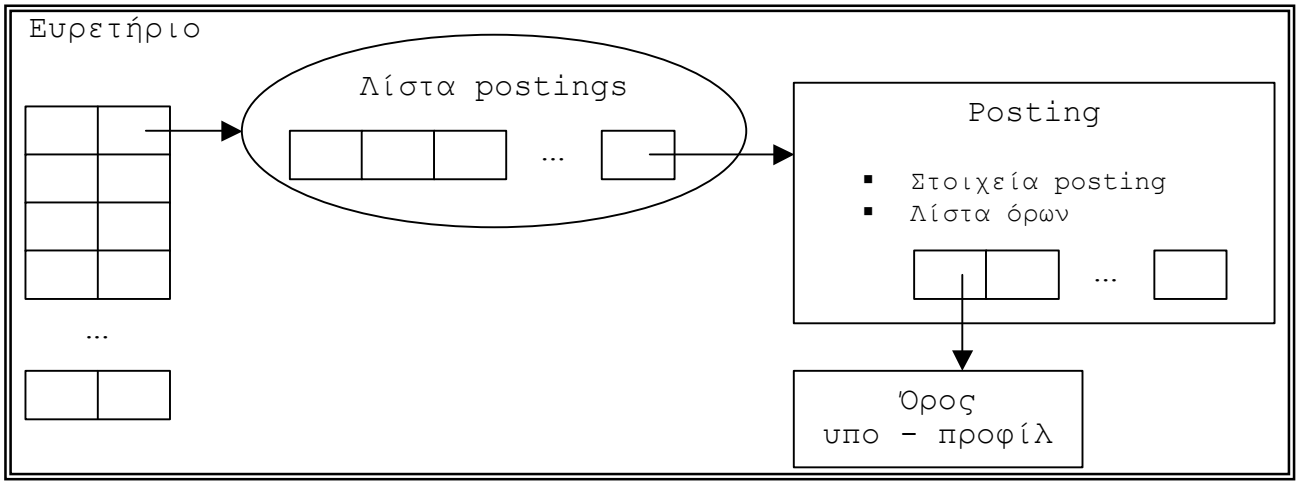
«λίστα όρων», το οποίο είναι μία λίστα με στοιχεία αντικείμενα τύπου «όρος με βαθμό». Αυτό το τελευταίο αντικείμενο έχει ως ιδιότητες τον όρο και το βαθμό του στο σύστημα.



Σχήμα 52. Δομή για τους όρους με τους βαθμούς τους

Β.3 Δομές μηχανής φιλτραρίσματος

Η μηχανή φιλτραρίσματος διατηρεί τη δομή του ευρετηρίου των προφίλ που φαίνεται στο Σχήμα 53 και περιλαμβάνει τέσσερα αντικείμενα.

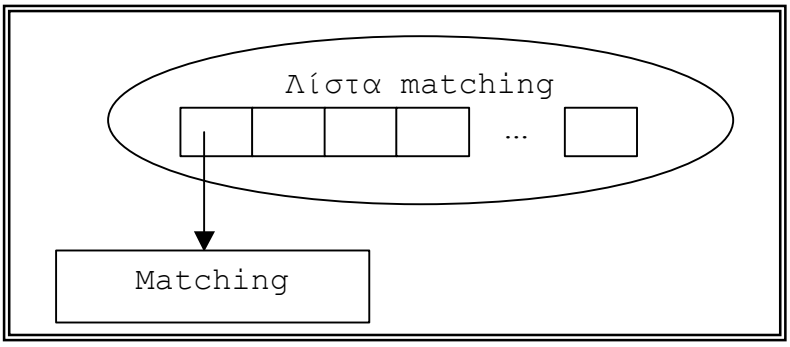


Σχήμα 53. Δομή ευρετηρίου των προφίλ

Αρχικά υπάρχει το αντικείμενο «ευρετήριο», το οποίο έχει τη μορφή hash table. Ως κλειδιά του hash table εμφανίζονται όροι και ως τιμές αντικείμενα τύπου «λίστα postings». Η κάθε λίστα posting έχει σαν στοιχεία αντικείμενα τύπου «posting». Κάθε τέτοιο αντικείμενο διατηρεί το προσδιοριστικό του υπο – προφίλ στο

οποίο αντιστοιχεί, το βάρος του όρου που βρίσκεται στο hash table της δομής και με τον οποίο αυτό το posting συνδέεται και το σύνολο των όρων που περιέχει. Τέλος, όπως φαίνεται και στο σχήμα περιέχει μία λίστα από όρους, δηλαδή από αντικείμενα τύπου «όρος υπο – προφίλ».

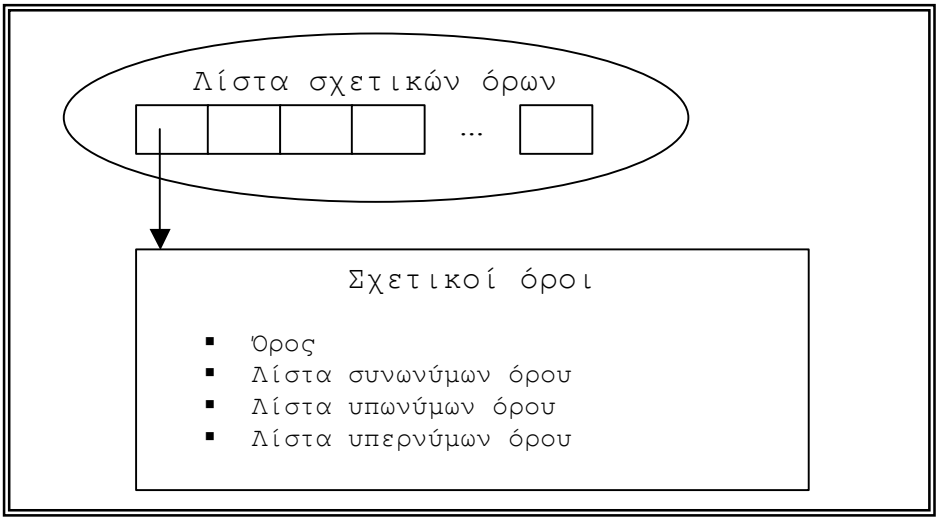
Μία βοηθητική δομή που διατηρείται στη μηχανή αναζήτησης είναι η «λίστα matching», η οποία παρουσιάζεται και στο Σχήμα 54 και διατηρεί τα ζεύγη κειμένων – υπο – προφίλ που ταιριάζουν μεταξύ τους. Αυτή η λίστα έχει σαν στοιχεία αντικείμενα τύπου «matching», τα οποία δεν περιλαμβάνουν τίποτα περισσότερο από το προσδιοριστικό του κειμένου, το προσδιοριστικό του υπο – προφίλ και μία μεταβλητή που δηλώνει εάν ο αντίστοιχος χρήστης έχει ενημερωθεί ότι αυτό το κείμενο ταιριάζει στο προφίλ του ή όχι.



Σχήμα 54. Δομή matching

Β.4 Δομές σημασιολογικού χειριστή

Ο σημασιολογικός χειριστής διατηρεί τη δομή των σχετικών όρων που φαίνεται στο Σχήμα 55. Η συγκεκριμένη δομή αποτελεί ένα αντικείμενο τύπου «λίστα σχετικών όρων», το οποία είναι μία λίστα με στοιχεία αντικείμενα τύπου «σχετικοί όροι». Όπως φαίνεται και στο σχήμα, κάθε τέτοιο αντικείμενο περιέχει τον όρο, στον οποίο αναφέρονται οι σχετικοί όροι, καθώς και τα συνώνυμα, τα υπώνυμα και τα υπερώνυμά του.



Σχήμα 55. Δομή σχετικών όρων

Β.5 Άλλα αντικείμενα του συστήματος

Πέρα από τα προαναφερθέντα αντικείμενα, που υλοποιούν τις βασικές δομές του συστήματος, υπάρχουν και κάποια άλλα, τα οποία αναλαμβάνουν ειδικές λειτουργίες στο σύστημα.

Αρχικά υπάρχει ένα αντικείμενο το οποίο διατηρεί το πρωτόκολλο επικοινωνίας ανάμεσα στην εφαρμογή και στον κάθε πελάτη. Αυτό το αντικείμενο χρησιμοποιείται από το χειριστή των χρηστών και περιγράφει ακριβώς το «διάλογο» ανάμεσα στον client και το server.

Όπως προαναφέρθηκε, η εφαρμογή γεννά νήματα τόσο για να εξυπηρετήσει του πελάτες όσο και για να διαχειριστεί τα κείμενα. Δύο αντικείμενα έχουν δημιουργηθεί, τα οποία υλοποιούν αυτά τα νήματα, ένα για τα νήματα των χρηστών και ένα για τα κείμενα. Αυτά τα αντικείμενα χρησιμοποιούνται από τον χειριστή των χρηστών και των κειμένων αντίστοιχα. Παράλληλα, οι δύο αυτοί χειριστές χρησιμοποιούν ένα αντικείμενο το οποίο υλοποιεί την επικοινωνία με τη βάση δεδομένων των χρηστών και των κειμένων.