ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΤΑ ΠΛΑΙΣΙΑ ΤΟΥ
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ (ΜΔΕ)
«ΕΠΙΣΤΗΜΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ»
ΤΟΥ ΤΜΗΜΑΤΟΣ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗ ΔΙΚΤΥΑΚΗ ΠΥΛΗ ΠΡΟΒΟΛΗΣ ΠΡΟΕΠΕΞΕΡΓΑΣΜΕΝΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ

Σιλιντζήρης Παναγιώτης
Α.Μ. 422

Επιβλέπων Καθηγητής
Χρήστος Μπούρας, Καθηγητής

Τριμελής Επιτροπή
Ιωάννης Γαροφαλάκης, Αναπληρωτής Καθηγητής
Χρήστος Μπούρας, Καθηγητής
Δημήτριος Χριστοδουλάκης, Καθηγητής

Πάτρα, Ιούνιος 2010

Πρόλογος

Η παρούσα διπλωματική εργασία αποτελεί τον επίλογο των μεταπτυχιακών μου σπουδών στα πλαίσια του Μεταπτυχιακού Διπλώματος Ειδίκευσης «Επιστήμη και Τεχνολογία Υπολογιστών» του Τμήματος Μηχανικών Η/Υ και Πληροφορικής. Το περιεχόμενο της εργασίας σχετίζεται την προσωποποιημένη στο χρήστη αναζήτηση άρθρων σε διαδικτυακή πύλη.

Πριν την παρουσίαση της εργασίας αυτής, αισθάνομαι την ανάγκη να ευχαριστήσω θερμά όλους όσους με καθοδήγησαν, με βοήθησαν και μου συμπαραστάθηκαν κατά τη διάρκεια της εκπόνησής της. Σε αυτό το σημείο θέλω να ευχαριστήσω τον Καθηγητή Χρήστο Μπούρα για την καθοδήγησή του αλλά και διότι μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Δε θα μπορούσα βέβαια να μην ευχαριστήσω θερμά τον Πουλόπουλο Βασίλη, υποψήφιο διδάκτορα, ο οποίος με βοηθούσε και με συμβούλευε σε όλη τη διάρκεια εκπόνησης αυτής της διπλωματικής εργασίας. Τέλος, ευχαριστώ τους ερευνητές της Ερευνητικής Μονάδας 6 του ΕΑΙΤΥ για τη βοήθεια που μου προσέφεραν.

Ομοίως, θέλω να ευχαριστήσω τον Αναπληρωτή Καθηγητή Ιωάννη Γαροφαλάκη και τον Καθηγητή Δημήτριο Χριστοδουλάκη, για την τιμή που μου έκαναν να αποτελέσουν μέλη της Τριμελούς Εξεταστικής Επιτροπής.

Τέλος, θα ήθελα να ευχαριστήσω όλους όσους με στήριξαν στην προσπάθεια που κατέβαλλα τον τελευταίο χρόνο των μεταπτυχιακών σπουδών μου.

Πάτρα, 10 Ιουνίου 2010

Σιλιντζήρης Β. Παναγιώτης

Επιτελική Σύνοψη

Σκοπός της παρούσας Μεταπτυχιακής Εργασίας είναι η μελέτη και η αξιολόγηση των δυνατοτήτων προηγμένης σημασιολογικής αναζήτησης (advanced semantic search) πάνω σε περιεχόμενο που προέρχεται από το Διαδικτύο. Στα πλαίσια της εργασίας, σχεδιασθηκε και υλοποιήθηκε υποσύστημα, το οποίο ενσωματώθηκε και αξιολογήθηκε πάνω στο μηχανισμό peRSSonal ο οποίος ανακτά, επεξεργάζεται και παρουσιάζει στους χρήστες του άρθρα και υλικό από διάφορες ειδησεογραφικές πύλες (news portals) του Διαδικτύου, προσαρμόζοντάς τα στις προσωπικές επιλογές του χρήστη. Η αναζήτηση παραμετροποιείται με στοιχεία όπως: χρονικό πλαίσιο δημοσίευσης τού υλικού (διάστημα από – έως), κατηγορία (πρότυπες κατηγορίες του συστήματος peRSSonal), φυσική γλώσσα στην οποία είναι γραμμένο καθώς και δυνατότητα για στατικό ή σημασιολογικό (εννοιολογικό) ταίριασμα (semantic matching) με τα άρθρα της βάσης.

Αρχικά, από την επερώτηση (query) του χρήστη δημιουργείται ένα σύνολο ριζών (stems) των λέξεων οι οποίες δόθηκαν. Η εξαγωγή των ριζών εκτελείται με υποβοήθηση από stemming αλγορίθμο για την αγγλική γλώσσα, ενώ ο σχεδιασμός του συστήματος προβλέπει και τη μελλοντική υποστήριξη διαφορετικών φυσικών γλωσσών καταβάλλοντας μικρό κόπο. Για τις λεκτικές ρίζες που προκύπτουν, εντοπίζονται σχετικές τους και ταυτόχρονα με τη διαδικασία αυτή διενεργείται αναζήτηση στη βάση δεδομένων για κωδικολέξεις (keywords) με βάση την κατηγορία του άρθρου, ούτως ώστε να εμπλουτιστεί το ερώτημα του χρήστη με επιπλέον πληροφορία, καθιστώντας πιο επιτυχημένη και στοχευμένη την αναζήτηση στην πληθώρα των άρθρων που υπάρχουν αποθηκευμένα στη βάση δεδομένων. Για αυτές τις κωδικολέξεις υπολογίζονται συντελεστές – βάρη που θα προσδιορίζουν τη συνάφειά τους με την επερώτηση του χρήστη.

Ανάλογα με τον τύπο της αναζήτησης, στατική η σημασιολογική, το υποσύστημα αναζήτησης συγκρίνει την επερώτηση του χρήστη με τα αποθηκευμένα άρθρα και για κάθε ένα από αυτά, ο αλγόριθμος υπολογίζει το βαθμό συνάφειάς του με την επερώτηση. Τα άρθρα που επιλέγονται τελικά είναι αυτά που ξεπερνούν ένα κατώφλι συνάφειας, το οποίο τα κατατάσσει εννοιολογικά πιο κοντά στην επερώτηση του χρήστη. Σημαντικό σημείο στο στάδιο αυτό, είναι η δυνατότητα, για τους εγγεγραμένους χρήστες της Δικτυακής Πύλης, να εκτελείται περεταίρω φιλτράρισμα πάνω στο πρωτογενές αποτέλεσμα, βάσει των προσωπικών τους επιλογών καθώς και πληροφορίας που προέρχεται από τη βάση δεδομένων και που διαμορφώνεται δυναμικά από την παρατήρηση της γενικής συμπεριφοράς των χρηστών κατά την πλοήγηση τους μέσα στον σύστημα (χρόνος παραμονής στα άρθρα, άρθρα που δεν προτιμώνται, συχνότητα επιλογής άρθρων από μια δεδομένη θεματική ενότητα κλπ). Σκοπός είναι η εξαγωγή πιο στοχευμένου συνόλου άρθρων που ικανοποιεί τελικά περισσότερο τον χρήστη.

Τέλος, για την βελτίωση της απόδοσης του συστήματος, σχεδιάστηκε και υλοποιήθηκε αλγόριθμος που εκτελεί caching στα αποτελέσματα των επερωτήσεων. Με τον τρόπο αυτό, κάθε νέα αναζήτηση θα λαμβάνει πολύ πιο γρήγορα τα cached αποτελέσματα προγενέστερων παρόμοιων αναζητήσεων, ξοδεύοντας το χρόνο στα πιο πρόσφατα άρθρα. Το caching εκτελείται δυναμικά, τροποποιώντας σε κάθε επερώτηση που υποβάλλεται τα αντίστοιχα cached αποτελέσματα και μεταβάλλοντας τις προτεραιότητές τους και τα βάρη τους, ώστε να οδηγεί την έξοδο ολοένα και πιο κοντά στα επιθυμητά άρθρα και παραμένοντας πιο κοντά στο εξελισόμενο προφίλ και στις προτιμήσεις του χρήστη.

Μέσα από την εργασία, προέκυψαν αποτελέσματα που έχουν να κάνουν με σύγκριση αλγορίθμων σε όλα τα παραπάνω στάδια του μηχανισμού αλλά και ανταπόκριση του μηχανισμού στις ανάγκες του χρήστη.

Executive Summary

The scope of the present MSc Thesis is the study and the evaluation of the features provided by an advanced semantic search over digital content which comes from the Internet. For the purposes of our work, we designed and implemented a module (subsystem), which was embedded and evaluated on the PeRSSonal news portal. The PeRSSonal news portal retrieves, processes and presents to the end user articles and other content from major News Portals of the Internet by adapting on the user's personal preferences and profile. For the search procedure, parameters such as the date interval, the thematic category and the article's language are used. Furthermore it is possible to use static or dynamic (semantic) matching with the articles of the database.

In the first phase of the procedure, from the query that the user submitted we create a set of keywords, which are the stemmed words of the words described in the initial query. The extraction stemmed words is executed by an algorithm which implements the Porter Stemmer technique. The system currently supports the English language in the search procedure but its modular architecture allows for the support of other languages as well with little effort. For the keywords produced with this procedure, we locate their synonyms and in the same time a search in the database is conducted in order to find other keywords based on the thematic category of the submitted query. This second set of keywords enriches the first set thus making the search more focused on the thematic category the user chose. For these keywords that enrich the initial query, weights are computed based on their relation with the keywords of the initial query.

Based on the type of the search (static or semantic), the search subsystem compares the enriched set of keywords with the articles stored in the database and for each one of these articles which match to the keywords of the query, a degree of relevance is computed. The articles that are selected to be in the final result are the ones that surpass a specific threshold of relevance which semantically brings them close to the user query. A significant point during this phase of the procedure is the possibility to execute for the registered users of the PeRSSonal portal a more detailed filtering on the primal result based on their personal preferences and data that is produced dynamically by observing their behavior (time they spend on the articles, not preferred articles, frequency of selecting a specific thematic category) in the system, during the sessions in that. The goal is the creation of a more focused result on the end user which satisfies him more.

In the final phase of the algorithm, and in order to optimize the algorithm's performance, we design and implement an algorithm which uses cache memory in the form of a database table and runs on the server machine. For each query that is submitted to the system, we store the retrieved results in this table and in the future queries, prior to triggering the search procedure, we compare the queries with the cached ones.In this way, every new search that already has a match in the cache table will consume much less time to execute as it will search only for articles which are not found in the cache. The caching algorithm is executed dynamically by modifying for every submitted query the cached results and by changing their priorities and their relevance weights in order to include in the output the desired articles and to stay closer to the user's profile and preferences.

From the experimental results of this work we had the chance to draw useful conclusions by the comparison of different algorithmic approaches for all the stages of the mechanism and by the response and performance of the algorithm as faced by the end user.

Περιεχόμενα

Κατάλογος εικόνων

Κατάλογος Αλγορίθμων

Κατάλογος Πινάκων

Γλωσσάρι

Association Pattern	Πρότυπο Συσχέτισης
Boolean	Δυαδική Λογική
Browser	Φυλλομετρητής Ιστού
Categorization	Κατηγοριοποίηση
Classification	Ταξινόμηση
Content	Περιεχόμενο
Corpus	Συλλογή κειμένων με συγκεκριμένες ιδιότητες
Crawler, Bot, Spider	Μηχανισμοί που πραγματοποιούν αυτόματη περιήγηση στις σελίδες του Διαδικτύου
Data Mining	Εξόρυξη Δεδομένων
Decision Tree	Δένδρο Απόφασης
E-Mail	Ηλεκτρονικό Ταχυδρομείο
Embedded Software	Ενσωματωμένο Λογισμικό
Flexible	Ευέλικτος
Format	Μορφοποίηση
Front-End	Περιβάλλον αλληλεπίδρασης χρήστη
Fuzzy	Ασαφές
Generic	Γενικού Περιεχομένου
HTML	Η βασική γλώσσα δομής του διαδικτύου (HyperText Markup Language)
Information Filtering	Φιλτράρισμα πληροφορίας
Information Retrieval	Ανάκτηση Πληροφορίας
Internet	Διαδίκτυο
Keywords	Λέξεις κλειδιά
Knowledge Mining	Εξόρυξη Γνώσης
Link	Σύνδεσμος (αναφέρεται σε ιστοσελίδα)
Machine Understandable	Κατανοητός από μηχανή
Metadata	Μεταδεδομένα
Module	Τμήμα, Κομμάτι
NLP	Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing)
News Portal	Σελίδες ειδησεογραφικού περιεχομένου
Ontology	Οντολογία, αντικείμενο
Portable	Φορητός
Portal	Δικτυακή Πύλη ενημερωτικού περιεχομένου
Preprocessing	Προεπεξεργασία
Punctuation	Στίξη
RSS / RSS Feed	Πρωτόκολλο που καθορίζει κανάλι επικοινωνίας με τη βοήθεια της γλώσσας XML
Search Engine	Μηχανή αναζήτησης
Semantic Web	Σημασιολογικός Ιστός
State of the Art	Οι τρέχουσες εξελίξεις στην επιστήμη
Stemmer	Πρόγραμμα που εφαρμόζει αλγόριθμο εξαγωγής της ρίζας μίας λέξης
Stemming	Η διαδικασία εξαγωγής της ρίζας μίας λέξης

Stopword	Πρόκειται για συγκεκριμένες λέξεις (ουσιαστικά) τα οποία είναι πολύ κοινότυπα στην καθομιλουμένη και συνεπώς δε μπορούν να αποτελέσουν τις λέξεις κλειδιά ενός κειμένου
Tag	Επικεφαλίδα. Ο όρος χρησιμοποιείται για τις δηλώσεις που χρησιμοποιούνται στη γλώσσα διαδικτύου HTML
Text Analysis	Ανάλυση κειμένου
Text Categorization	Κατηγοριοποίηση Κειμένου
Training Set	Σύνολο από κείμενα/λέξεις που μπορούν να χρησιμοποιηθούν για να αποκτήσει «γνώση» μία μηχανή.
User Profile	Προφίλ Χρήστη
Vector Space Model	Μοντέλο Κατηγοριοποίησης που βασίζεται στη χρήση διανυσμάτων και πινάκων
WWW	Παγκόσμιος Ιστός – Διαδίκτυο (World Wide Web)

Συντομογραφίες

DBMS	Database Management System
HTML	HyperText Mark-up Language
IF	Information Filtering
IR	Information Retrieval
LSI	Latent Semantic Indexing
RSS	Rich Site Summary
SVM	Support Vector Machine
URL	Uniform Resource Locator
VSM	Vector Space Model
WWW	World Wide Web
ΑΠ	Ανάκτηση Πληροφορίας
ΒΔ	Βάση Δεδομένων
ΠΣ	Πληροφοριακό Σύστημα

1

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό υπάρχουν εισαγωγικά στοιχεία για την εργασία

1. ΕΙΣΑΓΩΓΗ

Σκοπός της παρούσας Μεταπτυχιακής Εργασίας είναι η μελέτη και η αξιολόγηση των δυνατοτήτων προηγμένης σημασιολογικής αναζήτησης (advanced semantic search) πάνω σε περιεχόμενο που προέρχεται από το Διαδικτύο. Στα πλαίσια της εργασίας, σχεδιασθηκε και υλοποιήθηκε υποσύστημα, το οποίο ενσωματώθηκε και αξιολογήθηκε πάνω στο μηχανισμό peRSSonal ο οποίος ανακτά, επεξεργάζεται και παρουσιάζει στους χρήστες του άρθρα και υλικό από διάφορες ειδησεογραφικές πύλες (news portals) του Διαδικτύου, προσαρμόζοντάς τα στις προσωπικές επιλογές του χρήστη. Η αναζήτηση παραμετροποιείται με στοιχεία όπως: χρονικό πλαίσιο δημοσίευσης τού υλικού (διάστημα από – έως), κατηγορία (πρότυπες κατηγορίες του συστήματος peRSSonal), φυσική γλώσσα στην οποία είναι γραμμένο καθώς και δυνατότητα για στατικό ή σημασιολογικό (εννοιολογικό) ταίριασμα (semantic matching) με τα άρθρα της βάσης.

Αρχικά, από την επερώτηση (query) του χρήστη δημιουργείται ένα σύνολο ριζών (stems) των λέξεων οι οποίες δόθηκαν. Η εξαγωγή των ριζών εκτελείται με υποβοήθηση από stemming αλγορίθμο για την αγγλική γλώσσα, ενώ ο σχεδιασμός του συστήματος προβλέπει και τη μελλοντική υποστήριξη διαφορετικών φυσικών γλωσσών καταβάλλοντας μικρό κόπο. Για τις λεκτικές ρίζες που προκύπτουν, εντοπίζονται σχετικές τους και ταυτόχρονα με τη διαδικασία αυτή διενεργείται αναζήτηση στη βάση δεδομένων για κωδικολέξεις (keywords) με βάση την κατηγορία του άρθρου, ούτως ώστε να εμπλουτιστεί το ερώτημα του χρήστη με επιπλέον πληροφορία, καθιστώντας πιο επιτυχημένη και στοχευμένη την αναζήτηση στην πληθώρα των άρθρων που υπάρχουν αποθηκευμένα στη βάση δεδομένων. Για αυτές τις κωδικολέξεις υπολογίζονται συντελεστές – βάρη που θα προσδιορίζουν τη συνάφειά τους με την επερώτηση του χρήστη.

Ανάλογα με τον τύπο της αναζήτησης, στατική η σημασιολογική, το υποσύστημα αναζήτησης συγκρίνει την επερώτηση του χρήστη με τα αποθηκευμένα άρθρα και για κάθε ένα από αυτά, ο αλγόριθμος υπολογίζει το βαθμό συνάφειάς του με την επερώτηση. Τα άρθρα που επιλέγονται τελικά είναι αυτά που ξεπερνούν ένα κατώφλι συνάφειας, το οποίο τα κατατάσσει εννοιολογικά πιο κοντά στην επερώτηση του χρήστη. Σημαντικό σημείο στο στάδιο αυτό, είναι η δυνατότητα, για τους εγγεγραμένους χρήστες της Δικτυακής Πύλης, να εκτελείται περεταίρω φιλτράρισμα πάνω στο πρωτογενές αποτέλεσμα, βάσει των προσωπικών τους επιλογών καθώς και πληροφορίας που προέρχεται από τη βάση δεδομένων και που διαμορφώνεται δυναμικά από την παρατήρηση της γενικής συμπεριφοράς των χρηστών κατά την πλοήγηση τους μέσα στον σύστημα (χρόνος παραμονής στα άρθρα, άρθρα που δεν προτιμώνται, συχνότητα επιλογής άρθρων από μια δεδομένη θεματική ενότητα κλπ). Σκοπός είναι η εξαγωγή πιο στοχευμένου συνόλου άρθρων που ικανοποιεί τελικά περισσότερο τον χρήστη.

Τέλος, για την βελτίωση της απόδοσης του συστήματος, σχεδιάστηκε και υλοποιήθηκε αλγόριθμος που εκτελεί caching στα αποτελέσματα των επερωτήσεων. Με τον τρόπο αυτό, κάθε νέα αναζήτηση θα λαμβάνει πολύ πιο γρήγορα τα cached αποτελέσματα προγενέστερων παρόμοιων αναζητήσεων, ξοδεύοντας το χρόνο στα πιο πρόσφατα άρθρα. Το caching εκτελείται δυναμικά, τροποποιώντας σε κάθε επερώτηση που υποβάλλεται τα αντίστοιχα cached αποτελέσματα και μεταβάλλοντας τις προτεραιότητές τους και τα βάρη τους, ώστε να οδηγεί την έξοδο ολοένα και πιο κοντά στα επιθυμητά άρθρα και παραμένοντας πιο κοντά στο εξελισόμενο προφίλ και στις προτιμήσεις του χρήστη.

Μέσα από την εργασία, προέκυψαν αποτελέσματα που έχουν να κάνουν με σύγκριση αλγορίθμων σε όλα τα παραπάνω στάδια του μηχανισμού αλλά και ανταπόκριση του μηχανισμού στις ανάγκες του χρήστη. Η ερευνητική διατριβή που έγινε στα πλαίσια της συγκεκριμένης εργασίας οδήγησε στις παρακάτω δημοσιεύσεις:

Διεθνή Συνέδρια

Personalized News Search in WWW: Adapting on user's behavior. The Fourth International Conference on Internet and Web ASpplications and Services - ICIW 2009, Venice, Italy, C. Bouras, V. Poulopoulos, P. Silintziris, 24 - 28 May 2009, pp. 125 – 130

Abstract – Η προσωποποιημένη αναζήτηση στο Διαδίκτυο έχει γίνει στις μέρες μας ένα αρκετά υποσχόμενο ερευνητικό πεδίο στον τομέα της Ανάκτησης Πληροφορίας και του σχεδιασμού Μηχανών Αναζήτησης βελτιώνοντας και την ποιότητα των αποτελεσμάτων και την τελική εμπειρία του χρήστη. Στην εργασία αυτή, παρουσιάζουμε και αξιολογούμε το υποσύστημα, το οποίο εκτελεί την προηγμένη και προσωποποιημένη αναζήτηση στο peRSSonal, έναν διαδικτυακό μηχανισμό για την ανάκτηση, τη επεξεργασία και την παρουσίαση άρθρων και RSS feeds που συλλέγονται από μεγάλες ειδησεογραφικές πύλες του Παγκόσμιου Ιστού. Η προτεινόμενη τεχνική χρησιμοποιεί πληροφορία που παρέχεται άμεσα από τους χρήστες του συστήματος καθώς και από πληροφορία που μπορεί ο ίδιος ο μηχανισμός να συνθέσει μελετώντας και μαθαίνοντας από τη συμπεριφορά του χρήστη μέσα στο σύστημα κατά τη διάρκεια των επισκέψεων του σε αυτό. Καθώς αυτή η συμπεριφορά εξελίσσεται δυναμικά, το ίδιο συμβαίνει και στα ενδιαφέροντα του χρήστη υπό το πρίσμα της μηχανής αναζήτησης. Υιοθετώντας αυτή την προσέγγιση που βασίζεται στους χρήστες, καταφέρνουμε να παρουσιάσουμε ένα τελικό αποτέλεσμα που είναι πιο εστιασμένο στις ανάγκες του χρήστη και πιο ποιοτικό, αφού αξιοποιούμε τις προτιμήσεις του για να προσδιορίσουμε καλύτερα το αποτέλεσμα. Ο αλγόριθμος της αναζήτησης δεν λειτουργεί ανεξάρτητα αλλά συνεργάζεται και δένει με τα υπόλοιπα υποσυστήματα του μηχανισμού peRSSonal για να επιτύχει σε μεγαλύτερο βαθμό την ενσωμάτωση του στο μηχανισμό. Επιπλέον, αναπτύσουμε και προτείνουμε τη χρήση λογικής caching των αποτελεσμάτων των αναζητήσεων κάθε χρήστη προκειμένου να βελτιώσουμε το συνολικό χρόνο μελλοντικών παρόμοιων αναζητήσεων.

Date - based dynamic caching mechanism. IADIS European Conference on Data Mining 2009, Algavre, Portugal, C. Bouras, V. Poulopoulos, P. Silintziris, June 18 - 20 2009

Abstract – Οι Ειδησεογραφικές Διαδικτυακές πύλες που βασίζονται στο πρωτόκολλο RSS έχουν γίνει στις μέρες μας ένας από τους κυρίαρχους τρόπους που οι χρήστες του Διαδικτύου χρησιμοποιούν για να εντοπίζουν τις πληροφορίες και τις ειδήσεις που τους ενδιαφέρουν. Οι μηχανές αναζήτησης, οι οποίες λειτουργούν σε ένα μεγάλο μέρος αυτών των ιστοτόπων, δέχονται εκατομύρια επερωτήσεις κάθε μέρα. Και ενώ οι επερωτήσεις αυτές υποβάλλονται από χιλιάδες διαφορετικούς χρήστες, μελέτες έχουν δείξει ότι μικρά υποσύνολα από τις πιο δημοφιλείς επερωτήσεις αποτελούν το μεγαλύτερο κομμάτι του φόρτου που δέχονται οι μηχανές αναζήτησης. Μια δεύτερη αλήθεια έχει να κάνει με τη μεγάλη συχνότητα με την οποία ένας μεμονωμένος χρήστης έχει την τάση να υποβάλλει την ίδια ή παρόμοιες επερωτήσεις στο σύστημα. Συνδυάζοντας τα γεγονότα αυτά σε αυτή την εργασία, σχεδιάζουμε και αναλύουμε ένα αλγόριθμο caching που χρησιμοποιείται στο peRSSonal, έναν διαδικτυακό μηχανισμό για την ανάκτηση, τη επεξεργασία και την παρουσίαση άρθρων και RSS feeds που συλλέγονται από μεγάλες ειδησεογραφικές πύλες του Παγκόσμιου Ιστού. Χρησιμοποιώντας μικρό μέγεθος μνήμης και μειώνοντας την επιβάρυνση από σκοπιά υπολογισμών πετυχαίνουμε να αποθηκεύσουμε τα αποτελέσματα αναζητήσεων στον εξυπηρετητή τόσο για προσωποποιημένες όσο και για μη προσωποποιημένες αναζητήσεις. Ο αλγόριθμος που αναπτύσουμε δεν λειτουργεί ανεξάρτητα αλλά συνεργάζεται και δένει με τα υπόλοιπα υποσυστήματα του μηχανισμού peRSSonal για να επιτύχει σε μεγαλύτερο βαθμό την ενσωμάτωση του στο σύστημα.

2

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Στο κεφάλαιο αυτό περιγράφονται τα προβλήματα που καλείται να λύσει ο μηχανισμός που αναπτύχθηκε στη συγκεκριμένη εργασία.

Αναλυτικά:

- Συλλογή Δεδομένων
- Φιλτράρισμα Δεδομένων
- Προεπεξεργασία Πληροφορίας
- Προσωποποίηση στο Χρήστη
- Συμμετοχή του χρήστη στις διαδικασίες του συστήματος
- Προηγμένη Αναζήτηση

2. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η τελευταία δεκαετία μπορεί να χαρακτηριστεί ως η δεκατία που σήμανε την πιο μαζική είσοδο της τεχνολογίας στην καθημερινή ζωή. Όλο και περισσότεροι άνθρωποι σήμερα, με κυριότερο εκπρόσωπο τη νέα γενιά, αντιμετωπίζουν το Διαδίκτυο σαν ένα είδος σχεδόν «πρώτης ανάγκης». Το αγαθό που ονομάζεται Διαδίκτυο επεκτάθηκε δραματικά τα τελευταία χρόνια, με αποτέλεσμα να μιλούμε για μία κοινότητα χρηστών, η οποία απαρτίζεται από περισσότερα από 11 εκατομμύρια «σπίτια». Τα «σπίτια» αυτά αποτελούν τους δικτυακούς τόπους, σημεία ενημέρωσης και συνάντησης των χρηστών του διαδικτύου. Οι ιστότοποι αυτοί ανάλογα με τον σκοπό και τρόπο χρήσης τους και το κοινό στο οποίο απευθύνονται, μπορούν να κατηγοριοποιηθούν σε χώρους επικοινωνίας, ενημέρωσης, εμπορίου, διαφήμισης, πολιτισμού και διασκέδασης, ενώ για κάθε κατηγορία υπάρχουν αρκετές υποκατηγορίες. Ουσιαστικά λοιπόν μιλάμε για διαμόρφωση ενός νέου μοντέλου κοινωνίας βασισμένου σε ηλεκτρονικά πρότυπα με παροχή δημοκρατικών, ελευθέρων και μη υπηρεσιών. Όλο και περισσότερες εταιρίες και οργανισμοί καθώς επίσης και λιγότερο δομημένες ομάδες ανθρώπων διαθέτουν ένα δικτυακό τόπο σαν μέσο προβολής, σαν ένα τόπο συνάντησης, σαν ένα κομμάτι που είναι απαραίτητο για τη συμμόρφωση με τα νέα κοινωνικά πρότυπα που θέλουν το διαδίκτυο να αποτελεί αναπόσπαστο κομμάτι της ζωής μας.

Λόγω του τεράστιου όγκου πληροφορίας που διακινείται κάθε δευτερόλεπτο που περνά, της άναρχης δόμησης του, της ασαφούς νομοθεσίας γύρω από το τι επιτρέπεται και το τι όχι και της αίσθησης πλήρους ελευθερίας που δημιουργεί στους χρήστες του, αφού ο οποιοσδήποτε μπορεί να εκφράσει οποιαδήποτε άποψη δίχως φίλτρα και λογοκρισία έχουν προκύψει αρκετά προβλήματα στην κοινωνία του Διαδικτύου. Αντικείμενο της παρούσας εργασίας είναι η εξέταση εκείνης της μερίδας προβλημάτων που προκύπτουν από την διαρκή και κατεγιστική ροή δεδομένων και πληροφοριών. Θα εστιάσουμε σε εκείνα τα δεδομένα που προέρχονται από ενημερωτικές δικτυακές πύλες, τα γνωστά news portals, που έχουν κατακλύσει το Διαδίκτυο και τα οποία ως στόχο έχουν την ενημέρωση των χρηστών για τα πιο σημαντικά νέα σε παγκόσμιο επίπεδο. Μερικά και πολύ σημαντικά από αυτά είναι το CNN [114], το BBC [115], το Reuters [116], το FoxNews [117], καθώς και οι υπηρεσίες που προσφέρονται από τους πολυπληθείς και από τους πλέον αναγνωρίσιμους δικτυακούς τόπους Google [111] και Yahoo [112].

Σκοπός των προαναφερθέντων δικτυακών πυλών είναι η ενημέρωση των χρηστών για ότι διαδραματίζεται καθημερινά σε ολόκληρο τον κόσμο. Η παρουσίαση των νέων και των άρθρων φυσικά ακολουθεί μια συγκεκριμένη, και τις περισσότερες φορές καλά σχεδιασμένη δομή, παρ' όλα αυτά ο όγκος τους είναι τέτοιος, που καθίσταται πολύ δύσκολο για κάποιο μέσο χρήστη να παρακολουθήσει όλα τα άρθρα που δημοσιεύονται καθημερινά σε κάθε θεματική ενότητα-κατηγορία. Ακόμα και εστιάσουμε σε συγκεκριμένες κατηγορίες και αδιαφορήσουμε για τις υπόλοιπες, και πάλι, προκειμένου να επιτυγχάνεται πλήρης ενημέρωση πρέπει κάποιος να είναι μονίμως συντονισμένος στη δικτυακή πύλη. Όλα τα παραπάνω οδηγούν στην εξής προβληματική κατάσταση: οι χρήστες του διαδικτύου δυσκολεύονται στον εντοπισμό μίας είδησης που τους ενδιαφέρει με αποτέλεσμα να αναλώνουν το χρόνο τους στην αναζήτηση της είδησης, του νέου, του άρθρου, παρά στην ανάγνωση του ίδιου του άρθρου.

Στο περίπλοκο πρόβλημα του εντοπισμού ενός ενδιαφέροντος άρθρου στο διαδίκτυο μια πρώτη λύση έρχεται να δώσει η τεχνολογία και το πρότυπο RSS (Rich Site Summary), που σε ελεύθερη μετάφραση στα Ελληνικά θα μπορούσαμε να καθονομάσουμε ώς «Εμπλουτισμένη Περίληψη Ιστοσελίδας». Οι διαχειριστές των ειδησεογραφικών πυλών χρησιμοποιώντας το πρότυπο RSS αλλάζουν ριζικά την κατάσταση στο πεδίο της ηλεκτρονικής παγκόσμιας ειδησεογραφίας,

χαρίζοντας στους χρήστες ένα ακόμα δίαυλο επικοινωνίας και ενημέρωσης μέσω του Διαδικτύου. Ένα κανάλι RSS (RSS channel ή Web Feed) είναι μια ηλεκτρονική διεύθυνση στην οποία υπάρχουν ομαδοποιημένα διάφορα RSS έγγραφα και στην οποία συνδέονται οι χρήστες, χρησιμοποιώντας ένα ειδικό πρόγραμμα ανάγνωσης RSS feeds (RSS Reader) και έρχονται σε επαφή με αποκλειστικά με την πληροφορία που αναζητούν και όχι με όλο το περιεχόμενο της ηλεκτρονικής σελίδας στην οποία δημοσιεύθηκε το άρθρο, αποφεύγοντας με αυτό τον τρόπο τα «περιττά» επιπλέον στοιχεία της ιστοσελίδας. Η παρουσίαση της πληροφορίας ενός εγγράφου RSS γίνεται με δομημένο και ευδιάκριτο τρόπο για τους τελικούς χρήστες.

Σε δρόμο παράλληλο με την τεχνολογία του RSS, μια νέα τάση ξεκινά να επικρατεί στο διαδίκτυο, αυτή της προσωποποίησης στο χρήστη (personalization on user). Το Διαδίκτυο, ενώ στην αρχική του μορφή περιστρεφόταν γύρω από τους δικτυακούς τόπους έχοντας το χρήστη σε δευτερεύοντα ρόλο παρατηρητή, σήμερα έχει αρχίσει να προσανατολίζεται προς το χρήστη, καθιστώντας τον ως νέο άξονα αναφοράς στην παρουσίαση της πληροφορίας. Οι περισσότεροι ιστότοποι παρέχουν τη δυνατότητα για δωρεάν ή μη, εγγραφή των επισκεπτών τους με την ταυτόχρονη δημιουργία μιας ξεχωριστής σελίδας (preferences profile page) για κάθε ένα ξεχωριστά. Αυτό φυσικά δε σημαίνει ότι ο κάθε χρήστης γεμίζει τη ιστοσελίδα με προσωπικά του δεδομένα, απλά έχει το δικαίωμα να καθορίσει φίλτρα για το περιεχόμενο του διαδικτυακού τόπου που επιθυμεί να βλέπει στη δική του σελίδα καθώς και τον τρόπο με τον οποίο αυτό θα παρουσιάζεται. Και ενώ δίνεται με αυτόν τον τρόπο μια αίσθηση στον χρήστη ότι καθορίζει ο ίδιος τον τρόπο παρουσίασης των δεδομένων, η πραγματικότητα είναι αρκετά διαφορετική. Οι χρήστες γίνονται «δέσμιοι» αυτών των τεχνολογιών, που φαίνεται πως έρχονται, όχι μόνο εκμεταλλευόμενες την ανάπτυξη που παρουσιάζουν, αλλά για να πολεμήσουν τα «κανάλια επικοινωνίας» που απομάκρυναν τους χρήστες από τους δικτυακούς τόπους.

Διαμορφώνεται συνεπώς μια προβληματική κατάσταση με τα ακόλουθα χαρακτηριστικά:

- Οι χρήστες έχοντας κουραστεί ή βαρεθεί από τον όγκο της ανούσιας πληροφορίας που τους παρουσιάζεται έως ότου φτάσουν στην επιθυμητή πληροφορία, καταφεύγουν στη λύση των RSS.
- Λόγω της παραπάνω αλλαγής στον τρόπο «επίσκεψης» σε μια ιστοσελίδα, τα μεγάλα ειδησεογραφικά πρακτορεία παρατηρούν τη μείωση της «επισκεψιμότητας» των σελίδων του δικτυακού τους τόπου, ο οποίος ουσιαστικά παρακάμπτεται από το χρήστη, μιας και αυτός αρκείται στο κανάλι επικοινωνίας που έχει και αποφεύγει κάθε άμεση επίσκεψη σε αυτόν.
- Επιπρόσθετα, η ανάπτυξη νέων τεχνολογιών και παρεχόμενων υπηρεσιών για το Διαδίκτυο κάνει τους διαχειριστές δικτυακών τόπων να αποζητούν ακόμα μεγαλύτερη επισκεψιμότητα στις σελίδες τους προσφέροντας διαδραστικές υπηρεσίες, υπηρεσίες πολυμέσων κ.α.
- Γεννιέται μια διαμάχη ανάμεσα στις υπηρεσίες που απομακρύνουν τους χρήστες από τους δικτυακούς τόπους (RSS feeds) και σε αυτές που τους ωθούν προς αυτούς (προσωποποιημένης αναζήτησης), με τις δεύτερες να έχουν το συγκριτικό πλεονέκτημα λόγω της αυξημένης παροχής στοιχείων.

Είναι φανερό πως οι δικτυακοί τόποι, λειτουργώντας λιγο-πολύ ως επιχειρήσεις, επιδιώκουν την προσέλκυση χρηστών και την επίσκεψη αν είναι εφικτό όλων των σελίδων τους, ώστε να προβάλλονται όλες οι διαφημίσεις, να αξιοποιούνται οι προσφερόμενες υπηρεσίες και να χρησιμοποιείται και το τελευταίο bit πληροφορίας. Σε αυτούς τους δικτυακούς τόπους χρησιμοποιούνται μηχανές αναζήτησης ώστε να βοηθούν τους χρήστες να εντοπίζουν την πληροφορία που τους ενδιαφέρει καταβάλλοντας μικρότερο κόπο. Στην πλειοψηφία αυτών των μηχανών αναζήτησης, όταν διαφορετικοί χρήστες υποβάλλουν την ίδια επερώτηση στο σύστημα, επιστρέφονται τα ίδια αποτελέσματα και με την ίδια σειρά,

ανεξάρτητα από το ποιος υποβάλλει την επερώτηση. Προφανώς, είναι κάπως απίθανο όλοι οι χρήστες μιας μηχανής αναζήτησης να έχουν ακριβώς τις ίδιες ανάγκες σε πληροφόρηση. Για το λόγο αυτό δεν μπορεί να υπάρχει μια και μόνη προσέγγιση στο θέμα της αναζήτησης. Πραγματικά, έρευνες διαπίστωσαν ότι πάνω από τα μισά επιστρεφόμενα αποτελέσματα είναι σχεδόν άσχετα με την αρχική ανάγκη ενός χρήστη [94]. Επιπρόσθετα, ένας αριθμός μελετών έχει δείξει ότι η συντριπτική πλειοψηφία των επερωτήσεων σε μηχανές αναζήτησης είναι σύντομες και ελάχιστα περιγραφικές [95] και ότι διαφορετικοί χρήστες μπορούν να έχουν τελειώς διαφορετικές απαιτήσεις από την ίδια ακριβώς επερώτηση [96]. Η εξήγηση είναι πολύ απλή αφού μια λέξη της επερώτησης ή ένα υποσύνολο λέξεων δεν μπορεί να είναι πάντοτε τρόπος για να καθορίσουμε χωρίς αμφιβολία τι έιναι αυτό που επιθυμούσε ο χρήστης. Αυτό ακριβώς είναι το σημείο που υπεισέρχεται η προσωποποιημένη αναζήτηση. Υποθετικά, η ανάκτηση της πληροφορίας θα είναι περισσότερο αποτελεσματική αν ληφθούν υπόψιν οι ιδιοσυγκρασίες διαφορετικών χρηστών. Μια τέτοια στρατηγική θα μπορούσε να αποφασίζει αυτόνομα για κάθε χρήστη το αν ένα συγκεκριμένο άρθρο-αποτέλεσμα είναι όντως ενδιαφέρον για αυτόν και σε περίπτωση που δεν είναι να μην το εμφανίσει καθόλου στο τελικό αποτέλεσμα. Μοντελοποιώντας το χρήστη και τις προτιμήσεις του και εκτελέντας προσωποποιημένη αναζήτηση με βάση τις ατομικές του ανάγκες, μπορούμε να επιτύχουμε μεγαλύτερη ακρίβεια και ποιότητα στην ανάκτηση της σωστής πληροφορίας.

Λαμβάνοντας υπόψιν αυτή τη διαμορφωμένη κατάσταση, προκαλεί προβληματισμό το γεγονός ότι οι σχεδιαστές των υπηρεσιών αυτών έχουν παραλείψει σημαντικά στοιχεία και γεννιώνται διάφορα ερωτήματα: πόσο εξοικιωμένοι είναι οι χρήστες με αυτά τα πολύπλοκα συστήματα; Έχουν όλοι την ίδια αρκετά μεγάλη ταχύτητα για να αξιοποιούν τις παρεχόμενες υπηρεσίες; Έχει ληφθεί υπόψιν η άποψη τους για το τί θα επιθυμούσαν να τους παρέχεται; Κατάληξη όλων των παραπάνω είναι η ύπαρξη προσωποποιήμενων συστημάτων, όπου ο χρήστης αδυνατεί να πλοηγηθεί όπως επιθυμεί αφού χάνεται σε ένα κυκεώνα δεδομένων που του προβάλλονται, καθώς επίσης και η υπερπληθώρα καναλιών RSS που συμπληρώνουν ένα χαοτικό σκηνικό. Τρανταχτό παράδειγμα αποτελεί το RSS feed του CNN που αποτελείται από περισσότερα από 20 επικαλυπτόμενα κανάλια.

Συνοψίζοντας όσα προαναφέρθηκαν και εντοπίζοντας την έλλειψη λύσεων που να συνδυάζουν και τις δύο προοπτικές-τεχνολογίες, καταλήγουμε στα παρακάτω:

<u>Χαρακτηριστικά προσωποποιημένων σελιδών</u>:

- Είναι δύσχρηστες και πολύπλοκες.
- Χρησιμοποιούν, σε πολύ μεγάλο ποσοστό, μόνον λέξεις-κλειδιά και θεματικές ενότητες για την αναζήτηση.
- Σε κάθε περίπτωση, ο χρήστης είναι εκτός της διαδικασίας κατηγοριοποίησης ή κατασκευής περίληψης που παρουσιάζεται στην προσωποποιημένη σελίδα.

<u>RSS Feeds</u>:

- Υπάρχει ένας τεράστιος και ανεξέλεγκτος αριθμός καναλιών με μεγάλο βαθμό επικάλυψης του περιεχομένου μεταξύ τους.
- Ακόμα και μέσα στο ίδιο κανάλι ο αριθμός των RSS feeds είναι ιδιαίτερα αυξημένος.
- Δεν είναι εύκολο για μέσους χρήστες να τα χρησιμοποιήσουν σωστά.

Όπως είναι αναμενόμενο, χαμένος βγαίνει ο χρήστης, αφού η αναζήτηση και η παρακολούθηση ειδήσεων γίνεται δύσκολη και αναποτελεσματική. Αυτό που χρειάζεται είναι να καταστεί ο χρήστης άξονας αναφοράς, διαμορφώνοντας την κατηγοριοποίηση σε θεματικές ενότητες αλλά και τον τρόπο παρουσίασης των αποτελεσμάτων της αναζήτησης. Στη συνέχεια του κεφαλαίου, θα παρουσιασθεί

συνοπτικά κάθε διαδικασία του συστήματος και θα εξετασθούν τα προβλήματα που εντοπίζονται σε κάθε μια από αυτές τις διαδικασίες. Η συνολική διαδικασία είναι σειριακή με στόχο να παράγει το επιθυμητό αποτέλεσμα, δηλαδή την παρουσίαση προσωποποιημένων, κατηγοριοποιημένων άρθρων στον τελικό χρήστη.

2.1. Συλλογή δεδομένων

Η συλλογή των δεδομένων είναι ένα πολύ σημαντικό κομμάτι ενός μηχανισμού σαν αυτό που θέλουμε να κατασκευάσουμε αλλά και γενικότερα ένα πολύ σημαντικό κομμάτι των μηχανισμών αναζήτησης και των μηχανισμών που βασίζονται στη συλλογή πληροφορίας. Στην περίπτωσή μας η συλλογή δεδομένων περιορίζεται στη συλλογή άρθρων από μεγάλους ειδησεογραφικούς πληροφοριακούς κόμβους. Το πρόβλημα συλλογής των κυριότερων νέων είναι μεγάλο καθότι αν παρατηρήσουμε τη δομή και οργάνωση αυτών των σελίδων, αποτελεί πρόβλημα ο εντοπισμός αυτών των σελίδων αλλά και η συλλογή των πιο πρόσφατων ειδήσεων που είναι και το ζητούμενο.

Η συλλογή δεδομένων βασίζεται σε μηχανισμούς που περιδιαβαίνουν ολόκληρους τους ειδησεογραφικούς κόμβους και εντοπίζουν τα σημεία εκείνα που περιέχουν αρκετό κείμενο συγκριτικά με άλλες σελίδες που αποτελούν κεντρικούς κόμβους πληροφοριών.

Ωστόσο, οι νέες τεχνολογίες και κυρίως τα κανάλια επικοινωνίας που χρησιμοποιούνται από τους σύγχρονους δικτυακούς τόπους μπορούν να διευκολύνουν το πρόβλημα της συλλογής δεδομένων. Οι μηχανισμοί δεν είναι υποχρεωμένοι να «ανακαλύπτουν» τις πολλαπλές δυναμικές σελίδες που ανανεώνονται καθημερινά στους δικτυακούς τόπους. Αρκεί η συλλογή πληροφοριών από τα κανάλια επικοινωνίας που υπάρχουν για τη συγκέντρωση των πιο σημαντικών αλλαγών που προκύπτουν καθημερινά και εν προκειμένω τα νέα άρθρα που προστίθενται στα ειδησεογραφικά portal.

2.2. Φιλτράρισμα δεδομένων

Η συλλογή πληροφοριών έχει σαν αποτέλεσμα σελίδες που περιέχουν κυρίως HTML κώδικα, στον οποίο βεβαίως μπορούμε να εντοπίσουμε και το κείμενο το οποίο επιθυμούμε να εξάγουμε από τη σελίδα και το οποίο αποτελεί το κύριο σώμα του άρθρου. Για το φιλτράρισμα τέτοιου είδους δεδομένων έχουν γίνει πολλές προτάσεις, κυρίως για τον τρόπο με τον οποίο μπορεί να εξαχθεί και βασικά να εντοπιστεί μέσα στη σελίδα. Το πρόβλημα σε αυτή την περίπτωση είναι η απομόνωση του χρήσιμου μόνο κειμένου, το οποίο στην περίπτωση που εξετάζουμε είναι το σώμα του άρθρου αλλά και ο τίτλος του.

2.3. Προεπεξεργασία πληροφορίας

Η προεπεξεργασία πληροφορίας είναι μία διαδικασία κατά την οποία το χρήσιμο κείμενο υπόκειται σε διαδικασία αφαίρεσης των σημείων στίξης, των αριθμών που τυχόν περιέχει, αφαίρεση λέξεων οι οποίες δεν περικλείουν κάποιο νόημα και τέλος το πολύ σημαντικό κομμάτι του stemming το οποίο είναι η διαδικασία εύρεσης της ρίζας μίας λέξης. Σαν αποτέλεσμα έχει την εξαγωγή των κωδικολέξεων που υπάρχουν στο κείμενο, συνοδευμένο από τη συχνότητα την οποία παρουσιάζουν μέσα στο κείμενο αλλά και το σημείο του κειμένου στο οποίο εντοπίζονται. Για τους μηχανισμούς εξαγωγής κειμένου και απόρριψης οποιασδήποτε πληροφορίας που δεν σχετίζεται με το κείμενο, η προεπεξεργασία πληροφορίας είναι μία πρόκληση. Παρά το γεγονός ότι βασίζεται σε συγκεκριμένα και σταθερά βήματα, θα πρέπει να γίνει εκτενής ανάλυση του είδους της πληροφορίας που είναι επιθυμητή, προκειμένου το βήμα της προεπεξεργασίας να καταλήξει σε σημαντικά αποτελέσματα και πιο συγκεκριμένα στην εξαγωγή των σωστών κωδικολέξεων.

2.4. Προσωποποίηση στο χρήστη

Η προσωποποίηση στο χρήστη είναι διαδικασία κατά την οποία τα αποτελέσματα που εμφανίζονται τελικά στο χρήστη προσαρμόζονται, προκειμένου να είναι ικανοποιήσουν πιο αποτελεσματικά την αρχική του ανάγκη. Πιο συγκεκριμένα, τα στάδια της προσωποποίησης αφορούν στον εντοπισμό άρθρων τα οποία ενδιαφέρουν το χρήστη και στην παρουσίασή τους με τέτοιον τρόπο, ώστε να ταιριάζουν στις ανάγκες και το προφίλ (προτιμήσεις) του χρήστη. Το πρόβλημα που τίθεται είναι η εύρεση ενός «έξυπνος» αλγόριθμος ο οποίος θα μπορεί να αξιοποιεί όλες τις πληροφορίες που μπορούν να συγκεντρωθούν από την περιήγηση του χρήστη στο δικτυακό τόπο και αξιοποίηση αυτών των πληροφοριών προκειμένου να εμφανιστούν όσο το δυνατόν καλύτερα και πιο ποιοτικά αποτελέσματα.

2.5. Συμμετοχή του χρήστη στις διαδικασίες του συστήματος

Ο χρήστης είναι αυτός που δέχεται την τελική πληροφορία και αυτός που ουσιαστικά αξιολογεί την πληροφορία για τον εαυτό του. Αυτό σημαίνει πως ο χρήστης θα πρέπει να είναι αναπόσπαστο κομμάτι του συστήματος. Θα πρέπει να είναι σε θέση να διαμορφώσει διαδικασίες του πυρήνα του συστήματος όπως είναι η κατηγοριοποίηση και η εξαγωγή περίληψης.

Στα περισσότερα συστήματα τα οποία αντιμετωπίστηκαν κατά τη διάρκεια της μελέτης για τη συγκεκριμένη εργασία, παρατηρήθηκε πως ο χρήστης συμμετέχει μόνο στα επιτελικά στάδια των συστημάτων ενώ έχουν ήδη εκτελεστεί τα βασικά βήματα του πυρήνα των μηχανισμών. Η συμμετοχή του χρήστη στις διαδικασίες πυρήνα ενός large scale συστήματος είναι επίπονη διαδικασία η οποία απαιτεί αλγορίθμους που θα μπορούν να εκτελούνται αποδοτικά σε πραγματικό χρόνο προκειμένου ο χρήστης να διαμορφώνει όχι μόνον τα τελικά αποτελέσματα που εμφανίζονται σε αυτόν αλλά και συγκεκριμένες διαδικασίες ολόκληρου του συστήματος.

2.6. Προηγμένη Αναζήτηση

Η αναζήτηση είναι η διαδικασία υποβολής επερώτησης από το χρήστη που αποτελείται από μια η περισσότερες λέξεις - κλειδιά. Η διαδικασία αναζήτησης είναι ένα από τα πιο σημαντικά υποσυστήματα αφού αποτελεί έναν από τους πιο άμεσους τρόπους επικοινωνίας και αλληλεπίδρασης του χρήστη με το σύστημα. Σε ένα μεγάλο σύστημα, στο οποίο υπάρχουν αποθηκευμένα χιλιάδες άρθρα και που η βάση δεδομένων του μεγαλώνει κάθε μέρα με νέα άρθρα, είναι πολύ σημαντικό ο χρήστης να μπορεί να εντοπίζει σχετικά εύκολα την πληροφορία που επιθυμεί καλύτερα τις ανάγκες του σε πληροφόρηση. Χρειάζεται να προσφέρεται η δυνατότητα στο χρήστη να προσδιορίσει και άλλες παραμέτρους προκειμένου να περιγράψει με καλύτερη λεπτομέρια τα επιθυμητά από αυτόν αποτελέσματα εκτός από τις λέξεις - κλειδιά που εννοούνται. Οι παράμετροι πρέπει να καθορίζονται στη φόρμα αναζήτησης χρησιμοποιούνται σε διάφορα στάδια της ανάκτησης των άρθρων του τελικού αποτελέσματος όπως κατά την αναζήτηση ήδη αποθηκευμένων αποτελεσμάτων στην μνήμη cache.

3

STATE OF THE ART

Στο κεφάλαιο αυτό περιγράφεται το State of the Art για κάθε υποσύστημα του μηχανισμού που θα κατασκευάσουμε. Πιο συγκεκριμένα υπάρχουν στοιχεία για τις εξής θεματικές ενότητες:

- Σημασιολογικός Ιστός
- Εξόρυξη Πληροφορίας
- Μοντέλα Ανάκτησης Πληροφορίας
- Προεπεξεργασία Δεδομένων
- Αξιοποίηση Πληροφορίας
- Προφίλ Χρήστη σε Δυναμικά Περιβάλλοντα

3. STATE OF THE ART

Το παρόν κεφάλαιο θα αποτελέσει μια παρουσίαση των ζητημάτων με τα οποία θα καταπιαστούμε στη συγκεκριμένη εργασία. Επίσης, θα γίνει μια σύντομη ανάλυση για κάθε ένα από αυτά τα ζητήματα. Σκοπός είναι να δημιουργηθεί η κατάλληλη βάση για να μπορέσουμε στη συνέχεια να κατανοήσουμε τις έννοιες που θα χρησιμοποιηθούν στα επόμενα κεφάλαια. Επίσης, θα είναι πιο εύκολο να παρουσιαστούν οι πιο σημαντικές τεχνολογίες του τομέα. Θα παρουσιάσουμε συνοπτικά με έννοιες περί της ανάκτησης πληροφορίας από το διαδίκτυο, της εξαγωγής της χρήσιμης πληροφορίας από τις σελίδες που έχουν ανακτηθεί, του τρόπου με τον οποίο δημιουργείται το δυναμικό προφίλ των χρηστών και φυσικά θα εξετάσουμε το κομμάτι που έχει να κάνει με την προσωποποίση της διαδικασίας της αναζήτησης, με βάση τις ανάγκες του χρήστη. Τέλος, θα θίξουμε θέματα που έχουν να κάνουν με την άυξηση της ταχύτητας της αναζήτησης.

3.1. Σημασιολογικός Ιστός και Προβλήματα

Το Διαδίκτυο σήμερα αποτελεί τη μεγαλύτερη πηγή πληροφοριών. Μεγάλοι όγκοι δεδομένων αναζητούνται, ανταλλάσσονται και επεξεργάζονται μέσω του Παγκόσμιου Ιστού. Επειδή, όμως ο όγκος των δεδομένων του Ιστού έχει πάρει μεγάλες διαστάσεις χωρίς να υπάρχει ενιαίος τρόπος οργάνωσης, η ανταλλαγή και η επεξεργασία τους είναι πολύ δύσκολη. Ο Σημασιολογικός Ιστός έρχεται ακριβώς να εξυπηρετήσει την ανάγκη για ενιαία οργάνωση των δεδομένων, ώστε το Διαδίκτυο να γίνει μια αποδοτική παγκόσμια πλατφόρμα ανταλλαγής και επεξεργασίας από ετερογενείς πηγές πληροφορίας. Ένας γενικός ορισμός μας λέει ότι ο Σημασιολογικός Ιστός δίνει δομή, οργάνωση και σημασιολογία στα δεδομένα, ώστε να είναι, σε μεγάλο βαθμό, κατανοητά από μηχανές (machine understandable).

Ο όρος Σημασιολογικός Ιστός (Semantic Web) χρησιμοποιήθηκε για πρώτη φορά το 1998 από το δημιουργό του πρώτου φυλλομετρητή ιστοσελίδων και εξυπηρετητή διαδικτύου, Tim Berners-Lee. Από τότε καταβάλλεται μεγάλη προσπάθεια από την επιστημονική κοινότητα για την υλοποίησή του πάνω από τον Παγκόσμιο Ιστό. Στο βασικότερο επίπεδό του, ο Σημασιολογικός Ιστός αποτελεί μία συλλογή από συνοπτική πληροφορία για τη διακινούμενη πληροφορία, τα μεταδεδομένα, η οποία δεν είναι ορατή στον τελικό χρήστη. Τα μεταδεδομένα χρησιμοποιούνται για να περιγράψουν υπάρχοντα έγγραφα, ιστοσελίδες, βάσεις δεδομένων, προγράμματα που βρίσκονται στο διαδίκτυο. Οι εφαρμογές λογισμικού που κάνουν χρήση μεταδεδομένων αποκτούν καλύτερη κατανόηση της σημασιολογίας του περιεχομένου τους και άρα μπορούν να τα επεξεργαστούν με πιο αποδοτικό τρόπο. Η κατανόηση των μεταδεδομένων από τις μηχανές είναι δυνατή μέσω της χρήσης ειδικών λεξικών (των οντολογιών) τα οποία παρέχουν κοινούς κανόνες και λεξιλόγια για την ερμηνεία των δεδομένων. Με αυτό τον τρόπο είναι δυνατή η κοινή κατανόηση όρων και εννοιών από εφαρμογές που προέρχονται από διαφορετικά πληροφοριακά συστήματα. Απώτερος στόχος της όλης προσπάθειας είναι η ικανοποίηση των απαιτήσεων των συμμετεχόντων στην Κοινωνία της Πληροφορία για αυξημένη ποιότητα υπηρεσιών. Αυτό συνίσταται κυρίως στη βελτιωμένη αναζήτηση, εκτέλεση σύνθετων διεργασιών μέσω του Διαδικτύου και στην εξατομίκευση της πληροφορίας σύμφωνα με τις ανάγκες του εκάστοτε χρήστη.

Ένα από τα σημαντικότερα προβλήματα που καλείται να λύσει ο Σημασιολογικός Ιστός είναι η πρόσβαση στην πληροφορία. Σύμφωνα με πρόσφατες μελέτες, η ανθρωπότητα έχει παράγει από το 1999 μέχρι το 2003, τόσες νέες πληροφορίες όσες παρήγαγε όλα τα προηγούμενα χρόνια της ιστορίας της. Σε αυτό το διάστημα των τριών τελευταίων ετών παρήχθησαν 12 exabytes πληροφορίας υπό τη μορφή έντυπου, οπτικού ή και ηχητικού υλικού. Η

αυξανόμενη αυτή παραγωγή και η συνεχής βελτίωση των μεθόδων ψηφιοποίησης συμβάλλουν στην παραγωγή ενός ωκεανού ψηφιακών δεδομένων που προφανώς δύναται να δημιουργήσει μεγάλο αριθμό προβλημάτων. Το πιο σημαντικό ίσως από αυτά είναι ο τρόπος με τον οποίο θα μπορεί κανείς να διαχειριστεί όλη αυτή την πληροφορία. Δε θα πρέπει φυσικά να αμελούμε το γεγονός πως η ικανότητα παραγωγής, αποθήκευσης και μετάδοσης της πληροφορίας έχει ξεπεράσει κατά πολύ τις δυνατότητες αναζήτησης, πρόσβασης και παρουσίασης.

Λόγω του αυξανόμενου όγκου της πληροφορίας και των προβλημάτων αποτελεσματικής πρόσβασης, έχει γίνει τα τελευταία χρόνια ξεκάθαρο προς την επιστημονική κοινότητα ότι για την αύξηση της απόδοσης χρειάζονται νέες μέθοδοι υπολογισμού ικανές να προσαρμοστούν σε μία πληθώρα παραμέτρων τόσο αντικειμενικών όσο και υποκειμενικών. Η απόδοση ενός συστήματος πρόσβασης στην πληροφορία εκτιμάται μέσα από την ανάκληση και την ακρίβεια.

Η αναφορά στα προβλήματα που αντιμετωπίζουν τα σύγχρονα συστήματα πρόσβασης στην πληροφορία έχει άμεση σχέση με τον τύπο των ερωτήσεων που δέχονται ως είσοδο. Υπάρχουν δύο διαφορετικά είδη ερωτημάτων, οι ερωτήσεις γενικού περιεχομένου και ειδικού περιεχομένου. Το μέγεθος της απάντησης σε ερωτήσεις γενικού περιεχομένου είναι μεγάλο και παρουσιάζει εξαιρετικά μεγάλες αποκλίσεις ως προς τη σχετικότητα της ίδιας της ερώτησης. Το πρόβλημα εστιάζεται στην επιλογή ενός μικρού συνόλου από τις πιο σχετικές απαντήσεις, είναι δηλαδή πρόβλημα ακρίβειας. Αντίθετα, για τις ερωτήσεις ειδικού περιεχομένου, το διαθέσιμο σύνολο σχετικών απαντήσεων είναι μικρό και το πρόβλημα που προκύπτει είναι πρόβλημα ανάκτησης.

3.2. Εξόρυξη πληροφορίας από το Διαδίκτυο

Εξόρυξη πληροφορίας από το Διαδίκτυο ονομάζεται κάθε διαδικασία που έχει σαν αποτέλεσμα ανάκτηση πληροφορίας (Information Retrieval) από τον παγκόσμιο ιστό. Στο εξής θα αναφερόμαστε στον όρο ανάκτηση πληροφορίας ως IR για συντομία. Η ανακτώμενη πληροφορία δεν περιορίζεται απλώς σε σελίδες HTML, αλλά μπορεί να είναι και αρχεία πολυμέσων ή οποιοδήποτε είδος αρχείου μπορεί να μεταφερθεί πάνω από το Διαδίκτυο. Η ανάγκη για ανάκτηση πληροφορίας πηγάζει από τις αρχές της δεκαετίας του 50 όταν ο Mooers [1] εξέφρασε ανοιχτά σε δημοσίευσή του την ανάγκη για ανάκτηση πληροφορίας. Αργότερα, στη δεκαετία του 60, το IR είχε γίνει πλέον ένα πολύ δημοφιλές θέμα καθώς πολλοί ερευνητές πίστευαν ότι μπορούν να αυτοματοποιήσουν μέχρι τότε χειροκίνητες διαδικασίες όπως η δεικτοδότηση και η αναζήτηση [2, 3].

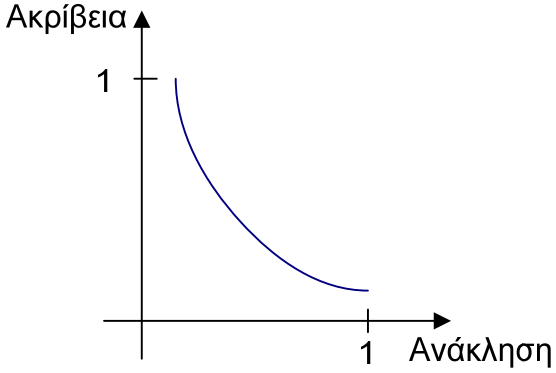
Προκειμένου να πετύχει το στόχο της η κοινότητα IR όρισε δύο βασικές ενέργειες που έχουν γίνει αντικείμενα έρευνας για πολλά χρόνια και είναι: η δεικτοδότηση και η αναζήτηση. Η δεικτοδότηση αναφέρεται στον τρόπο με τον οποίο αναπαρίσταται η πληροφορία για τους σκοπούς της ανάκτησης. Η αναζήτηση αναφέρεται στον τρόπο με τον οποίο δομείται η πληροφορία όταν πραγματοποιείται ένα ερώτημα. Παρόλο που οι δύο αυτές διαδικασίες αποτελούν τον πυρήνα ενός συστήματος IR, άλλες διαδικασίες είναι αυτές που κερδίζουν έδαφος, όπως τεχνικές αναπαράστασης της πληροφορίας, με σκοπό να βελτιωθεί η αποτελεσματικότητα της ανάκτησης [4].

Στην παρούσα φάση, το IR αντιμετωπίζει μία σειρά από θέματα. Αρχικά, εφαρμόστηκε σε Βάσεις Δεδομένων βιβλιοθηκών, όπου σε ένα αρχείο αποθηκεύονταν γενικά χαρακτηριστικά κάθε εγγράφου, όπως ο τίτλος και ο συγγραφέας, και η αναζήτηση γινόταν βάσει αυτών των στοιχείων. Στη συνέχεια, και εξ αιτίας της αύξησης του μεγέθους των αποθηκευτικών μέσων, ολόκληρο το κείμενο αποθηκευόταν σε αρχείο και η αναζήτηση ήταν εφικτή σε ολόκληρες συλλογές από κείμενα. Έτσι μέχρι ενός σημείου το IR αντιπροσώπευε την ανάκτηση κειμένων. Αργότερα και έως σήμερα, δίνεται περισσότερη σημασία στον όρο πληροφορία. Άλλωστε σήμερα δεν έχουμε μόνο έγγραφα πάνω στα οποία γίνεται η αναζήτηση αλλά και αρχεία πολυμέσων. Ωστόσο το βασικό κλειδί στην

υπόθεση του IR είναι ανάκτηση κειμένων ή πληροφορίας που προσεγγίζουν περισσότερο τις ανάγκες του χρήστη που πραγματοποιεί την αναζήτηση.

Ένα από τα βασικά στοιχεία του IR είναι η μέτρηση του κατά πόσο τα ανακτημένα κείμενα είναι σχετικά με το ερώτημα που κάνουμε [5]. Έτσι λοιπόν, ένα βασικό στοιχείο στο οποίο εστιάζουμε είναι η εύρεση μετρικών που θα μπορούν να αναπαραστήσουν αριθμητικά τη σχετικότητα των αποτελεσμάτων ενός συστήματος IR. Πολλές μετρικές έχουν αναπτυχθεί με τις δύο πιο γνωστές να είναι η ανάκληση και η ακρίβεια. Η ακρίβεια μας δίνει το επί τοις εκατό ποσοστό των σχετικών κειμένων εν συγκρίσει με αυτά που ανακτήθηκαν ενώ η ανάκληση μας δίνει το επι τοις εκατό ποσοστό των κειμένων που ανακτήθηκαν εν συγκρίσει με μία συλλογή που γνωρίζουμε ότι περιέχει όλα τα σχετικά.

Η συνηθισμένη απόκριση που έχει ένα σύστημα IR είναι αυτή που παρουσιάζεται στην Εικόνα 1, στην οποία φαίνεται ότι τα μεγέθη ακρίβεια και ανάκληση είναι αντιστρόφως ανάλογα. Αυτό σημαίνει πως για αν αυξήσουμε την ανάκληση θα μειωθεί η ακρίβεια. Φυσικά ισχύει και το αντίστροφο [6].



Εικόνα 1: Σχεδιάγραμμα ακρίβειας – ανάκλησης

Ένα σύστημα IR μπορεί να πετύχει κατά μέσο όρο περίπου 30% ανάκληση και 30% ακρίβεια. Οι τιμές αυτές δεν έχουν καμία σύγκριση με ένα σύστημα διαχείρισης βάσης δεδομένων (DBMS) που τα ποσοστά αυτού προσεγγίζουν το 100%. Ωστόσο θα μπορούσε κανείς να πει πως και τα δύο συστήματα πραγματοποιούν την ίδια διαδικασία, δηλαδή ανάκτηση πληροφορίας. Αυτό βέβαια έχει να κάνει με τον τρόπο με τον οποίο δομείται ένα σύστημα DBMS και ο οποίος είναι τέτοιος ώστε να εξυπηρετεί απόλυτα τις ανάγκες ενός χρήστη.

Αυτή η δυσκολία που αντιμετωπίζουν τα συστήματα IR (μικρές τιμές ανάκλησης και ακρίβειας) γεννούν ένα άλλο επιστημονικό πεδίο το οποίο υπάρχει παράλληλα με το IR και είναι το IF (Information Filtering). Σε ένα κλασσικό άρθρο οι Belkin και Croft παρουσίασαν δύο διαφορετικούς ορισμούς για τα δύο παραπάνω θέματα οι οποίοι έχουν κοινές τεχνικές αλλά διαφέρουν σε τρία βασικά στοιχεία [7]. Πρώτον, στο IR όταν ο χρήστης κάνει ένα ερώτημα περιμένει άμεση απόκριση. Στο IF ο χρήστης μπορεί να περιμένει, εν γνώσει του, για μεγάλο χρονικό διάστημα μέχρι να του παρουσιαστεί μία απάντηση. Επιπρόσθετα το IF χειρίζεται και θέματα που από τη φύση του είναι δυναμικά και εντάσσει στο μηχανισμού του στοιχεία εκμάθησης σύμφωνα με τα κείμενα που προσθέτει στη συλλογή του. Τέλος, το βασικότερο είναι πως το IR αναζητά παραπλήσια κείμενα από μία μεγάλη συλλογή κειμένων σε αντίθεση με το IF το οποίο προσπαθεί να αφαιρέσει από μία συλλογή τα εισερχόμενα κείμενα που δεν είναι σχετικά.

Παρ' όλες τις διαφορές που έχουν τα δύο αυτά πεδία δεν πρέπει να αμελούμε πως έχουν παραπλήσιο σκοπό: να εξασφαλίσουν ότι τα κείμενα που θα παρουσιαστούν στο χρήστη είναι σχετικά με το ερώτημά του.

Τα διαγράμματα ακρίβειας/ανάκλησης είναι χρήσιμα εφόσον μελετούμε την απόδοση ανάκτησης διαφορετικών αλγορίθμων σε ένα σύνολο από πρότυπες πληροφοριακές ανάγκες. Ωστόσο υπάρχουν περιπτώσεις στις οποίες θα θέλαμε να

συγκρίνουμε την απόδοση αλγορίθμων ανάκτησης για ατομικές πληροφοριακές ανάγκες. Οι λόγοι για να το κάνουμε αυτό είναι δύο:

1. η χρήση μέσων τιμών που προκύπτουν από την εκτέλεση διαφόρων ερωτημάτων μπορεί να αποκρύπτει σημαντικές ανωμαλίες στον αλγόριθμο ανάκτησης
2. όταν συγκρίνουμε δύο αλγορίθμους μπορεί να θέλουμε να μελετήσουμε κατά πόσο ο ένας είναι καλύτερος του άλλου για κάθε μία από τις πληροφοριακές ανάγκες που έχουμε και όχι συνολικά.

Σε τέτοιες περιπτώσεις υπολογίζουμε μία μόνο τιμή ακρίβειας για κάθε ερώτημα, η οποία θα μπορούσε να θεωρηθεί σαν σύνοψη του συνολικού διαγράμματος ακρίβειας/ανάκλησης. Συνήθως αυτή η τιμή είναι η ακρίβεια σε κάποιο συγκεκριμένο επίπεδο ανάκλησης. Φυσικά αυτές είναι λίγες από τις πολλές προσεγγίσεις που μπορούν να γίνουν.

3.3. Μοντέλα ανάκτησης πληροφορίας

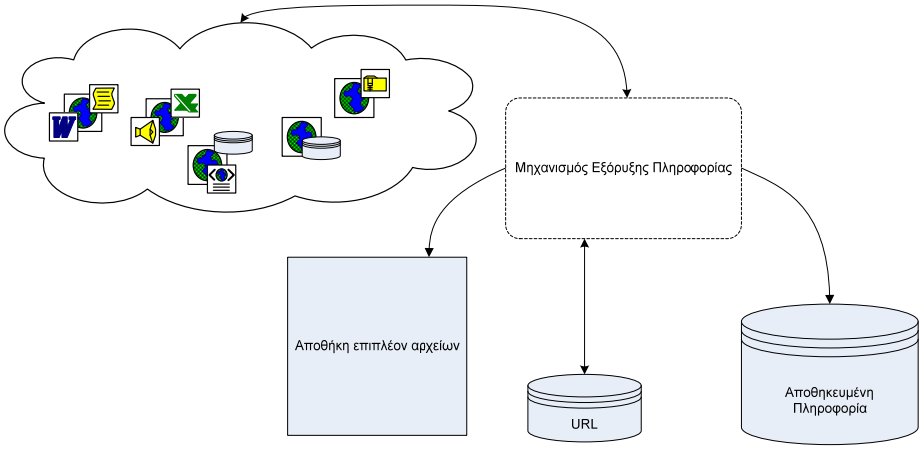
Τα τρία κλασσικά μοντέλα στην Ανάκτηση Πληροφορίας είναι το Boolean, το Vector Space και το Πιθανοτικό. Στο μοντέλο Boolean, τόσο τα κείμενα όσο και τα ερωτήματα αντιμετωπίζονται ως ένα σύνολο από όρους δεικτοδότησης. Κατά συνέπεια το μοντέλο μπορεί να θεωρηθεί ως συνολοθεωρητικό. Στο Vector Space, τα κείμενα και τα ερωτήματα αναπαρίστανται ως διανύσματα σε έναν t-διάστατο χώρο. Έτσι λέμε ότι το μοντέλο είναι αλγεβρικό. Το Πιθανοτικό μοντέλο εισάγει έναν τρόπο αναπαράστασης, ο οποίος βασίζεται στην πιθανοθεωρία. Κατά συνέπεια το μοντέλο είναι πιθανοτικού χαρακτήρα. Με τον καιρό προτάθηκαν διάφορες νέες προσεγγίσεις σε καθεμιά από τις κατηγορίες βασικών μοντέλων. Έτσι έχουμε στο συνολοθεωρητικό πεδίο τα μοντέλα, ασαφές (fuzzy) Boolean και επεκτεταμένο Boolean. Στα αλγεβρικά μοντέλα έχουμε το γενικευμένο vector space, την λανθάνουσα σημασιολογική δεικτοδότηση (LSI) και το μοντέλο των νευρωνικών δικτύων. Στον πιθανοτικό τομέα εμφανίστηκαν τα δίκτυα εξαγωγής συμπεράσματος (inference networks) και τα δίκτυα πεποίθησης (belief networks). Εκτός από την χρήση του περιεχομένου των κειμένων, ορισμένα μοντέλα εκμεταλλεύονται και την εσωτερική δομή που φυσιολογικά υπάρχει στο γραπτό λόγο. Σε αυτή την περίπτωση λέμε ότι έχουμε ένα δομημένο μοντέλο. Για τη δομημένη ανάκτηση κειμένου, συναντούμε δύο μοντέλα, τις μη επικαλυπτόμενες λίστες (non-overlapping lists) και τους κοντινούς κόμβους (proximal nodes).

3.3.1. Αρχιτεκτονική μηχανισμών εξόρυξης

Όλες οι μηχανές αναζήτησης πραγματοποιούν ανάκτηση πληροφορίας προκειμένου να μπορούν να εξυπηρετούν τους χρήστες τους. Έτσι, μέχρι σήμερα έχει κατασκευαστεί πληθώρα προγραμμάτων τα οποία είτε λειτουργώντας σαν αυτόνομες μονάδες είτε σε συνεργασία μεταξύ τους πραγματοποιούν εξόρυξη πληροφορίας. Η γενική ιδέα ενός μηχανισμού εξόρυξης πληροφορίας είναι εξαιρετικά απλή και φαίνεται στην Εικόνα 2.

Ένας τέτοιος μηχανισμός μπορεί να είναι ένας απλός υπολογιστής ή ακόμα και μερικές χιλιάδες υπολογιστές που λειτουργούν κάτω από την επίβλεψη ενός. Ο μηχανισμός ξεκινά να λειτουργεί περιδιαβαίνοντας σελίδες του Διαδικτύου. Οι HTML σελίδες αποθηκεύονται σε μία βάση δεδομένων μαζί με επιπρόσθετες πληροφορίες για αυτές οι οποίες μπορεί να περιλαμβάνουν: το URL, την ώρα που ανακτήθηκε η σελίδα, το μέγεθός της και άλλα. Σε μία ξεχωριστή (συνήθως) βάση δεδομένων αποθηκεύονται όλα τα URL που έχουν ανακτηθεί και τα οποία ανακτώνται ανά τακτά χρονικά διαστήματα. Παράλληλα κάθε σελίδα αναλύεται προκειμένου να εξαχθούν από αυτή όλα τα links που περιέχει (σύμβολο <a> στην HTML). Τα links που «διαβάζει» ο μηχανισμός συγκρίνονται με αυτά που υπάρχουν αποθηκευμένα στη βάση δεδομένων URL και γίνονται οι κατάλληλες προσθήκες. Τέλος, κάποια επιπλέον αρχεία (doc, css, xml, scripts, πολυμέσα) αποθηκεύονται

συνήθως σε καταλόγους που ονομάζονται κατάλληλα από τον μηχανισμό, έτσι ώστε να είναι σε θέση να τα προσπελάσει ανά πάσα στιγμή.



Εικόνα 2: Μηχανισμός Εξόρυξης Πληροφορίας

Μερικοί από τους πιο γνωστούς μηχανισμούς που πραγματοποιούν εξόρυξη πληροφορίας είναι οι crawlers, τα bots, τα spiders κ.α. Η λειτουργία τους είναι ουσιαστικά ίδια και βασίζεται στην αρχιτεκτονική που φαίνεται στο παραπάνω σχήμα.

3.3.2. Τεχνολογίες ανάκτησης δεδομένων από το Διαδίκτυο

Η ανάκτηση πληροφορίας είναι μία έννοια η οποία αναφέρεται σε κάθε μηχανισμό ο οποίος μέσω ενός αλγορίθμου «επιστρέφει» αποτελέσματα από ένα σύνολο στοιχείων. Μιλώντας για ανάκτηση πληροφορίας από το διαδίκτυο θα πρέπει να αναλογιστούμε τη μοναδικότητα των στοιχείων που χαρακτηρίζουν το Διαδίκτυο και συνεπώς αλλάζουν τη διαδικασία ανάκτησης δεδομένων από αυτό. Τα κύρια χαρακτηριστικά του Διαδικτύου είναι:

- Εξαιρετικά μεγάλο μέγεθος
- Δυναμικός χαρακτήρας
- Περιέχει ετερογενές υλικό
- Υπάρχει μεγάλο εύρος γλωσσών
- Διπλές εγγραφές
- Πολλά links από μία σελίδα σε άλλη
- Πολλοί και διαφορετικών ειδών χρήστες
- Διαφορετική συμπεριφορά από τους χρήστες

κλασσικά συστήματα ανάκτησης πληροφορίας οι μετρικές που χρησιμοποιούνται για την αξιολόγηση είναι:

- Η ανάκληση: Το ποσοστό των σελίδων που έχουν επιστραφεί και είναι σχετικές
- Η ακρίβεια: Το ποσοστό των σχετικών σελίδων που έχουν επιστραφεί
- Η ακρίβεια στα πρώτα 10 αποτελέσματα

Σε ένα σύστημα όμως που έχει να κάνει με ανάκτηση πληροφορίας από το διαδίκτυο θα πρέπει:

«Τα αποτελέσματα που επιστρέφονται θα πρέπει να έχει υψηλή σχετικότητα με το ερώτημα και αλλά και υψηλή ποιότητα, δηλαδή, με λίγα λόγια, θα πρέπει τα αποτελέσματα να είναι μόνο τα αναγκαία και απαραίτητα».

Αυτό σημαίνει πως σε ένα τέτοιο σύστημα θα πρέπει να χρησιμοποιηθούν διαφορετικές μετρικές με τη βοήθεια των οποίων θα είναι σε θέση οι μηχανισμοί ανάκτησης πληροφορίας να μπορούν να αξιολογήσουν τα ερωτήματα των

χρηστών και να επιστρέψουν τα πιο σωστά και πιο αντιπροσωπευτικά αποτελέσματα.

3.3.3. Εξόρυξη γνώσης και δεδομένων

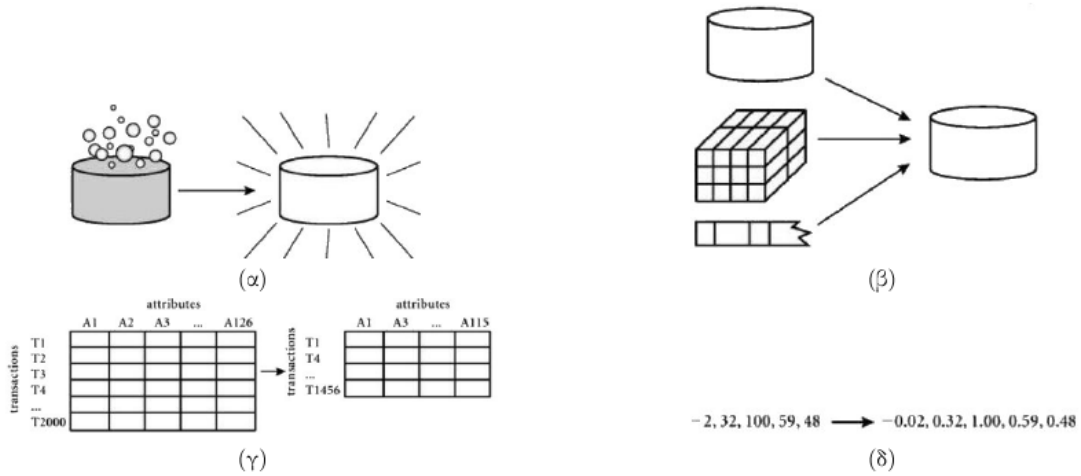
Η ανακάλυψη γνώσης από βάσεις δεδομένων, αναφέρεται στη διεργασία εξόρυξης γνώσης από τις μεγάλες αποθήκες δεδομένων οι οποίες συλλέγουν τα δεδομένα μέσα από την τεράστια κίνηση του παγκοσμίου ιστού. Ο όρος εξόρυξη δεδομένων χρησιμοποιείται ως συνώνυμο της ανακάλυψης γνώσης από βάσεις δεδομένων, καθώς επίσης και για αναφορά στις πραγματικές τεχνικές που χρησιμοποιούνται για την ανάλυση και την εξαγωγή της από διάφορα σύνολα δεδομένων. Πολλοί ερευνητές θεωρούν τον όρο εξόρυξη δεδομένων μη αντιπροσωπευτικό της διαδικασίας που περιγράφει, υποστηρίζοντας ότι ο όρος εξόρυξη γνώσης θα ήταν μια πιο κατάλληλη περιγραφή. Ο όρος εξόρυξη δεδομένων (Data Mining) είναι αυτός που έχει επικρατήσει και χαρακτηρίζει τη διαδικασία της εύρεσης δομών γνώσης οι οποίες περιγράφουν με ακρίβεια μεγάλα σύνολα πρωτογενών δεδομένων. Οι δομές αυτές αναδεικνύουν γνώση (συσχετίσεις ή κανόνες) που είναι κρυμμένοι μέσα στα δεδομένα και δεν μπορούν να εξαχθούν με «γυμνό»' μάτι. Οι προκύπτουσες δομές είναι πλούσιες σε σημασιολογία και εκμεταλλεύονται πιθανές κοινές ιδιότητες των πρωτογενών δεδομένων.

Οι δύο βασικοί στόχοι της εξόρυξης δεδομένων (γνώσης) είναι η εφαρμογή τεχνικών περιγραφής και πρόβλεψης σε μεγάλα σύνολα δεδομένων. Η πρόβλεψη στοχεύει στον υπολογισμό της μελλοντικής αξίας ή στην πρόβλεψη της συμπεριφοράς κάποιων μεταβλητών που παρουσιάζουν ενδιαφέρον (π.χ. το ενδιαφέρον ενός αναγνώστη για κείμενα διαφόρων κατηγοριών) και οι οποίες βασίζονται στη συμπεριφορά άλλων μεταβλητών. Η περιγραφή επικεντρώνεται στην ανακάλυψη προτύπων και αναπαριστά τα δεδομένα μιας πολύπλοκης βάσης δεδομένων με έναν κατανοητό και αξιοποιήσιμο τρόπο. Ως προς την εξόρυξη γνώσης, η περιγραφή τείνει να είναι περισσότερο σημαντική από την πρόβλεψη.

3.4. Προεπεξεργασία Δεδομένων

Τα δεδομένα που κατακλύζουν τις σύγχρονες βάσεις δεδομένων και τον παγκόσμιο ιστό σήμερα, είναι πολύ επιρρεπή σε θόρυβο, σε ανεπάρκεια ή συνοχή λόγω κυρίως του τεράστιου όγκου και της ετερογένειας τον πηγών τους. Δεδομένα χαμηλής ποιότητας οδηγούν σε χαμηλής ποιότητας εξόρυξη πληροφορίας. Το θεμελιώδες ερώτημα που τίθεται είναι: πώς μπορούν να προεπεξεργαστούν τα δεδομένα, ώστε να βελτιωθεί η ποιότητά τους και επομένως τα αποτελέσματα της εξόρυξης πληροφορίας.

Υπάρχει ένα πλήθος μεθόδων που χρησιμοποιούνται για την προεπεξεργασία δεδομένων. Το καθάρισμα δεδομένων μπορεί να έχει εφαρμογή στην αφαίρεση του θορύβου από τα δεδομένα και στην διόρθωση των ασυνεπειών σε αυτά. Η ολοκλήρωση των δεδομένων συνενώνει δεδομένα από διάφορες πηγές σε συναφή αποθήκη δεδομένων, όπως π. χ. μια βάση δεδομένων. Ο μετασχηματισμός των δεδομένων, όπως η κανονικοποίηση μπορεί να χρησιμοποιηθεί από τη διαδικασία προεπεξεργασίας δεδομένων. Για παράδειγμα, η κανονικοποίηση μπορεί να βελτιώσει την ακρίβεια και την αποτελεσματικότητα των αλγορίθμων εξόρυξης δεδομένων ενσωματώνοντας μετρικές απόστασης. Η αφαίρεση δεδομένων, μπορεί να μειώσει το μέγεθος των δεδομένων, συναθροίζοντας, απαλείφοντας τα πλεονάζοντα χαρακτηριστικά, ή ομαδοποιώντας τα δεδομένα. Αυτές οι τεχνικές δεν είναι αμοιβαία αποκλειόμενες· μπορούν να δουλέψουν μαζί. Για παράδειγμα, το καθάρισμα δεδομένων μπορεί να περιλαμβάνει μετασχηματισμούς για την διόρθωση λανθασμένων δεδομένων. Οι τεχνικές προεπεξεργασίας δεδομένων, όταν εφαρμόζονται πριν την εξόρυξη πληροφορίας, μπορούν να βελτιώσουν σημαντικά την ποιότητα της πληροφορίας που εξορύσσεται ή τον χρόνο που απαιτείται γι' αυτή τη διαδικασία.



Εικόνα 3: Τεχνικές προεπεξεργασίας δεδομένων (α)Καθαρισμός δεδομένων (β)Ολοκλήρωση δεδομένων (γ)Αφαίρεση δεδομένων (δ)Μετασχηματισμός δεδομένων

3.5. Αξιοποίηση Πληροφορίας

Η πληροφορία που ανακτάται τόσο από το μηχανισμό εξόρυξης όσο και από το μηχανισμό κατηγοριοποίησης είναι υπέρογκη. Αρκεί να φανταστεί κάποιος ότι από 100 τυχαίες ηλεκτρονικές διευθύνσεις εξάγονται 90-95 κείμενα, από τα οποία λαμβάνουμε 2000 διακριτές λέξεις (πριν τη διαδικασία του stemming) και από τις οποίες προκύπτουν 8000-10000 συσχετίσεις κείμενο-λέξη-βάρος. Για το λόγο αυτό θα πρέπει να υπάρχει ένας ισχυρός μηχανισμός που να είναι σε θέση να αξιοποιήσει τη συγκεκριμένη πληροφορία και να μπορεί να βελτιώσει τους τρόπους που γίνονται ερωτήματα στη βάση και προσθήκες νέων εγγραφών.

Αυτό που θα πρέπει να μας απασχολήσει περισσότερο για το συγκεκριμένο σύστημα είναι να δημιουργηθεί ένας μηχανισμός διαχείρισης της πληροφορίας. Η πληροφορία δε θα πρέπει να είναι στάσιμη. Συνεχώς θα ανανεώνεται, και θα πρέπει ανελλιπώς να διαγράφονται ή να τροποποιούνται τα στοιχεία τα οποία δε συγκεντρώνουν το ενδιαφέρον των χρηστών του συστήματος.

Προκειμένου να αξιοποιηθεί η πληροφορία θα πρέπει να δημιουργηθούν περιβάλλοντα διαχείρισης και μηχανισμοί ανάλυσης των ερωτημάτων και εύρεσης απάντησης. Παράλληλα θα πρέπει να υπάρχει τρόπος με τον οποίο να είναι εφικτή η ανάλυσης πληροφορίας από τις κινήσεις του χρήστη. Ωστόσο αυτό είναι ένα θέμα που θα καλυφθεί στην επόμενη ενότητα.

Τα συστήματα διαχείρισης πληροφορίας και ανάλυσης ερωτημάτων χρήστη, θα βασίζονται σε web interface προκειμένου να είναι άμεση η διασύνδεση με τον «πραγματικό» κόσμο. Είναι ουσιαστικά μία προσπάθεια να προσεγγίσουμε περισσότερα πραγματικά δεδομένα ξεφεύγοντας από την πληροφορία εκπαίδευσης. Επίσης, με αυτό τον τρόπο θα κάνουμε το μηχανισμό μας πιο διάφανο προς το χρήστη καθώς και πιο φιλικό.

Τα εργαλεία διαχείρισης δε θα περιέχουν πολύπλοκες συναρτήσεις, μα ούτε και πολύπλοκο περιβάλλον. Ο όγκος της πληροφορίας κάνει απαγορευτική την άμεση προσέγγισή της, συνεπώς ο διαχειριστής του συστήματος θα πρέπει να είναι σε θέση να έχει μια γενική εποπτεία του συστήματος διατηρώντας παράλληλα ανεκτά τα επίπεδα πρόσβασης σε εξειδικευμένα στοιχεία του συστήματος.

3.6. Προφίλ Χρήστη σε Δυναμικά Περιβάλλοντα

Ένα πολύ σημαντικό στοιχείο της εργασίας είναι το προφίλ χρήστη σε δυναμικό περιβάλλον. Είναι το στοιχείο που χαρακτηρίζει την πύλη ποιοτικού

περιεχομένου και είναι ένα από τα βασικά στοιχεία που δίνουν νόημα στη λέξη ποιότητα της πύλης.

Το δυναμικό περιβάλλον της πύλης θα δίνει τη δυνατότητα πρόσβασης σε πληροφορία η οποία ενδιαφέρει το χρήστη, καταργώντας τα περιθώρια εμφάνισης ανεπιθύμητων αποτελεσμάτων. Προκειμένου να γίνει κατανοητό θα πρέπει να προσδιοριστεί ο όρος προφίλ χρήστη.

Στο άκουσμα του όρου προφίλ χρήστη θα περίμενε κανείς να έρθει αντιμέτωπος με προσωπικά στοιχεία του χρήστη (όνομα, επώνυμο κλπ). Όσο κι αν ακούγεται παράξενο, σε ένα δυναμικό περιβάλλον ίσως δεν έχει και τόσο μεγάλη σημασία ο προσδιορισμός του χρήστη σαν φυσικό πρόσωπο αλλά περισσότερο σαν χρήστης του διαδικτύου. Βασικός στόχος της δημιουργίας του προφίλ ενός χρήστη είναι να προσδιοριστεί με όσο μεγαλύτερη ακρίβεια η δράση του φυσικού προσώπου όταν έρχεται αντιμέτωπος με το διαδίκτυο. Είναι μεγάλο επίτευγμα να μπορεί κανείς να προσδιορίσει την επόμενη κίνηση που θα πραγματοποιήσει ο χρήστης (π.χ. ποιο σύνδεσμο θα ακολουθήσει στην επόμενη κίνηση). Ακούγεται σαν παιχνίδι πρόβλεψης και ίσως θα μπορούσε να παρομοιαστεί με κάτι τέτοιο. Ωστόσο είναι κάτι πιο σύνθετο και βασίζεται σε μία πληθώρα στοιχείων. Τι ερωτήματα πραγματοποιεί ο χρήστης, ποιες σελίδες επισκέπτεται πιο συχνά από τα αποτελέσματα που του εμφανίζονται, τι έχει δηλώσει σαν «αγαπημένες κατηγορίες» αποτελούν μερικά από τα βασικά στοιχεία πάνω στα οποία βασίζεται η δημιουργία του προφίλ ενός χρήστη.

Στο συγκεκριμένο σύστημα, το ενδιαφέρον μας επικεντρώνεται στην αξιολόγηση που κάνει ο χρήστης όταν του παρουσιάζονται τα αποτελέσματα της αναζήτησής του. Ένα παράδειγμα θα ήταν αρκετό για να κατανοήσει κανείς το νόημα που έχει το «δυναμικό προφίλ» στη συγκεκριμένη δικτυακή πύλη. Έστω ένας χρήστης του διαδικτύου που χρησιμοποιεί τη συγκεκριμένη δικτυακή πύλη και επιθυμεί να βλέπει καθημερινά τα περιεχόμενα της κατηγορίας business. Το προφίλ έχει ήδη δημιουργηθεί και περιλαμβάνει την πολύ γενική κατηγορία business. Όταν παρουσιάζονται στο χρήστη αποτελέσματα (τίτλος άρθρου, μικρό απόσπασμα άρθρου), τότε ο χρήστης επιλέγει κάποιο ή κάποια αποτελέσματα για να τα εξετάσει περαιτέρω. Το κάθε κείμενο όμως αποτελείται, συν τοις άλλοις, και από κάποιες λέξεις-κλειδιά. Μόλις κάποιος χρήστης επιλέξει κάποιο κείμενο, οι λέξεις-κλειδιά που υπάρχουν στο συγκεκριμένο, αυτομάτως αποκτούν αξία για το συγκεκριμένο χρήστη και εισάγονται αυτόματα στο προφίλ του. Αυτή η πληροφορία είναι πολύ σημαντική προκειμένου το σύστημα να είναι σε θέση να κάνει μεγαλύτερη αξιολόγηση των κειμένων που θα παρουσιάσει στο χρήστη. Έτσι, την επόμενη φορά που ο χρήστης θα δει τα αποτελέσματα για την κατηγορία που επιθυμεί τα κείμενα θα είναι ταξινομημένα βάσει των λέξεων-κλειδιών που έχουν τη μεγαλύτερη βαθμολογία για κάθε χρήστη. Με αυτό τον τρόπο αποκτά μεγαλύτερη αξία το κείμενο που περιέχει πολλές λέξεις-κλειδιά για ένα συγκεκριμένο χρήστη. Η συγκέντρωση των αποτελεσμάτων συνολικά για τους χρήστες μίας κατηγορίας μπορεί να οδηγήσει σε μεγαλύτερη διαβάθμιση κάθε κατηγορίας και δημιουργία εικονικών υποκατηγοριών που θα είναι χωρισμένες βάση της απόκρισης των χρηστών. Θεωρητικά ένα τέτοιο μοντέλο, εικονικής ουσιαστικά, κατηγοριοποίησης είναι πιο αποτελεσματικό από κάθε αλγοριθμικό μοντέλο καθώς η κατηγοριοποίηση δε γίνεται από τη μηχανή αλλά από τον άνθρωπο.

3.7. Προσωποποιημένη Αναζήτηση

Η διαδικασία αναζήτησης είναι από τους πιο άμεσους τρόπους επικοινωνίας του χρήστη με το σύστημα μας. Είναι αναγκαίο να προσφέρεται η δυνατότητα στο χρήστη να μπορεί να εντοπίσει την πληροφορία που τον ενδιαφέρει χωρίς να χρειαστεί να καταβληθεί πολύς κόπος. Επίσης, είναι πολύ σημαντικό να μειωθεί στο ελάχιστο ο αριθμός των άρθρων του αποτελέσματος της αναζήτησης που είναι άσχετα με την αρχική επερώτηση του χρήστη. Στη διαδικασία αναζήτησης,

δίνονται στο χρήστη ορισμένες προηγμένες δυνατότητες. Εκτός από τις λέξεις - κλειδιά της επερώτησης, ο χρήστης μπορεί να προσδιορίσει τα ακόλουθα:

- Θεματική Ενότητα των άρθρων του αποτελέσματος: Για κάθε άρθρο υπολογίζεται η σχετικότητα του με κάποια από τις θεματικές ενότητες του συστήματος. Έτσι, αν ο χρήστης έχει επιλέξει μια συγκεκριμένη θεματική ενότητα, τότε στα άρθρα του αποτελέσματος το σύστημα θα δώσει προβάδισμα σε άρθρα που ανήκουν στην συγκεκριμένη κατηγορία.
- Χρονικό Διάστημα: σε μορφή "από - εώς" ο χρήστης μπορεί να προσδιορίσει το χρονικό διάστημα δημοσίευσης των άρθρων του αποτελέσματος.
- Λογική σύνδεση των κωδικολέξεων: εδώ υλοποιείται η λογική OR και AND για τις λέξεις - κλειδιά της επερώτησης

Με βάση το προφίλ που έχει διαμορφωθεί για τον χρήστη, είναι δυνατό να τροποποιηθεί με τρόπο διαφανή η επερώτηση που υπέβαλλε αυτός και να εμπλουτιστεί με περισσότερες λέξεις - κλειδιά ή εκφράσεις. Ως εκ τούτου, είναι δυνατό να ενταχθούν στα άρθρα του αποτελέσματος άρθρα τα οποία δεν θα υπήρχαν αν χρησιμοποιούταν η αρχική επερώτηση. Στο τελικό αποτέλεσμα τα άρθρα αναδιατάσσονται ανάλογα με τη σχετικότητα τους στην εμπλουτισμένη επερώτηση του χρήστη.

Ταυτόχρονα για την επιτάχυνση της διαδικασίας της αναζήτησης, χρησιμοποιείται ένα μηχανισμός caching των αποτελεσμάτων από προηγούμενες συνεδρίες του χρήστη στο σύστημα. Κάθε φορά που ο χρήστης υποβάλλει μια επερώτηση, ελέγχεται αν στο πρόσφατο παρελθόν ο συγκεκριμένος χρήστης έχει υποβάλλει παρόμοια ή ακριβώς την ίδια επερώτηση. Ο μηχανισμός φροντίζει ώστε τα αποτελέσματα να διατηρούνται όσο το δυνατόν περισσότερο "φρέσκα" (να μην χρησιμοποιούνται πολύ παλιά cached αποτελέσματα) και επιπλέον το μέγεθος της μνήμης cache στον εξηπηρετητή να διατηρείται μικρό.

4

ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Στο κεφάλαιο αυτό περιγράφονται σχετικές εργασίες με κάθε υποσύστημα της συγκεκριμένης εργασίας. Οι σχετικές εργασίες περιλαμβάνουν:

- Προεπεξεργασία Δεδομένων
- Αναζήτηση Προσωποποιημένη στο Χρήστη
- Caching των αποτελεσμάτων

4. ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Η αναζήτηση για σχετικές εργασίες μας φέρνει αντιμέτωπους με μία σειρά από συστήματα που έχουν αναπτυχθεί προκειμένου να διευκολύνουν τους χρήστες κατά την προσπάθεια αναζήτησης για πληροφορίες που αφορούν άρθρα. Τα συστήματα αυτά έχουν, το καθένα, ένα διαφορετικό τρόπο προσέγγισης του θέματος. Σε άλλα σημεία συγκλίνουν και σε άλλα αποκλίνουν ενώ η «γενική ιδέα» εντοπίζεται κυρίως στους μηχανισμούς κατηγοριοποίησης και προσωποποίησης στο χρήστη.

Ελπιδοφόρα είναι τα μηνύματα που έρχονται από το δικτυακό τόπο Google [111] όπου μία νέα υπηρεσία ειδήσεων έχει ήδη αρχίσει να προβάλλεται από τη Google και να χρησιμοποιείται από ολοένα και περισσότερους χρήστες. Σε αυτή την περίπτωση οι χρήστες βλέπουν σε βασικές κατηγορίες τα νέα και έχουν τη δυνατότητα προσωποποίησης των νέων που τους παρουσιάζονται. Σημαντικό είναι το γεγονός πως η υπηρεσία παρέχεται στη γλώσσα που επιθυμεί ο χρήστης με αποτέλεσμα να είναι σε θέση να αναγνώσει όλες τις τελευταίες ειδήσεις που έχουν συλλεχθεί από μεγάλα ειδησεογραφικά πρακτορεία.

4.1. Προεπεξεργασία δεδομένων

Στη θεωρία, τα βασισμένα σε κείμενο χαρακτηριστικά ενός εγγράφου μπορούν να περιλαμβάνουν κάθε λέξη / φράση η οποία μπορεί να εμφανίζεται σε ένα δεδομένο σύνολο κειμένων. Όμως, επειδή κάτι τέτοιο είναι υπολογιστικά μη-ρεαλιστικό, χρειαζόμαστε κάποια μέθοδο προεπεξεργασίας κειμένων για την αναγνώριση των λέξεων - κλειδιών (κωδικολέξεων ή αλλιώς keywords) και φράσεων οι οποίες μπορεί να μας είναι χρήσιμες. Διάφορες τεχνικές έχουν προταθεί για την αναγνώριση των keywords ενός κειμένου όπως τα Hidden Markov Models [41], η Naive Bayes [43] και τα Support Vector Machines [42] όμως όλες αυτές οι μέθοδοι τείνουν να κάνουν χρήση συγκεκριμένης γνώσης μετα-πληροφορίας για τη γλώσσα του κειμένου. Άλλες μέθοδοι χρησιμοποιούν στατιστικές πληροφορίες, όπως η συχνότητα μιας λέξης. Μια ευρέως γνωστή τεχνική είναι η TF-IDF (Term Frequency - Inverse Document Frequency), όπου TF είναι το πλήθος των εμφανίσεων ενός όρου σε ένα δεδομένο σύνολο κείμενων συγκρινόμενο με το πλήθος των κειμένων που περιέχουν το συγκεκριμένο όρο, και IDF είναι ένα μέτρο των συνολικών κειμένων σε μια συλλογή κειμένων, συγκρινόμενο με το συνολικό αριθμό κειμένων που περιέχουν μια δεδομένη λέξη [44]. Σχετικές τεχνικές, οι οποίες περιλαμβάνουν άλλες στατιστικές που πηγάζουν από το σύνολο των κειμένων, έχουν επίσης προταθεί τα πρόσφατα χρόνια· π.χ. κέρδος πληροφορίας [45], odds ratio [46], CORI [47], κλπ. Οι τεχνικές αυτές προσφέρουν μια βελτιωμένη προσέγγιση.

4.1.1. Ανάλυση

Στην ανάκτηση πληροφορίας, η σχέση μεταξύ ενός ερωτήματος χρήστη και ενός κειμένου καθορίζεται κυρίως από το πλήθος των όρων που έχουν κοινούς. Δυστυχώς, οι λέξεις έχουν πολλές μορφολογικές παραλλαγές οι οποίες δεν αναγνωρίζονται από αλγόριθμους που βασίζονται στο ταίριασμα όρων χωρίς να προηγηθεί κάποιας μορφής επεξεργασία φυσικής γλώσσας (Natural Language Processing). Στις περισσότερες των περιπτώσεων, αυτές οι παραλλαγές έχουν παρόμοιες εννοιολογικές ερμηνείες και μπορούν να αντιμετωπισθούν ως ισοδύναμες στα πλαίσια εφαρμογών ανάκτησης πληροφορίας (σε αντίθεση με τις γλωσσολογικές). Ως εκ' τούτου, ένα πλήθος αλγορίθμων κατάλληλων για τη διαδικασία του stemming έχουν αναπτυχθεί ώστε να περιορίσουν τις μορφολογικές παραλλαγές στην αρχική τους ρίζα.

Το πρόβλημα του stemming έχει προσεγγιστεί από μια μεγάλη ποικιλία μεθόδων που περιγράφονται στο [48] και περιλαμβάνουν αφαίρεση της κατάληξης, τμηματοποίηση λέξης και λεξιλογική μορφοποίηση. Δύο από τους διασημότερους αλγορίθμους, ο Lovins [49] και ο Porter [50], βασίζονται στην αφαίρεση της κατάληξης. Ο αλγόριθμος Lovins βρίσκει το μακρύτερο ταίριασμα από μια μεγάλη λίστα καταλήξεων, ενώ ο Porter χρησιμοποιεί έναν επαναληπτικό αλγόριθμο με μικρότερο αριθμό καταλήξεων και μερικούς κανόνες. Ένας ακόμη αλγόριθμος, ο Paice/Husk [51], χρησιμοποιεί αποκλειστικά ένα σύνολο κανόνων ενώ ακολουθεί επαναληπτική προσέγγιση.

Στο [52] περιγράφονται τα προβλήματα που σχετίζονται με αυτές τις προσεγγίσεις. Οι περισσότεροι stemmers λειτουργούν χωρίς λεξικό και επομένως αγνοούν το νόημα των λέξεων, κάτι που οδηγεί σε ορισμένα λάθη κατά τη διαδικασία του stemming. Λέξεις διφορούμενες μειώνονται στην ίδια ρίζα και λέξεις με παρόμοιο νόημα δεν μειώνονται στην ίδια ρίζα. Για παράδειγμα, ο Porter stemmer μειώνει τις λέξεις general, generous, generation, generic στην ίδια ρίζα.

Παράλληλα, η έξοδος (stems) που παράγεται από τους αλγορίθμους, συνήθως δεν περιέχει πραγματικές λέξεις, κάτι που την κάνει δύσχρηστη για εργασίες που έχουν να κάνουν με ανάκτηση πληροφορίας. Διαδραστικές τεχνικές οι οποίες απαιτούν είσοδο από τον χρήστη απαιτούν από αυτόν την εργασία με stems και όχι πραγματικών λέξεων. Προβλήματα αυτού του τύπου αντιμετωπίζονται προσεγγίζοντας τη διαδικασία με μορφολογική ανάλυση.

Υπάρχει ένας μεγάλος αριθμός εργασιών που έχουν εξετάσει τον αντίκτυπο των stemming αλγορίθμων στην απόδοση της ανάκτησης πληροφορίας. Στο [53] δίνεται μια καλή περίληψη, αναφέροντας ότι τα συνδυασμένα αποτελέσματα των προηγούμενων μελετών καθιστούν ασαφές εάν η διαδικασία του stemming είναι χρήσιμη. Στις περιπτώσεις όπου το stemming είναι χρήσιμο τείνει να ασκήσει μόνο μικρή επίδραση στην απόδοση, και η επιλογή του stemmer μεταξύ των πιο κοινών παραλλαγών δεν είναι σημαντική. Εντούτοις, δεν υπάρχει κανένα στοιχείο ότι ένα λογικός stemmer μπορεί να βλάψει την απόδοση της ανάκτησης πληροφορίας.

Αντίθετα, μια πρόσφατη μελέτη [54] εντοπίζει μια αύξηση 15-35% στην απόδοση ανάκτησης όταν το stemming χρησιμοποιείται σε μερικές συλλογές (CACM και npl). Αναφέρεται ότι αυτές οι συλλογές έχουν και ερωτήματα και έγγραφα τα οποία είναι εξαιρετικά σύντομα. Για συλλογές με μεγαλύτερα κείμενα, οι stemming αλγόριθμοι χαρακτηρίζονται από μια σχετική αύξηση στην απόδοση της διαδικασίας ανάκτησης πληροφορίας.

4.2. Αναζήτηση Προσωποποιημένη στο Χρήστη

Σύμφωνα με τον Mobasher [19], «η προσωποποίηση στο διαδίκτυο μπορεί να περιγραφή σαν κάθε ενέργεια που σαν σκοπό έχει να κάνει τη Διαδικτυακή εμπειρία ενός χρήστη να είναι βάσει των αναγκών που έχει κάθε χρήστης». Σε γενικές γραμμές αυτό σημαίνει αλλαγή της παρουσίασης των δεδομένων ενός Δικτυακού τόπου προς το χρήστη σύμφωνα με τις εκάστοτε ρητές και εννοούμενες επιλογές του χρήστη. Αυτό είναι σχετικά εύκολο όταν αναφέρομαστε σε ένα και μόνον δικτυακό τόπο. Ο χρήστης καλείται να δηλώσει ρητά τις προτιμήσεις του ενώ παράλληλα το σύστημα «μαθαίνει» τις προτιμήσεις του χρήστη. Αυτό συναντάται σε πολλούς δικτυακούς τόπους.

Ο έλεγχος της δραστηριότητας του χρήστη σε πολλαπλούς δικτυακούς τόπους και ο εντοπισμός των πραγματικών αναγκών του και επιλογών είναι μία μεγάλη πρόκληση. Αυτό συνεπάγεται πως τη στιγμή που ένας χρήστης επισκέπτεται ένα δικτυακό τόπο, υπάρχει ήδη ένα προφίλ του και το σύστημα είναι άμεσα σε θέση να προσαρμοστεί στις ανάγκες του συγκεκριμένου χρήστη. Πολλές προσεγγίσεις πάνω στο συγκεκριμένο θέμα έχουν δοκιμαστεί: Single Sign On συστήματα **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** [21], προσωποποίηση στη μεριά του χρήστη [20] και βέβαια όλα τα συστήματα spyware και ad trackers. Πολλά από αυτά τα συστήματα παρουσιάζουν προβλήματα με τη

νομοθεσία καθώς προσβάλλουν την ιδιωτικότητα του χρήστη ενώ τα συστήματα που εφαρμόζουν την προσωποποίηση στη μεριά του χρήστη έχουν χαμηλή αποδοτικότητα.

Μία σειρά από πρωτοβουλίες στην W3C έχουν σαν σκοπό την καθολική προσωποποίηση. Το OPS (Open Profiling Standard) [22] είναι ένα προτεινόμενο W3C standard το οποίο έχει υποβληθεί από τις εταιρίες Netscape, Verisign και Firefly από το 1997. Παρουσιάζει ένα σχήμα τυποποίησης και ένα πρωτόκολλο ανταλλαγής δεδομένων που αφορούν το προφίλ ενός χρήστη, όπως για παράδειγμα το όνομα, τη διεύθυνση και τον ταχυδρομικό κώδικα. Ωστόσο, δεν τέθηκε ποτέ σε χρήση. Η ιδέα ανταλλαγής πληροφορίας είναι πολύ χρήσιμη, όμως πολλοί χρήστες δε θα επιθυμούσαν τη δημοσιοποίηση τέτοιων στοιχείων. Για την προσωποποίηση θα ήταν χρησιμότερο να διαμοιράζονται πληροφορίες που αφορούν την περιαγωγή ενός χρήστη στους δικτυακούς τόπους.

Το PIDL (Personalized Information Description Language) [23] είναι ένα πρωτόκολλο που υποβλήθηκε στην W3C από την εταιρία NEC το 1999. Πρόκειται για έναν τρόπο δόμησης εγγράφου που περιέχει στοιχεία για τις προτιμήσεις ενός χρήστη κατά τη διάρκεια που βρίσκεται σε διάφορους δικτυακούς τόπους. Είναι προφανές πως κάτι τέτοιο έρχεται ενάντια στα στοιχεία ιδιωτικότητας του χρήστη που έχουμε ήδη αναφέρει. Είχε προταθεί αρχικά για χρήση σε multicast, μία τεχνολογία που τελικά δεν αναπτύχθηκε όσο αναμενόταν.

Το CC/PP (Composite Capabilities/Preference Profiles) [24] είναι ένα W3C στάνταρ που προτάθηκε το 1999 και βρίσκεται μέχρι και σήμερα σε χρήση. Επιτρέπει σε κινητούς χρήστες να εκφράσουν τις προτιμήσεις ενός χρήστη σε έναν κεντρικοποιημένο εξυπηρετητή. Παρά το γεγονός ότι οι κινητές τεχνολογίες έχουν πολλούς περιορισμούς στην ανταλλαγή δεδομένων, αυτή η αρχιτεκτονική θα μπορούσε να αποτελέσει τη βάση για ένα σύστημα διαμοιρασμού των προτιμήσεων ενός χρήστη.

Το P3P (Platform for Privacy Preferences) [25] έρχεται σε αντίθεση με κάθε σύστημα προσωποποίησης που βασίζεται στο διαμοιρασμό των στοιχείων ενός χρήστη μεταξύ δικτυακών τόπων. Αυτή η σύσταση της W3C που έγινε το 2002 έχει σχεδιαστεί ώστε να επιτρέπει στους χρήστες να ελέγχουν τα προσωπικά τους δεδομένα που θα παρουσιάζονται στους διάφορους δικτυακούς τόπους που επισκέπτεται.

Κανένα από τα παραπάνω δεν επιτρέπει την προσωποποίηση σε πολλαπλούς δικτυακούς τόπους. Αν αναλογιστούμε τα εμπορικά συστήματα θα δούμε πως πρόκειται για ένα σημαντικό κομμάτι τους, κυρίως όσον αφορά θέματα μάρκετινγκ. Οι εταιρίες επιθυμούν να γνωρίζουν τις ανάγκες των «πελατών» τους πρωτού αυτοί επισκευθούν το «κατάστημά» τους. Έτσι, πολλοί δικτυακοί τόποι, όπως για παράδειγμα η προσωποποίηση και οι συστάσεις που παρουσιάζονται στο δικτυακό τόπο του Amazon.com **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** το εφαρμόζουν σε ατομικό επίπεδο. Από τις πρώτες κιόλας σελίδες που επισκέπτεται ο χρήστης διαμορφώνεται ένα προφίλ του προκειμένου ο δικτυακός τόπος να προσαρμόζεται σιγά - σιγά στις ανάγκες του.

Η μελέτη του θέματος που αφορά τις επιλογές ενός χρήστη καθώς και τη συμπεριφοράς αυτού κατά την επίσκεψη πολλών διαφορετικών δικτυακών τόπων έχει πραγματοποιηθεί από πολλές εταιρίες και έχουν γίνει πολλές προτάσεις. Αν εξαιρέσουμε τις προσπάθειες στις οποίες ανακύπτουν ηθικά αλλά και νομικά ζητήματα παραβίασης της ιδιωτικότητας καταλήγουμε αποκλειστικά στα συστήματα SSO (Single Sign On) όπως είναι το Microsoft Passport **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** και το Liberty Alliance [26]. Αυτά παρέχουν μία ενιαία βάση δεδομένων που περιέχει τα προσωπικά στοιχεία και τις επιλογές του. Οι χρήστες προσθέτουν από μόνοι τους στοιχεία στη βάση δεδομένων στα οποία έχουν ελεύθερη πρόσβαση εταιρίες που είναι συμβεβλημένες με τα εκάστοτε SSO συστήματα.

Βασικό πρόβλημα αυτής της προσέγγισης είναι η εξασφάλιση της ασφάλειας του συστήματος καθότι ο χρήστης μπορεί να αποθηκεύει ευαίσθητα δεδομένα. Το συγκεκριμένο θέμα τονίζεται ακόμα και στα προϊόντα των εταιριών (για παράδειγμα η Sun το τονίζει ιδιαίτερα στο πρόγραμμα Liberty [27]). Πως θα εμπιστευτεί ένας χρήστης το πρόγραμμα το οποίο του τονίζει ιδιαίτερα πως δεν είναι ασφαλές; Τα νεότερα SSO συστήματα όπως το Liberty Alliance [26] και το SXIP [[28] έχουν δώσει ιδιαίτερη προσοχή στο συγκεκριμένα θέμα προκειμένου να βελτιωθούν. Μάλιστα το SIXP επιτρέπει σε ένα χρήστη να διαθέτει πολλαπλά προφίλ ανάλογα με το μέγεθος των δεδομένων που επιθυμεί να είναι ορατά σε διάφορους δικτυακούς τόπους ορίζοντας με αυτό τον τρόπο αυτόνομα το επίπεδο ασφάλειας. Παράλληλα είναι ένα σύστημα ανοιχτού κώδικα προκειμένου οι χρήστες να μπορούν να δουν επακριβώς τι στοιχεία τους διαμοιράζονται και με ποιον τρόπο. Αυτό βέβαια δεν ξεπερνά τα προβλήματα που παρουσιάζονται. Οι χρήστες πρέπει να αποφασίσουν αν οι εταιρίες στις οποίες θα εμπιστευτούν τα προσωπικά τους δεδομένα είναι έμπιστες ή όχι. Αυτό συνεπάγεται και την αποτυχία τετοιων συστημάτων με χαρακτηριστικό παράδειγμα το σύστημα Passport σαν τεχνολογία καθότι οι χρήστες δεν έχουν κάποια ιδιαίτερη προτίμηση στα SSO συστήματα. Παράλληλα, όπως αναφέρει και ο Gartner **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, «όσο οι χρήστες δε δείχνουν να αποδέχονται τέτοια συστήματα οι εταιρίες δεν πρόκειται να κάνουν απολύτως καμία επένδυση».

Υπάρχουν βέβαια και συστήματα τα οποία δεν απαιτούν την εισαγωγή στοιχείων από το χρήστη αλλά χρησιμοποιούν μεταδεδομένα που υπάρχουν από τα ίχνη που αφήνει ένας χρήστης καθώς πραγματοποιεί περιαγωγή σε σελίδες του διαδικτύου. Το WAWA (Wisconsin Adaptive Web Assistant) [29] είναι ένα σύστημα το οποίο προσπαθεί να εντοπίσει τις σελίδες που μπορεί να αφορούν κάποιο χρήστη ανάλογα με το history που εντοπίζει στο φυλλομετρητή. Αντίστοιχα το Syskill and Webert [32] είναι ένα πρόγραμμα το οποίο μαθαίνει να βαθμολογεί τις σελίδες που επισκέπτεται ο χρήστης και αποφασίζει ποιες είναι οι σελίδες που πιθανόν ενδιαφέρουν το χρήστη. Το σύστημα αυτό χρησιμοποιεί το προφίλ χρήστη που το ίδιο κατασκευάζει και προτείνει στο χρήστη συνδέσμους που ενδεχόμενα τον ενδιαφέρουν το χρήστη ή πραγματοποιεί ερωτήματα σε μηχανές αναζήτησης με λέξεις κλειδιά από το διαμορφωμένο προφίλ χρήστη. Ο Chan [31] περιγράφει ένα παραπλήσιο σύστημα το οποίο περιέχει δύο στοιχεία: το Web Access Graph (WAG) και τον Page Interest Estimator (PIE). Το WAG εντοπίζει ίχνη σε ιστοσελίδες που μπορεί να αφορούν το χρήστη και το PIE «μαθαίνει» τον τρόπο με τον οποίο επισκέπτεται ένας χρήστης μία σελίδα βάσει των επιλογών που κάνει.

Οι Widyantoro, Ioerger και Yen [30] ανέπτυξαν ένα σύστημα το οποίο βασίζεται σε έναν τριπλό περιγραφέα προκειμένου να καταγράφουν τη δυναμική ενός χρήστη απέναντι στο διαδίκτυο. Το μοντέλο αυτό διατηρεί μία μία περιγραφή για κάθε ίχνος που αφήνει ο χρήστης στο διαδίκτυο σε ένα μεγάλο βάθος χρόνου και το συνδυάζει με δεδομένα που αποθηκεύονται προσωρινά προκειμένου να κάνει προβλέψεις για τις ιστοσελίδες που μπορεί να αφορούν το χρήστη.

Οι Goecks και Shavlik [33] προτείνουν ένα σύστημα που «μαθαίνει» τα ενδιαφέροντα του χρήστη ελέγχοντας περισσότερα στοιχεία που αφορούν τις σελίδες που επισκέπτεται. Παρατηρούν για παράδειγμα τις κινήσεις που κάνει ο χρήστης με το ποντίκι εκτός από την απλή διαδικασία ελέγχου των σελίδων που επισκέπτεται ο χρήστης.

Στον τομέα της προσωποιημένης αναζήτησης, μπορούμε να ταυτοποιήσουμε δύο διαφορετικές προσεγγίσεις. Η επέκταση των επερωτήσεων και η επεξεργασία των αποτελεσμάτων αλληλοσυμπληρώνονται και, λαμβάνοντας υπόψιν τόσο το χρήστη όσο και το ίδιο το περιεχόμενο, είναι δυνατόν να κερδίσουμε σε αποτελεσματικότητα στην αναζήτηση [97]. Η επέκταση της επερώτησης είναι ένας τρόπος για να αντιμετωπίσουμε το πρόβλημα της ακαταλληλότητας ορισμένων λέξεων-κλειδιών μέσα στην επερώτηση, το οποίο δημιουργείται όταν οι χρήστες χρησιμοποιούν διαφορετικούς όρους για να περιγράψουν μια έννοια από αυτούς

που χρησιμοποιούν οι ίδιοι οι συγγραφείς των άρθρων που επιθυμούν να ανακτήσουν. Αυτό επιτυγχάνεται με τον εμπλουτισμό της αρχικής επερώτησης με πιο σχετικές λέξεις οι φράσεις, πράγμα το οποίο όπως αποδεικνύεται αυξάνει την αποτελεσματικότητα της αναζήτησης [98]. Το περιεχόμενο της αναζήτησης μπορεί να λάβει διαφορετικές μορφές, όπως θεματική ενότητα που επιλέγεται από τον ίδιο το χρήστη [99] ή συνδυασμό από τίτλο και περιγραφή των επιλεγμένων αποτελεσμάτων αφού έχει υποβληθεί η αρχική επερώτηση [100]. Από την άλλη μεριά, η επεξεργασία του αποτελέσματος, περιλαμβάνει το φιλτράρισμα και την αναδιοργάνωση των άρθρων του αποτελέσματος με σκοπό να παραχθεί ένα πιο καλά προσδιορισμένο και ποιοτικό αποτέλεσμα σε ότι έχει να κάνει με την κάλυψη των αναγκών του χρήστη. Το φιλτράρισμα μπορεί να εκτελεστεί είτε στο δικτυακό τόπο στον οποίο ανήκουν τα επιστρεφόμενα αποτελεσμάτα [101], απομακρύνοντας κατ' αυτόν τον τρόπο άρθρα που ανήκουν σε συγκεκριμένους δικτυακούς τόπους, ή στα αρθρα αυτά καθαυτά που μπορεί να είναι ή να μην είναι κοντα στο ενδιαφέρον του χρήστη. Αυτό το τελευταίο καθορίζεται από άμεσα (rankings) ή έμμεσα (χρονική διάρκεια παραμονής ενός χρήστη σε κάποιο άρθρο καθώς και σειρά με την οπόια επισκέφθηκε τα άρθρα. Μια άλλη προσέγγιση για την επεξεργασία των άρθρων του αποτελέσματος είναι η αναδιοργάνωση και ο επαναπροσδιορισμός της σειράς με την οποία θα εμφανιστούν τα άρθρα στο τελικό αποτέλεσμα. Αυτό μπορεί να συμπεριλάβει και τη δημιουργία δυναμικών προφίλ για τους χρήστες του συστήματος με την πάροδο του χρόνου εκμεταλλευόμενοι στοιχεία όπως υποβληθείσες επερωτήσεις και σύνδεσμοι που ο χρήστης επισκέφθηκε [102]. Επερωτήσεις που υποβλήθηκαν στο παρελθόν και αναδρομές στο ιστορικό των άρθρων που επισκέφθηκε ο χρήστης χρησιμοποιούνται για την αλλαγή της σειράς των άρθρων του αποτελέσματος όπως στο [103]. Αυτές οι τεχνικές που περιγράφτηκαν μπορούν να χρησιμοποιηθούν στον εξυπηρετητή (server-side), ωστόσο έχουν προταθεί στο παρελθόν και τεχνικές και προσεγγίσεις οι οποίες εκτελούνται στο χώρου του πελάτη (client-side) όπως στο [104], όπου η επέκταση της επερώτησης και ο επαναπροσδιορισμός της σειράς των αποτελεσμάτων βασίζονται στην αμέσως προήγουμενη (ή στις προηγούμενες επερωτήσεις) καθώς και στο ιστορικό των άρθρων που ο χρήστης επέλεξε να επισκεφτεί και να διαβάσει.

4.3. Caching των αποτελεσμάτων

Σε ότι αφορά προηγούμενες εργασίες στον τομέα του caching αποτελεσμάτων σε μηχανές αναζήτησης, ο Markatos [105]. Τα αναφερόμενα αποτελέσματα δείχνουν ότι υπάρχουν σημαντικά κέρδη σε αποτελεσματικότητα χρησιμοποιώντας τεχνικές caching πράγμα το οποίο δικαιολογείται μιας και υπάρχει αρκετά μεγάλη τοπικότητα στις λέξεις-κλειδιά που χρησιμοποιούνται στις επερωτήσεις που υποβάλλονται από τους χρήστες. Στο [106], οι Xie και O'Hallaron διαπίστωσαν την ύπαρξη κατανομής Zipf στις συχνότητες επερωτήσεων, όπου διαφορετικοί χρήστες υποβάλλουν παρόμοιες επερωτήσεις ενώ επερωτήσεις μεγαλύτερες σε μήκος (μεγαλύτερος αριθμός λέξεων-κλειδιών) δεν μοιράζονται από αρκετούς χρήστες. Για την διαχείριση της μνήμης cache, οι Lempel και Moran [107] προτείνουν μια τεχνική που εξετάζει τις πιθανότητες στις κατανομές συχνοτήτων επερωτήσεων των χρηστών μιας μηχανής αναζήτησης. Οι Fagni et al [108] περιγράφουν μια Static Dynamic Cache, όπου μέρος της μνήμης cache είναι read-only ή στατικό και αποτελείται από ένα σύνολο συχνών επερωτήσεων από ένα παρελθοντικό ιστορικό επερωτήσεων. Το δυναμικό κομμάτι χρησιμοποιείται για το caching επερωτήσεων που δεν υπάρχουν στο στατικό κομμάτι. Σχετικά με την τοπικότητα των λέξεων-κλειδιών στις επερωτήσεις, σε προγενέστερες εργασίες, οι Jansen kai Spink [109] παρέχουν στοιχεία για βραχυπρόθεσμες αλληλεπιδράσεις χρηστών με μηχανές αναζήτησης και διαπιστώνουν ότι υπάρχει ένα αρκετά μεγάλο ποσοστό τοπικότητας στις υποβαλλόμενες επερωτήσεις. Οι Teevan et al. [110] εξέτασαν τη συμπεριφορά παραπάνω των 1000 χρηστών σε ότι αφορά αναζητήσεις σε μηχανές αναζήτησης για τη διάρκεια ενός χρόνου. Τα ευρήματα τους είναι ότι κατά τη

διάρχεια του χρόνου, περίπου το ένα τρίτο των υποβαλλόμενων επερωτήσεων ήταν επαναλήψεις επερωτήσεων που είχαν ήδη υποβληθεί από τον ίδιο χρήστη. Αν και αυτές οι μελέτες δεν έχουν εστιάσει στο θέμα του caching των αποτελεσμάτων των μηχανών αναζήτσηςη, όλες τους διαπιστώνουν ότι υπάρχει αρκετά μεγάλη τοπικότητα πράγμα το οποίο ευνοεί την υιοθέτηση τεχνικών caching σε μηχανές αναζήτησης.

5

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

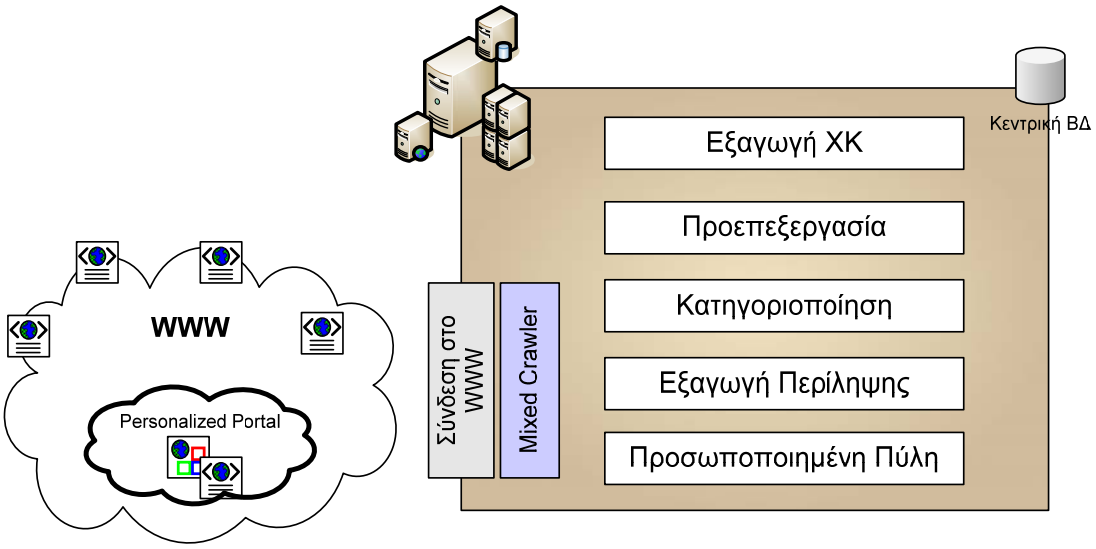
Στο κεφάλαιο αυτό περιγράφεται η αρχιτεκτονική του συστήματος που αναπτύχθηκε. Πιο συγκεκριμένα υπάρχουν στοιχεία που αφορούν:

- Τη γενική αρχιτεκτονική
- Τα υποσυστήματα συλλογής πληροφορίας, εξαγωγής κειμένου, προεπεξεργασίας, κατηγοριοποίησης, εξαγωγής περίληψης, παρουσίασης πληροφορίας & προσωποποίηση στο χρήστη

5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

5.1. Γενική Αρχιτεκτονική

Το σύστημα πάνω στο οποίο βασίσαμε και αξιολογήσαμε τους αλγόριθμους προσωποποιημένης αναζήτησης που θα παρουσιαστούν σε επόμενο κεφάλαιο αποτελείται από μικρότερα υποσυστήματα προκειμένου να είναι εύκολη η αυτόνομη σχεδίαση, κατασκευή και χρήση τους καθότι οι μηχανισμοί που οδηγούν στο επιθυμητό αποτέλεσμα είναι πολλοί. Το παρακάτω γενικό σχήμα μας δίνει τη γενική αρχιτεκτονική του συστήματος.



Εικόνα 4: Γενική Αρχιτεκτονική του Συστήματος

5.2. Υποσυστήματα

Εν συνεχεία θα παρουσιαστούν συνοπτικά τα υποσυστήματα του μηχανισμού εκτός της διαδικασίας προσωποποίησης στο χρήστη της οποίας πιο αναλυτική παρουσίαση θα γίνει στην επόμενη παράγραφο.

5.2.1. Συλλογή Πληροφορίας

Για τη συλλογή της πληροφορίας για το σύστημά μας και προκειμένου να τροφοδοτούμε αδιάκοπτα το σύστημα με άρθρα από το Διαδίκτυο εκμεταλλευόμαστε την τάση που επικρατεί σε όλους τους δικτυακούς τόπους να προσφέρουν κανάλια άμεσης επικοινωνίας με τους χρήστες και δεν μιλούμε για κάτι διαφορετικό από την τεχνολογία των RSS feeds. Με χρήση ενός απλού mixed selective crawler, που συνδυάζει wrapper και crawler λαμβάνουμε τις HTML σελίδες. Ο wrapper στο μηχανισμό συλλογής πληροφορίας εντοπίζει μέσα στα XML αρχεία εκείνα τα σημεία τα οποία περιέχουν πληροφορίες για τα άρθρα που θέλουμε να εξάγουμε. Εξάγοντας τον τίτλο του άρθρου καθώς επίσης και τη διεύθυνση στην οποία βρίσκεται μπορούμε στη συνέχεια με χρήση crawler να «επισκεπτόμαστε» τα εξαγόμενα URL και να λαμβάνουμε τον HTML κώδικα.

5.2.2. Εξαγωγή Χρήσιμου κειμένου (φιλτράρισμα)

Για την εξαγωγή του χρήσιμου κειμένου χρησιμοποιείται η ιδιότητα της HTML να μπορεί να αναπαρασταθεί σε δενδρική μορφή σύμφωνα με το μοντέλο Document Object (DOM). Η διαδικασία εκτελείται σε δύο επίπεδα, αυτό της

εφαρμογής και αυτό της βάσης δεδομένων. Από τη βάση δεδομένων λαμβάνουμε τον HTML κώδικα του άρθρου καθώς και άλλες πληροφορίες που έχουν εξαχθεί από το προηγούμενο στάδιο. Η HTML σελίδα αποδομείται και εντοπίζονται τα συστατικά του DOM που περιέχουν χρήσιμες πληροφορίες για το μηχανισμό.

5.2.3. Προεπεξεργασία κειμένου

Κατά την φάση της προεπεξεργασίας του κειμένου εξάγονται οι λέξεις κλειδιά για κάθε άρθρο. Η ανάγνωση των πληροφοριών για την επεξεργασία των άρθρων μπορεί να γίνει είτε από αρχείο είτε από τη βάση δεδομένων. Παράμετροι που χρησιμοποιούνται για το λόγο αυτό αφορούν το ελάχιστο μήκος λέξης, το αν θα γίνει ή όχι αποθήκευση αριθμών, τι είδους λίστα stopwords θα χρησιμοποιηθεί καθώς και τον αλγόριθμο που θα χρησιμοποιηθεί για εξαγωγή ρίζας από τις λέξεις (stemming).

5.2.4. Κατηγοριοποίηση Κειμένου

Το υποσύστημα της κατηγοριοποίησης κειμένου αποτελεί ένα από τα κεντρικά συστατικά του μηχανισμού το οποίο σε συνδυασμό με εκείνο της εξαγωγής περίληψης αποτελούν τον πυρήνα του μηχανισμού. Η είσοδος του υποσύστηματος κατηγοριοποίησης κειμένου είναι XML αρχεία τα οποία περιέχουν την έξοδο του υποσυστήματος εξαγωγής κωδικολέξεων. Ο βασικός στόχος του υποσυστήματος αυτού είναι η εφαρμογή αλγορίθμων κατηγοριοποίησης στο κείμενο και επομένως η αντιστοίχηση του κειμένου με κάποια από τις ήδη υπάρχουσες κατηγορίες. Η έξοδος του υποσυστήματος κατηγοριοποίησης, οι συσχετίσεις δηλαδή του κειμένου με κάθε κατηγορία, αποθηκεύονται στη βάση δεδομένων του συστήματος.

5.2.5. Εξαγωγή Περίληψης Κειμένου

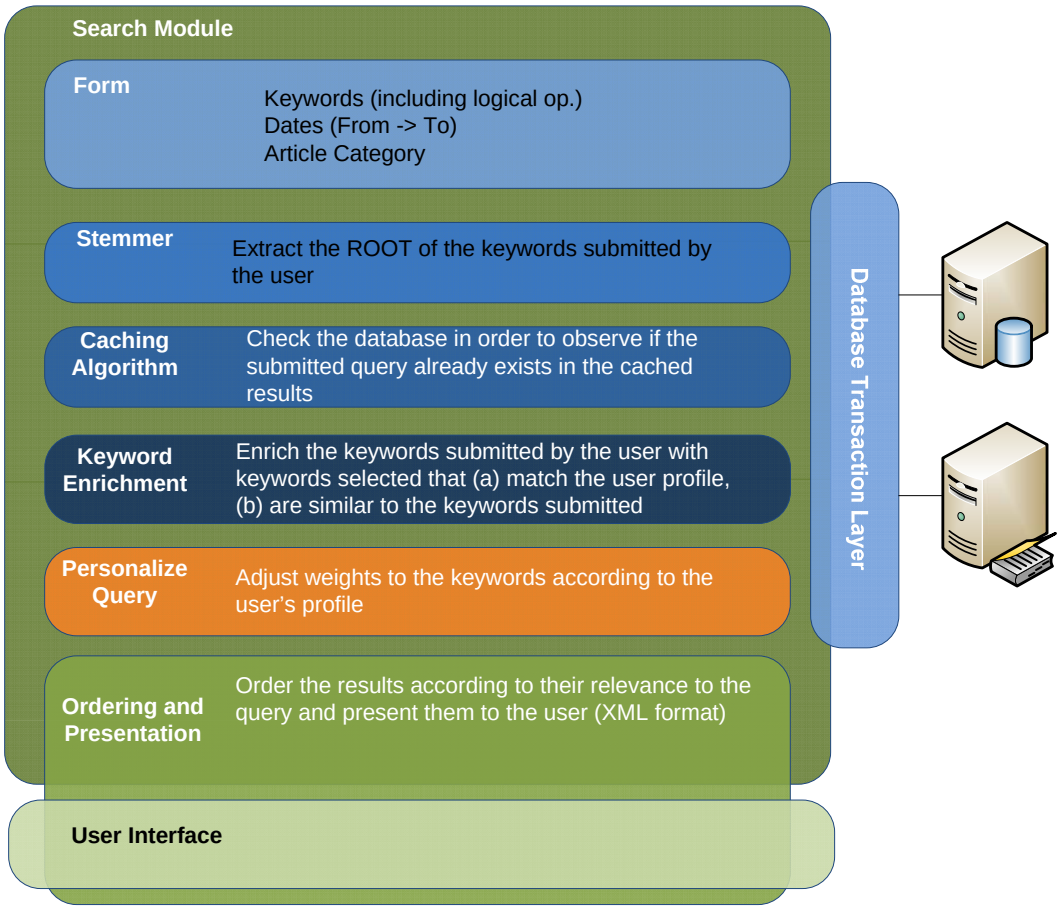
Η διαδικασία εξαγωγής αυτόματης περίληψης βασίζεται σε ευρεστικές μεθόδους και πιο συγκεκριμένα σε αξιολόγηση και βαθμοδότηση των προτάσεων του κειμένου προκειμένου να εξαχθούν οι πιο σημαντικές από αυτές και να αποτελέσουν την περίληψη του κειμένου. Η έξοδος επομένως του υποσυστήματος αυτόματης εξαγωγής περίληψης κειμένου είναι μια φθίνουσα σειρά προτάσεων με βάση το σκορ που αξιολογεί ο μηχανισμός πως πρέπει να έχουν όσον αφορά την σημαντικότητά τους για να αναπαραστήσουν το κείμενο.

5.3. Παρουσίαση Πληροφορίας και Προσωποποίηση στο χρήστη

Το τελικό στάδιο της αρχιτεκτονικής του συστήματος είναι η παρουσίαση της πληροφορίας στον τελικό χρήστη. Πρόκειται ίσως για το πιο σημαντικό στάδιο του συστήματος καθώς αποτελεί το περιβάλλον διεπαφής με τους χρήστες. Απώτερος σκοπός είναι ο χρήστης να μην αντιλαμβάνεται όλες τις διεργασίες που λαμβάνουν χώρα και να απολαμβάνει ποιοτικά και γρήγορα αποτελέσματα βάση των προσωπικών του επιλογών. Για την προσωποποίηση στο χρήστη μπορούν να χρησιμοποιηθούν δύο μέθοδοι:

- Ο χρήστης να δώσει κάποια πληροφορία στο σύστημα και το σύστημα να ξεκινήσει παρουσιάζοντας εξ αρχής προσωποποιημένα αποτελέσματα και να συγκλίνει γρήγορα στις ανάγκες του χρήστη.
- Ο χρήστης να μη δώσει καθόλου πληροφορία στο σύστημα και το σύστημα να ξεκινήσει παρουσιάζοντας γενικές πληροφορίες και να αργήσει να συγκλίνει στις προσωπικές επιλογές του χρήστη.

Σε κάθε περίπτωση το επιθυμητό επιτυγχάνεται και πρόκειται για τη σύγκλιση των πληροφοριών που παρουσιάζονται στις ανάγκες του χρήστη. Το παρακάτω σχήμα απεικονίζει την αρχιτεκτονική του συγκεκριμένου μηχανισμού.



Εικόνα 5: Αρχιτεκτονική της Προσωποποιημένης Πύλης

Όπως φαίνεται και από το παραπάνω σχήμα, ο χρήστης υποβάλλει τις λέξεις κλειδιά για την αναζήτηση. Παράλληλα με τις κωδικολέξεις που είναι πολύ βασικές για την αναζήτηση, ο χρήστης παρέχει και κάποιες άλλες παραμέτρους για την διαδικασία όπως:

- Χρονική περίοδο για τα άρθρα
- Λογικό τελεστή που συνδέει τις κωδικολέξεις
- Θεματική Κατηγορία ενδιαφέροντος

Οι κωδικολέξεις τροφοδοτούνται σε ένα αλγόριθμο stemming ώστε να εξαχθούν οι ρίζες τους. Αυτό είναι απαραίτητο διότι οι λέξεις από τα άρθρα που φτάνουν στο σύστημα καθημερινά αποθηκεύνται με αυτήν την μορφή στη βάση δεδομένων. Στην συνέχεια και πριν εκκινήσει ο αλγόριθμος της αναζήτησης εκτελείται μια διαδικασία που ελέγχει αν υπάρχουν cached αποτελέσματα στη ΒΔ από προγενέστερες αναζητήσεις. Αν ο ίδιος χρήστης έχει υποβάλλει στο παρελθόν μια παρόμοια επερώτηση τότε τα αποτελέσματα θα προκύψουν αρκετά γρηγορότερα αφού δε θα χρειαστεί να γίνει νέα αναζήτηση στην βάση δεδομένων. Ο μηχανισμός του caching θα παρουσιαστεί ξεχωριστά στο κεφάλαιο 8 και η αξιολόγηση των αποτελεσμάτων θα παρουσιαστεί στο κεφάλαιο 9. Στο επόμενο βήμα, η επερώτηση επεκτείνεται με παραπάνω λέξεις κλειδιά. Τα κριτήρια που χρησιμοποιούνται για τον εμπλουτισμό της επερώτησης αφορούν:

- Στο προφίλ που έχει διαμορφωθεί για το συγκεκριμένο χρήστη μετά από πολλές συνεδρίες του (sessions) εντός του συστήματος.
- Στις ίδιες τις λέξεις κλειδιά που υποβλήθηκαν στην επερώτηση. Το σύστημα βρίσκει επιπλέον λέξεις κλειδιά σχετικές με τις κωδικολέξεις που υπέβαλλε ο χρήστης. Η θεματική κατηγορία που επιλέχθηκε είναι σημαντικός παράγοντας στη διαδικασία αυτή.

Στο προτελευταίο στάδιο του υποσυστήματος προσωποποίησης στο χρήστη, ένας αλγόριθμος απόδοσης βαρών χρησιμοποιείται για να ρυθμίσει τη σημαντικότητα της κάθε κωδικολέξης στην διαδικασία της αναζήτησης σχετικών άρθρων στην ΒΔ. Και σε αυτό το σημείο, αξίζει να σημειώσουμε ότι το προφίλ του χρήστη παίζει ρόλο στον τρόπο με τον οποίο θα αποδωθούν τα βάρη στις κωδικολέξεις αφού για κάθε χρήστη υπάρχει ένα σύνολο λέξεων που έχουν χαρακτηριστεί ώς «αγαπημένες» με αποτέλεσμα να αποδίδονται σε αυτές μεγαλύτερα βάρη. Στη τελική φάση λειτουργίας του υποσυστήματος, τα άρθρα που ανακτώνται από τη βάση δεδομένων επαναδιατάσσονται με βάση τη σχετικότητα τους ώς προς την επερώτηση και στη συνέχεια παρουσιάζονται στον τελικό χρήστη.

6

ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Στο κεφάλαιο αυτό οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση κάθε υποσυστήματος του μηχανισμού.

6. ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Οι τεχνολογίες που χρησιμοποίηθηκαν σε κάθε επίπεδο του μηχανισμού είναι διαφορετικές προκειμένου να επιτευχθεί η μέγιστη απόδοση συνολικά του συστήματος με τη χρήση κάθε μίας από αυτές.

Η επιλογή της τεχνολογίας που θα ακολουθηθεί κατά την κατασκευή ενός σύνθετου συστήματος είναι εξαιρετικά σημαντική προκειμένου να δημιουργηθεί ένα καθολικό σύστημα το οποίο να είναι ευέλικτο, να υποστηρίζει εύκολα αλλαγές και αναβαθμίσεις, να αποτελείται από υποσυστήματα και τέλος να βασίζεται σε ανοιχτά πρότυπα. Το σύστημα που υλοποιήθηκε είναι σύνθετο καθότι έχει βάση το διαδίκτυο αλλά ένα σημαντικό κομμάτι του, ίσως ο πυρήνας, κρύβεται στο μηχανισμό που πραγματοποιεί κατηγοριοποίηση κειμένου και γενικότερα διαχείριση πληροφορίας. Ο τελευταίος μηχανισμός ουσιαστικά δεν έχει καμία επαφή με το διαδίκτυο και φυσικά δεν είναι και απαραίτητο να έχει. Βέβαια, τα δεδομένα που δέχεται προέρχονται από εξόρυξη πληροφορίας στο διαδίκτυο (HTML σελίδες) ενώ τα δεδομένα που εξάγει χρησιμοποιούνται προκειμένου να τροφοδοτήσουν το portal με περιεχόμενο.

6.1. Βάση Δεδομένων

Οι πιθανές επιλογές που έχουμε όσον αφορά τη βάση δεδομένων του συστήματος προέρχονται από την επιλογή των τεχνολογιών για τους μηχανισμούς κατηγοριοποίησης και κατασκευής του Portal. Συνεπώς θα πρέπει να επιλεγεί μία βάση δεδομένων η οποία να είναι πλήρως συμβατή με το μηχανισμό που θα κατηγοριοποιεί καθώς επίσης και με τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για την κατασκευή του portal. Θεωρούμε πως ο μηχανισμός δημιουργίας του δυναμικού προφίλ δύναται να ενταχθεί, είτε στο μηχανισμό κατηγοριοποίησης είτε στο μηχανισμό κατασκευής του portal.

6.1.1. Γιατί MySQL

Η MySQL είναι η δημοφιλέστερη Βάση Δεδομένων ανοιχτού κώδικα που προσφέρεται από το Δίκτυο MySQL. Η αρχιτεκτονική της την κάνουν να είναι εξαιρετικά γρήγορη και πολύ εύκολη σε αλλαγές και αναβαθμίσεις. Επιτρέπει επαναχρησιμοποίηση κώδικα όπου αυτό είναι αναγκαίο και παρέχει ένα μινιμαλιστικό τρόπο δημιουργίας στοιχείων διαχείρισης βάσης δεδομένων τέτοια ώστε να κάνουν τη MySQL ασύγκριτη σε ταχύτητα, σε κατάληψη χώρου, σταθερότητα και ευκολία. Ο μοναδικός στο είδος του διαχωρισμός του κεντρικού πυρήνα του server από το μηχανισμό αποθήκευσης κάνει δυνατή την ύπαρξη αυστηρού ελέγχου σε συναλλαγές και μείωση ταχύτητας ή ύπαρξη θεαματικά μεγάλης ταχύτητας με απευθείας προσπέλαση των δεδομένων στοιχεία που μπορεί να χρησιμοποιηθούν ανάλογα με τις ανάγκες των χρηστών.

Η MySQL περιλαμβάνει αποθήκευση σε μηχανή InnoDB, η οποία υποστηρίζει ασφάλεια στις συναλλαγές και ACID-συμβατή μηχανή αποθήκευσης με commit, rollback, crash recovery και low-level locking δυνατότητες.

Η έκδοση της MySQL που βρίσκεται αυτή τη στιγμή σε σταθερή κατάσταση είναι η 4.1.12 και υποστηρίζει πολλά στοιχεία που αφορούν την απόδοση, τη διεθνοποίηση και τη δυνατότητα ένταξης του MySQL server σε άλλα στοιχεία υλικού και λογισμικού. Τα πιο βασικά στοιχεία που χαρακτηρίζουν τη MySQL είναι:

- Υποερωτήματα, που επιτρέπουν στους χρήστες να κάνουν σύνθετα ερωτήματα με μεγάλη ευκολία και αποδοτικά.
- Γρήγορη επικοινωνία μεταξύ server και client μέσα από ένα καινούριο πρωτόκολλο
- Μικρότερη κατανάλωση πόρων από το server μέσα από βελτιστοποίηση στις βιβλιοθήκες

- Υποστήριξη Unicode, διεθνείς χαρακτήρες και υποστήριξη αποθήκευσης στην πλειοψηφία των συνόλων χαρακτήρων
- Υποστήριξη τύπων GIS για ερωτήματα που αφορούν χάρτες και γεωγραφικά δεδομένα

Τα παραπάνω στοιχεία κάνουν τη MySQL ένα υπερπολύτιμο εργαλείο στα χέρια κάποιου χρήστη και τη θέτουν στην 1η θέση για επιλογή ως βάση δεδομένων του συστήματός μας.

6.1.2. Γιατί PostgreSQL

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων βασισμένη στα αντικείμενα. Ουσιαστικά προέρχεται από την POSTGRES, V 4.2, που έχει δημιουργηθεί στο πανεπιστήμιο της Καλιφόρνια στο τμήμα Επιστήμης των Υπολογιστών του Μπέρκλεϋ. Μάλιστα το συγκεκριμένα σύστημα υλοποίησε πολλές λειτουργικότητες πολλά χρόνια πριν εφαρμοστούν στα πιο γνωστά από τα σημερινά συστήματα βάσεων δεδομένων.

Η PostgreSQL είναι ένας ανοιχτού κώδικα απόγονος του αρχικού κώδικα που γράφηκε στο Μπέρκλεϋ. Υποστηρίζει SQL92 και SQL99 και προσφέρει πολλά στοιχεία που υποστηρίζουν οι περισσότερες βάσεις δεδομένων τελευταίας τεχνολογίας όπως:

- Σύνθετα ερωτήματα
- Foreign Keys
- Triggers
- Διαφορετικές όψεις
- Ακεραιότητα στις συναλλαγές
- Συνεργασία ταυτόχρονων πολλαπλών εκδόσεων

Επιπρόσθετα, η PostgreSQL μπορεί να εμπλουτιστεί σε στοιχεία από κάποιον έμπειρο χρήστη με πολλούς τρόπους ώστε να υποστηρίζει νέα:

- Τύπους δεδομένων
- Συναρτήσεις
- Διαχειριστές
- Συναθροιστικές συναρτήσεις
- Μεθόδους ευρετηρίου
- Διαδικασιακές γλώσσες

Τέλος, αξίζει να τονιστεί η γενναιοδωρία της άδειας κάτω από την οποία βρίσκεται η PostgreSQL σύμφωνα με την οποία μπορεί να χρησιμοποιηθεί, αλλαχθεί και διακινηθεί από τον καθένα χωρίς κανένα κόστος.

6.1.3. Επιλέγοντας τη Βάση Δεδομένων

Σύμφωνα με τα παραπάνω αλλά και λαμβάνοντας υπόψη μας τους σκοπούς που έχει το σύστημά μας καταλήξαμε στην επιλογή της MySQL σαν τη βάση δεδομένων που θα χρησιμοποιηθεί στο σύστημα. Συγκρίνοντας τις δύο βάσεις δεδομένων μπορούμε να καταλήξουμε στο ότι διαθέτουν πολλά κοινά στοιχεία, ωστόσο η MySQL φαίνεται να είναι πιο διαδεδομένη, ένας λόγος ο οποίος την κάνει πιο ισχυρή. Επιπρόσθετα τα στοιχεία διεθνοποίησης που διαθέτει φαίνονται πολύ χρήσιμα για ένα σύστημα το οποίο μελλοντικά μπορεί να επεκταθεί ώστε να υποστηρίζει πολλές γλώσσες. Ένα άλλο στοιχείο που μας οδηγεί στην επιλογή της MySQL είναι και το γεγονός πως οι βοηθητικοί crawlers που τροφοδοτούν το σύστημά μας με σελίδες HTML υποστηρίζουν βάση δεδομένων MySQL. Τέλος θα πρέπει να λάβουμε υπόψη μας το γεγονός πως δημιουργούμε ένα σύστημα πολυεπίπεδο με τη βάση δεδομένων να είναι ο ουσιαστικός σύνδεσμος μεταξύ των περισσοτέρων κομματιών και συνεπώς μία βάση δεδομένων με μεγάλη σταθερότητα και αξιοπιστία θα προσέδιδε κύρος στο συνολικό σύστημα.

Καταλήγουμε λοιπόν στη χρήση Mysql Server έκδοση 4.12.

6.2. Τεχνολογία Δημιουργίας Portal

Όσον αφορά την τεχνολογία που θα χρησιμοποιηθεί για τη δημιουργία του portal θα πρέπει να επισημανθεί ότι θα χρησιμοποιηθεί κάποια τεχνολογία δημιουργίας δυναμικών σελίδων. Οι σελίδες θα πρέπει να έχουν απλή δομή και κατανοητή προκειμένου να μην αποπροσανατολίζεται ο χρήστης. Για το σκοπό αυτό η δυνατότητα που μας δίνεται είναι να χρησιμοποιήσουμε μία εκ των PHP ή JSP. Η τεχνολογία ASP.NET αποκλείεται γιατί αίρει το χαρακτήρα ανοικτού κώδικα που βασίζεται σε ανοικτά στάνταρ.

6.2.1. Γιατί PHP

Η ευκολία στη χρήση αλλά και η ομοιότητα με της πιο κοινές γλώσσες δομημένου προγραμματισμού κάνουν την PHP μία γλώσσα η οποία ελκύει τους προγραμματιστές και οι πιο έμπειροι από αυτούς βρίσκουν εύκολη τη δημιουργία σύνθετων εφαρμογών από την πρώτη στιγμή που θα έρθουν σε επαφή με την PHP. Επίσης επιτρέπει στους έμπειρους χρήστες να δημιουργήσουν εφαρμογές Διαδικτύου με δυναμικό περιεχόμενο χωρίς να χρειάζεται να αναλωθούν σε πρακτικές ή να χρειαστεί να αποστηθίσουν σειρές από συναρτήσεις.

Ένα από τα πιο ελκυστικά κομμάτια της PHP είναι το γεγονός ότι είναι κάτι περισσότερο από μια προγραμματιστική γλώσσα. Εξαιτίας της κλιμακωτής σχεδίασής της, μπορεί να χρησιμοποιηθεί και για τη δημιουργία γραφικών περιβαλλόντων απεικόνισης, και για την εκτέλεση προγραμμάτων μέσω της γραμμής εντολών

Η PHP επιτρέπει την αλληλεπίδραση με ένα μεγάλο αριθμό σχεσιακών βάσεων δεδομένων όπως είναι οι Mysql, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL και SQLite ενώ η σύνταξη που χρησιμοποιείται είναι απλή και κατανοητή. Τρέχει στα περισσότερα λειτουργικά συστήματα όπως UNIX, Linux, Windows και Mac OS X και μπορεί να υποστηριχθεί σχεδόν από όλους τους γνωστούς εξυπηρετητές εφαρμογών Διαδικτύου.

Η PHP είναι αποτέλεσμα μίας σειράς προσπαθειών από πολλούς συμμετέχοντες. Τα δικαιώματα παρέχονται με ένα SD-style license. Τέλος, μετά την έκδοση 4 η PHP υποστηρίζεται από τη μηχανή Zend.

6.2.2. Γιατί JSP

Η JSP έρχεται σαν απάντηση της Java στις τεχνολογίες εφαρμογών διαδικτύου. Χρησιμοποιεί τεχνολογία που βασίζεται είτε σε Java Servlets ή σε Java Beans και προσφέρει δυνατότητα ανάλογα με την επιλογή της τεχνολογίας να δημιουργηθούν από πολύ απλές Διαδικτυακές εφαρμογές μέχρι πολύ σύνθετες.

Όσον αφορά την αρχιτεκτονική, η jsp μπορεί να θεωρηθεί σαν servlet με πολύ υψηλού επιπέδου αφαίρεση η οποία υλοποιείται σαν επέκταση του API 2.1 των Servlet.

Όσον αφορά τη σύνταξη, μία σελίδα γραμμένη σε JSP μπορεί να χωριστεί στα εξής κομμάτια

Στατικό περιεχόμενο (π.χ. HTML)

- JSP directives
- JSP μεταβλητές και στοιχεία κώδικα
- JSP action
- Tags γραμμένα από το χρήστη

Πρόκειται για τη γλώσσα προγραμματισμού που χρησιμοποιείται στις περισσότερες σύνθετες εφαρμογές που δημιουργούνται στο Διαδίκτυο γιατί προσφέρει τη δυνατότητα με τη χρήση συνδυασμού καθαρής Java, μέσω των

Beans και μίας C-like γλώσσας προγραμματισμού για τη δημιουργία απλού δυναμικού περιεχομένου. Ωστόσο προορίζεται κυρίως για έμπειρους χρήστες που μπορούν να καταλάβουν τη διαφορά αντικειμενοστραφούς και συναρτησιακού προγραμματισμού και να τα συνδυάσουν κατάλληλα προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα.

6.3. Τελική επιλογή τεχνολογιών

Η τελική επιλογή τεχνολογιών όπως αναφέρθηκε και στην αρχή του κεφαλαίου βασίζεται στο γεγονός ότι θα γίνει συνδυασμός τεχνολογιών που θα συνδυάζουν καθαρό αντικειμενοστραφή κώδικα με σελίδες του διαδικτύου. Θα μπορούσε κανείς να πει πως η επιλογή Java, JSP και Oracle θα ήταν ιδανικός για ένα τέτοιο σύστημα καθότι είναι εκ των πραγμάτων τεχνολογίες που η δυνατότητα διασύνδεσής τους είναι εύκολη και οι δυνατότητες που προσφέρει ο συγκεκριμένος συνδυασμός είναι πολλές.

Ωστόσο, επειδή ακριβώς τα υποσυστήματα που απαρτίζουν το μηχανισμό που δημιουργήσαμε μπορούν να λειτουργήσουν ανεξάρτητα και αυτόνομα, η επιλογή των τεχνολογιών έγινε περισσότερο βάση γενικών αρχών και προτύπων προκειμένου να καταλήξουμε σε ένα τελικό σύστημα ανοιχτό, και ευέλικτο το οποίο θα μπορεί να επιδέχεται βελτιώσεις σε κάθε κομμάτι του ξεχωριστά. Έγινε, δηλαδή, προσπάθεια να μη δημιουργηθούν επικαλύψεις στον κώδικα αλλά η διασύνδεση των υποσυστημάτων να γίνει σε επίπεδο βάσης δεδομένων. Αυτό βέβαια δε μας απαγορεύει να χρησιμοποιούμε ένα κεντρικό μηχανισμό που θα κάνει διαχείριση όλων των υποσυστημάτων. Συνεπώς καταλήγουμε σε γλώσσα διαδικτύου PHP με υποστήριξη βάσης δεδομένων MySQL γιατί επιθυμούμε απλότητα σε επίπεδο web site, και σε Java και C++ με υποστήριξη βάσης δεδομένων MySQL προκειμένου να γίνονται όλες οι διαδικασίες που χρειάζονται εκτενείς αναλύσεις και υπολογισμούς.

6.4. Μηχανισμός παρουσίασης πληροφορίας και προσωποποίησης

Ο μηχανισμός παρουσίασης πληροφορίας και προσωποποίησης ανήκει στο κομμάτι που αφορά το δικτυακό τόπο και ως εκ τούτου υλοποιείται αποκλειστικά σε PHP που είναι και η γλώσσα κατασκευής του δικτυακού. Παράλληλα, για την καλύτερη παρουσίαση των δεδομένων στον τελικό χρήστη γίνεται εκτενής χρήση τεχνολογίας AJAX. Σε αυτό το κομμάτι έχει προκύψει αρκετές φορές το ζήτημα επιλογής τεχνολογίας καθότι μία πιο ολοκληρωμένη πρόταση θα ήταν υλοποίηση όλων των μηχανισμών σε Java και επιλογή JSP με Enterprise Java beans για το δικτυακό τόπο. Ωστόσο, η απόκριση της γλώσσας προγραμματισμού Java στις διαδικασίες πυρήνα του συστήματος μας είναι πολύ πιο αργή από τη C++. Αυτό συμβαίνει κυρίως, όπως έχει ήδη αναφερθεί, στην καλύτερη αντιμετώπιση που έχει η C++ όταν εκτελεί διαδικασίες χαμηλού επιπέδου.

6.5. Διασύνδεση μηχανισμών

Η διασύνδεση των μηχανισμών βασίζεται αποκλειστικά στο επίπεδο βάσης δεδομένων αλλά και στη σειριακή εκτέλεση των διαδικασιών που προσφέρει το λειτουργικό σύστημα. Το γεγονός ότι χρησιμοποιούνται πολλαπλά επίπεδα στην υλοποίηση είναι σωτήριο για ένα τέτοιο σύστημα καθότι υπάρχει ένα επίπεδο το οποίο είναι κοινό για όλα τα υποσυστήματα και συνεπώς είναι εφικτή η ανταλλαγή δεδομένων. Παράλληλα, όλοι οι μηχανισμοί του συστήματος έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να δέχονται δεδομένα από δύο διαφορετικά κανάλια και αντίστοιχα να εξάγουν τα δεδομένα σε δύο διαφορετικά κανάλια, το ένα αυτό της βάσης δεδομένων και το άλλο σε μορφή XML. Μιλούμε για το κλασσικό πρότυπο μίας n-tier αρχιτεκτονικής η οποία επιτυγχάνει διασύνδεση των αυτόνομων μηχανισμών που την αποτελούν στο επίπεδο καναλιού επικοινωνίας. Με αυτό τον τρόπο έχουν μηχανισμούς που αποδεσμεύονται όσο αφορά το κομμάτι της υλοποίησης και δεν έχουν κανένα περιορισμό αρκεί να μπορούν να «διαβάσουν»

δεδομένα από βάση δεδομένων ή από XML αρχεία και αντίστοιχα να είναι σε θέση να «γράψουν» σε βάση δεδομένων ή σε XML αρχεία.

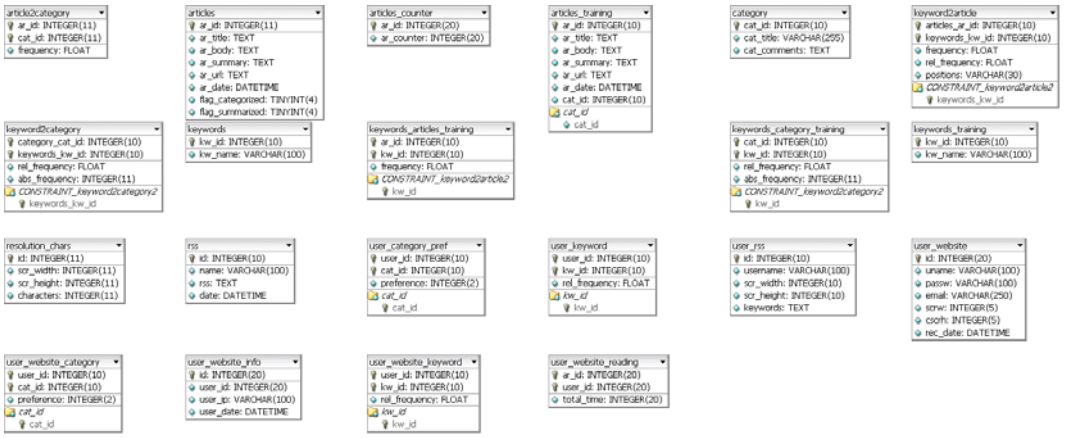
7

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στο κεφάλαιο αυτό περιγράφεται η Βάση Δεδομένων του συστήματος. Πιο αναλυτικά παρουσιάζονται εκτενώς οι πίνακες που χρησιμοποιεί συνολικά το σύστημα σε όλα τα στάδια λειτουργίας του.

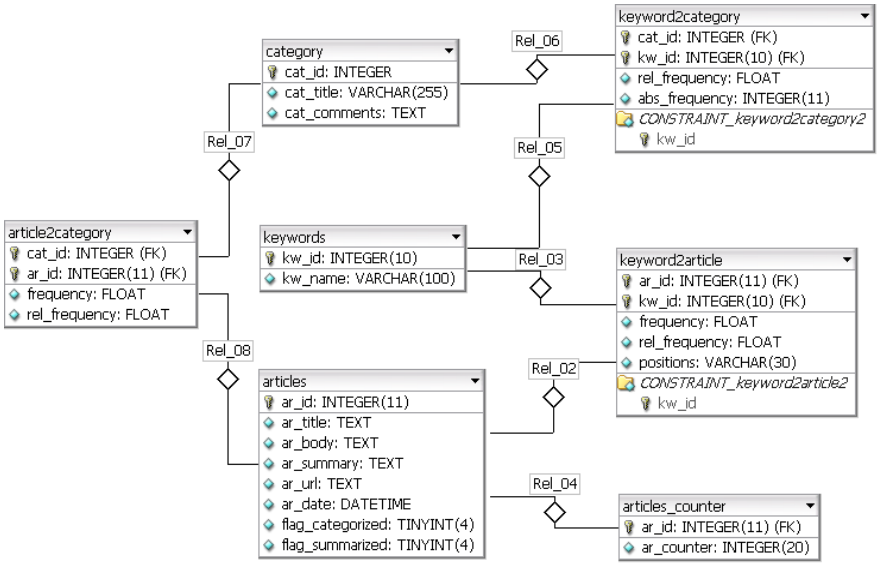
7. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η βάση δεδομένων που χρησιμοποιούμε στο σύστημά μας είναι η MySQL 5.0.44 και η οποία αποτελεί και το ουσιαστικό επίπεδο διασύνδεσης μεταξύ των διαφορετικών υποσυστημάτων που έχουν υλοποιηθεί. Μία γενική εικόνα της βάσης δεδομένων φαίνεται στην εικόνα 6.

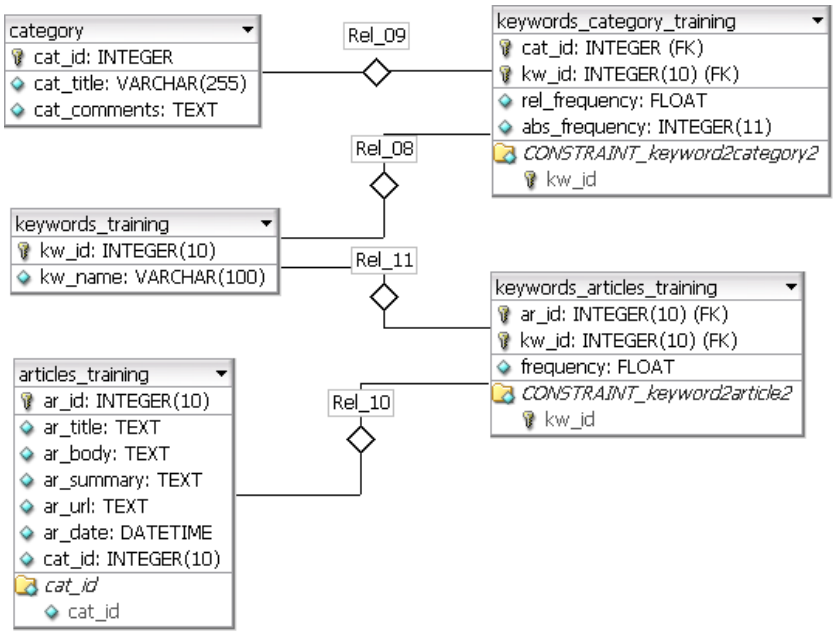


Εικόνα 6: Οι πίνακες της βάσης δεδομένων

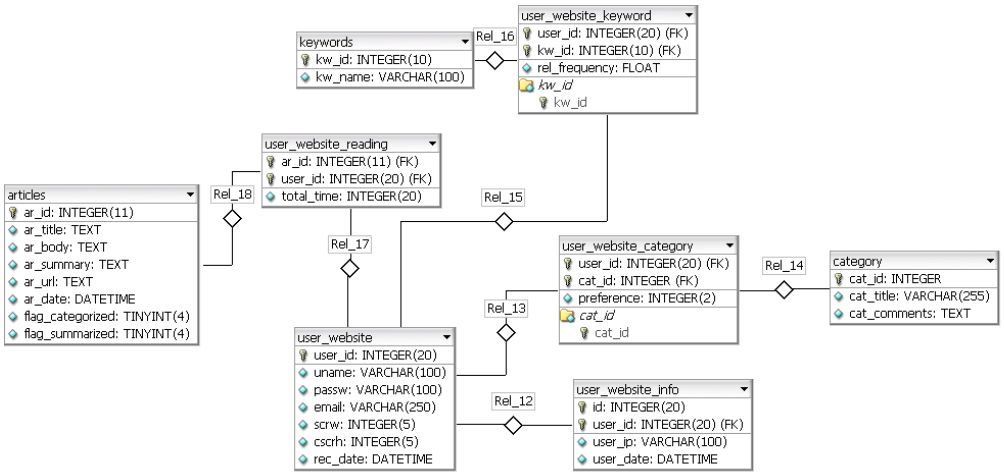
Η εικόνα της βάσης δεδομένων είναι πολύ γενική και οι πίνακές της μπορούν να ομαδοποιηθούν προκειμένου να παρουσιαστεί ο ακριβής τρόπος με τον οποίο γίνεται η αλληλεπίδραση μεταξύ των πινάκων της.



Εικόνα 7: Πίνακες που αφορούν τα άρθρα που εισέρχονται στο σύστημα



Εικόνα 8: Πίνακες που αφορούν τη βάση γνώσης του συστήματος



Εικόνα 9: Πίνακες που αφορούν τους χρήστες του συστήματος

7.1. Ανάλυση γενικών πινάκων και πινάκων βάσης γνώσης

Στο επόμενο κομμάτι, ακολουθεί η ανάλυση των πινάκων που υπάρχουν στη βάση δεδομένων και λεπτομερής παρουσίασή τους σε κάθε σημείο χρήσης τους στις διαδικασίες του συστήματος.

7.1.1. article2category

Πρόκειται για έναν πίνακα που περιέχει στοιχεία που συσχετίζουν τα άρθρα της βάσης με τις θεματικές κατηγορίες του συστήματος που υπάρχουν στον πίνακα category. Δεν είναι απαραίτητο κάθε άρθρο να σχετίζεται μόνο με μια κατηγορία αλλά γι' αυτό το σύστημα μας υπολογίζει τη συσχέτιση του άρθρου με κάθε κατηγορία του συστήματος. Τα πεδία του συγκεκριμένου πίνακα είναι τα ακόλουθα:

- **ar_id**: πρόκειται για ένα ξένο κλειδί που αποτελεί μοναδικό αναγνωριστικό κάθε άρθρου του συστήματος. Αναφέρεται στον πίνακα articles και σε ζευγάρι μαζί με το επόμενο πεδίο cat_id αποτελούν πρωτεύον κλειδί του πίνακα.
- **cat_id**: επίσης ξένο κλειδί στον πίνακα category. αναφέρεται στην κατηγορία με την οποία συσχετίζεται κάθε άρθρο.
- **frequency**: πρόκειται για τη σχετική συχνότητα βάσει της οποία συσχετίζουμε κάθε άρθρο του συστήματος με μια κατηγορία. Βάση αυτής της συχνότητας χαρακτηρίζεται ένα άρθρο αντιπροσωπευτικό ή όχι για μια θεματική κατηγορία.

7.1.2. articles_counter

Πρόκειται για τον πίνακα στον οποίο καταγράφονται τα hits τα οποία δέχεται κάθε άρθρο από τους χρήστες του συστήματος. Ο πίνακας αυτός χρησιμοποιείται ως μετρική για την κατάδειξη των άρθρων στα οποία δείχνουν ενδιαφέρον οι χρήστες του συστήματος. Τα πεδία είναι τα εξής:

- **ar_id**: πρόκειται για ένα ξένο κλειδί που αποτελεί μοναδικό αναγνωριστικό κάθε άρθρου του συστήματος. Αναφέρεται στον πίνακα articles και σε ζευγάρι μαζί με το επόμενο πεδίο cat_id αποτελούν πρωτεύον κλειδί του πίνακα.
- **ar_counter**: ένας ακέραιος αριθμός που καταγράφει τον ακριβή αριθμό των hits που έχει δεχθεί το άρθρο.

7.1.3. extraction_article_sentences

πρόκειται για τον πίνακα στον οποίο αποθηκεύονται τα κείμενα των άρθρων διαχωρισμένα σε προτάσεις πριν και μετά την προεπεξεργασία. Τα πεδία είναι τα ακόλουθα:

- **id**: το πρωτεύον κλειδί του πίνακα. Είναι και μοναδικό αναγνωριστικό κάθε καταχωρημένης πρότασης από τα κείμενα.
- **ar_id**: ξένο κλειδί που αναφέρεται στο αναγνωριστικό του άρθρου. Αντιστοιχίζεται στον πίνακα articles της βάσης.
- **sentence_number**: ακέραιος αριθμός που αντιστοιχεί στην σειρά που έχει μια πρόταση μέσα στο σώμα του άρθρου. Κάθε άρθρο κατά την προεπεξεργασία χωρίζεται σε μια ή και σε περισσότερες προτάσεις.
- **sentence_preprocessed**: είναι η πρόταση στην προεπεξεργασμένη της μορφή. έχουν αφαιρεθεί τα stopwords και οι λέξεις υπάρχουν σε μορφή stem (ρίζας).
- **sentence**: είναι η πρόταση όπως αυτή ανακτήθηκε στο κύριο σώμα κάθε άρθρου.

7.1.4. extraction_kw

Ένας από τους πιο συχνά χρησιμοποιούμενους πίνακες της βάσης δεδομένων. Περιέχει όλες τις λέξεις κλειδιά που προκύπτουν από την προεπεξεργασία των άρθρων και οι οποίες έχουν περάσει από διαδικασία stemming. Τα πεδία του πίνακα είναι τα ακόλουθα:

- **kw_id**: το αναγνωριστικό της κάθε άρθρου το οποίο αποτελεί και το πρωτεύον κλειδί του πίνακα.
- **kw_name**: η λέξη κλειδί σε μορφή ρίζας (stemmed)
- **lang_id**: το αναγνωριστικό της γλώσσας στην οποία ανήκει η λέξη. αποτελεί ξένο κλειδί στον πίνακα language.

7.1.5. extraction_kw2ar

Ο πίνακας αυτός χρησιμοποιείται προκειμένου να αποθηκευτούν οι λέξεις κλειδιά που έχουν εξαχθεί από τα άρθρα. Για τη βάση γνώσης δεν είναι ένας πίνακας ο οποίος έχει άμεση χρησιμότητα για τους αλγορίθμους του μηχανισμού. Ωστόσο, είναι ένας βοηθητικός πίνακας γιατί χρησιμοποιείται προκειμένου να ελεγχθούν άρθρα τα οποία βρίσκονται στη βάση γνώσης και είναι προβληματικά. Ως προβληματικά αναφέρονται τα άρθρα των οποίων οι λέξεις κλειδιά δεν ανταποκρίνονται στην ενότητα την οποία αντιπροσωπεύει μία κατηγορία.

- **ar_id**: το μοναδικό ξένο κλειδί που αναφέρεται στο άρθρο από το οποίο εξάγουμε τις λέξεις κλειδιά.
- **kw_id**: το μοναδικό ξένο κλειδί που αναφέρεται στις λέξεις κλειδιά που εξάγονται από το άρθρο με αναγνωριστικό κλειδί ar_id. Μαζί με το τελευταίο αποτελούν πρωτεύον κλειδί του συγκεκριμένου πίνακα συνεπώς δεν μπορούν να υπάρχουν διπλές εγγραφές με τα ίδια χαρακτηριστικά ar_id και kw_id.
- **abs_frequency**: η απόλυτη συχνότητα εμφάνης μιας λέξης κλειδί σε ένα άρθρο, δηλαδή το πλήθος των φορών που εμφανίζεται η λέξη στο άρθρο.
- **rel_frequency**: η σχετική συχνότητα με την οποία εμφανίζεται μια συγκεκριμένη λέξη κλειδί για ένα συγκεκριμένο άρθρο. Προκύπτει από το κλάσμα της απόλυτης συχνότητας αυτής της λέξης προς άθροισμα των απολύτων συχνοτήτων όλων το λέξεων κλειδιών του άρθρου. Αποτελεί μια βασική μετρική που μπορεί να δείξει τη σημαντικότητα μιας λέξης κλειδιού για ένα άρθρο.
- **sentences**: το πλήθος των προτάσεων από το άρθρο στις οποίες εμφανίζεται η λέξη κλειδί. Επίσης χρησιμοποιείται ως μετρική για να δείξει τη σχετικότητα της λέξης κλειδιού με το άρθρο.

7.1.6. keywords_category_training

Πρόκειται ίσως για τον πιο σημαντικό πίνακα της βάσης γνώσης. Σε αυτό τον πίνακα αποθηκεύονται πληροφορίες που αφορούν τις λέξεις κλειδιά που αντιπροσωπεύουν μία κατηγορία, ενώ παράλληλα αποθηκεύεται πληροφορία που αφορά το πόσο σημαντική είναι μία λέξη για μία κατηγορία.

- **cat_id**: Το μοναδικό ξένο κλειδί που αφορά την κατηγορία στην οποία ανήκει μία λέξη κλειδί.
- **kw_id**: Το μοναδικό ξένο κλειδί που αφορά τη λέξη κλειδί που ανήκει σε μία κατηγορία.
- **rel_frequency**: Πρόκειται για τη σχετική συχνότητα με την οποία εμφανίζεται μία λέξη κλειδί σε μία κατηγορία.
- **abs_frequency**: Η απόλυτη συχνότητα με την οποία εμφανίζεται μία λέξη κλειδί σε μία κατηγορία.

7.1.7. search_caching

ο πίνακας αυτός χρησιμοποείται κατά την αναζήτηση άρθρων στη βάση δεδεμένων με σκοπό την βελτίωση της απόδοσης του συστήματος μέσω μεθόδου caching. Εδώ αποθηκεύονται τα αποτελέσματα από επερωτήσεις που έγιναν στο παρελθόν μαζί με παραμέτρους για τις επερωτήσεις αυτές. Σε περίπτωση που κάποια επερώτηση επαναλαμβάνεται από κάποιο χρήστη, το σύστημα δεν εκτελεί αναζήτηση από την αρχή αλλά φέρνει τα cached δεδομένα γρήγορα από τη βάση ενώ σε ορισμένες περιπτώσεις εκτελείται μια αναζήτηση πιο περιορισμένης έκτασης για εμπλουτισμό του αποτελέσματος με πιο πρόσφατα άρθρα. Τα πεδία του πίνακα είναι τα ακόλουθα:

- **search_id**: το πρωτεύον κλειδί του πίνακα το οποίο αποτελεί μοναδικό αναγνωριστικό για κάθε σύνολο cached αποτελεσμάτων απο παρελθοντικές επερωτήσεις.

- **user_id**: ξένο κλειδί που αναφέρεται στον πίνακα user_website και περιέχει το αναγνωριστικό του χρήστη που έκτέλεσε την επερώτηση για πρώτη φορά.
- **exec_time**: ο ακριβής χρόνος εκτέλεσης της επερώτησης. Χρησιμοποιείται ως φίλτρο για να ελέγχεται το κατά πόσο τα cached δεδομένα δεν έχουν "παλιώσει".
- **query**: λίστα των λέξεων κλειδιών που συμμετείχαν στην επερώτηση. Μπορεί να περιέχει ένα η παραπάνω αναγνωριστικά από λέξεις κλειδιά του συστήματος.
- **parameters**: μια ακέραιη τιμή που αντιστοιχεί στον τρόπο με τον οποίο αναζητώνται τα αποτελέσματα βάσει των λέξεων κλειδιών. Οι τιμές μπορεί να είναι 0 ή 1 ανάλογα με τον λογικό τελεστή, ΚΑΙ ή Ή που χρησιμοποιήθηκε κατά την υποβολή της επερώτησης. Επίσης εδώ αποθηκεύεται και η θεματική κατηγορία στην οποία αναζητήθηκαν τα άρθρα εαν προσδιορίστηκε τέτοια κατηγορία κατά την υποβολή της επερώτησης.
- **answer**: το σύνολο των cached αποτελεσμάτων σε μορφή XML ώστε να είναι πιο αποτελεσματική η μετέπειτα επεξεργασία τους από το σύστημα. Για κάθε επιστρεφόμενο άρθρο αποθηκεύονται ορισμένες βασικές πληροφορίες όπως το αναγνωριστικό του, ο τίτλος του, η ημερομηνία κατά την οποία αποθηκεύτηκε στο σύστημα καθώς και η σχετικότητα του ώς προς την επερώτηση όπως αυτή μετρήθηκε κατά την πρώτη εκτέλεση της επερώτησης.
- **dates**: οι ημερομηνίες για τις οποίες εκτελέστηκε αρχικά η επερώτηση. Το πεδίο αυτό χρησιμοποιείται ως βασικό φίλτρο για τον έλεγχο της εγκυρότητας των cached δεδεομένων. Σε περίπτωση εμπλουτισμού των δεδομένων αυτών με πιο επίκαιρα, οι τιμές αυτών των ημερομηνιών είναι δυνατό να τροποποιηθούν.

7.1.8. user_website_category

Ο πίνακας αυτός αποθηκεύει τις πρωταρχικές επιλογές του χρήστη που αφορούν τις κατηγορίες προτίμησης των χρηστών.

- **user_id**: Το μοναδικό ξένο κλειδί που αντιπροσωπεύει τους χρήστες.
- **cat_id**: το μοναδικό ξένο κλειδί που αντιπροσωπεύει τις κατηγορίες.

7.1.9. user_website_info

Ο πίνακας αυτό χρησιμοποιείται σαν log για τις ενέργειες του χρήστη. Καταγράφει τις ημερομηνίες και την IP από την οποία έχουν πραγματοποιήσει σύνδεση οι χρήστες και βοηθά στην καλύτερη παρουσίαση των νέων άρθρων στους χρήστες αφού το σύστημα είναι σε θέση να γνωρίζει ποια άρθρα έχουν προστεθεί στο σύστημα από την τελευταία φορά που το επισκεφθήκαν οι χρήστες του συστήματος.

- **id**:Το μοναδικό αναγνωριστικό κλειδί που αφορά τις εγγραφές που γίνονται στο συγκεκριμένο πίνακα. Χρησιμοποιείται γιατί το ξένο κλειδί user_id δε μπορεί να είναι κλειδί στον συγκεκριμένο πίνακα λόγω της πληθώρας των εγγραφών χρήστη που υπάρχουν στο συγκεκριμένο πίνακα και αφορούν μεμονωμένους χρήστες.
- **user_id**: Το μοναδικό ξένο κλειδί που αντιπροσωπεύει τους χρήστες
- **user_ip**: Η IP από την οποία έχει συνδεθεί ο χρήστης
- **user_date**: Η ημερομηνία που συνδέθηκε ο χρήστης

7.1.10. user_website_keyword

Τα πεδία του πίνακα είναι τα ακόλουθα:

- **user_id**: Το μοναδικό ξένο κλειδί που αντιπροσωπεύει τους χρήστες
- **kw_id**: Το μοναδικό ξένο κλειδί που αντιπροσωπεύει τις λέξεις κλειδιά
- **rel_frequency**: Η σχετική συχνότητα που αντιπροσωπεύει κατά πόσο ο χρήστης ενδιαφέρεται για τη συγκεκριμένη λέξη κλειδί. Οι τιμές είναι θετικές και αρνητικές ενώ συνήθεις τιμές για το συγκεκριμένο πεδίο είναι -2,00 έως 2,00. Το φαινόμενο μία λέξη να ξεφεύγει από αυτά τα όρια είναι (α) ο χρήστης να μην ενδιαφέρεται καθόλου για μία λέξη κλειδί και (β) ο χρήστης να νδιαφέρεται πολύ για μία λέξη κλειδί όταν οι τιμές είναι μικρότερες του -2 και μεγαλύτερες του +2 αντίστοιχα.

7.1.11. language

Στο συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες για τις γλώσσες τις οποίες υποστηρίζει ή πρόκειται να υποστηρίξει το σύστημα μελλοντικά. Τα πεδία είναι τα ακόλουθα:

- **id**: το αναγνωριστικό της κάθε γλώσσας το οποίο αποτελεί και το πρωτεύον κλειδί του πίνακα
- **s_name**: η συντομογραφία της γλώσσας, δηλαδή οι δύο ή τρεις πρώτοι χαρακτήρες που την ταυτοποιούν μοναδικά για το χρήστη.
- **name**: το πλήρες όνομα της γλώσσας
- **flag_path**: το όνομα του αρχείου εικόνας με την οποία θα φαίνεται η γλώσσα στις σελίδες του συστήματος
- **stemming_filename & stemming_file**: το όνομα του αρχείου και το περιεχόμενο που θα χρησιμοποιεί ο stemmer για να βρίσκει τις ρίζες των λέξεων της γλώσσας.
- **stopwords_filename & stopwords_file**: το όνομα του αρχείου και το περιεχόμενο που θα χρησιμοποιεί το σύστημα για να ανακαλύπτει στα κείμενα τα stopwords τα οποία στη συνέχεια θα αφαιρούνται από το σώμα του κειμένου προκειμένου να μείνει μόνο το χρήσιμο κείμενο

7.1.12. category

Ο πίνακας αυτός περιέχει τα στοιχεία των θεματικών κατηγοριών που υπάρχουν στο σύστημα. Οι κατηγορίες προκύπτουν από τα στοιχεία που διαθέτει η βάση γνώσης του συστήματος. Τα πεδία του συγκεκριμένου πίνακα είναι τα εξής:

- **cat_id**: το μοναδικό αναγνωριστικό κλειδί για τις εγγραφές του πίνακα.
- **cat_title**: το όνομα της συγκεκριμένης κατηγορίας όπως παρουσιάζεται στις σελίδες του συστήματος
- **cat_comments**: Μικρή περιγραφή για τα στοιχεία της κάθε κατηγορίας. Προορίζεται ώστε μελλοντικά να χρησιμοποιηθεί εφόσον το σύστημα χρειαστεί μεταδεδομένα για κάθε κατηγορία.
- **cat_sub**: χρησιμοποιείται για κατηγορίες που αποτελούν υποκατηγορίες άλλων κατηγοριών δείχνοντας τα αναγνωριστικά τους (cat_id). Σε κατηγορίες όπως τα sports υπάρχουν υποκατηγορίες όπως football, basketball, tennis, motorsports.
- **sqrt_sum_sq_kws**
- **sports**: ειδικό flag που χρησιμοποιείται μόνο για τις κατηγορίες των sports.

7.1.13. user_website

στον πίνακα αυτό αποθηκεύονται πληροφορίες για κάθε χρήστη του συστήματος. Σε κάθε διαδικασία εγγραφής ή σύνδεσης χρήστη στο σύστημα χρησιμοποιείται ο συγκεκριμένος πίνακας. Τα πεδία είναι τα εξής:

- **id**: το αριθμητικό αναγνωριστικό κάθε χρήστη που είναι ταυτόχρονα και πρωτεύον κλειδί του πίνακα.
- **email**: η ηλεκτρονική διεύθυνση κάθε χρήστη. Εφόσον ο χρήστης χρησιμοποιεί τη δυνατότητα RSS του δικτυακού τόπου η ηλεκτρονική του διεύθυνση δεν είναι αναγκαία πληροφορία οστώσο στο σύστημα αποτελεί και το ψευδόνυμο του χρήστη που είναι μοναδικό και τον ταυτοποιεί.
- **passw**: ο κωδικός του χρήστη που μαζί με την ηλεκτρονική του διεύθυνση χρησιμοποιείται για την ταυτοποίηση του στο σύστημα. Για τη βελτιστοποίηση της ασφάλειας του συστήματος χρησιμοποιείται md5 (Message-Digest algorithm 5) κωδικοποίηση.
- **scrw**: πρόκειται για το πλάτος της οθόνης που χρησιμοποιεί ο χρήστης σε pixels. Χρησιμεύει έτσι ώστε να αποστέλεται το σωστό μέγεθος κειμένου στον τελικό χρήστη.
- **scrh**: Πρόκειται για το ύψος της οθόνης του χρήστη σε pixels (screen height). Χρησιμεύει στο να αποσταλεί το σωστό μέγεθος κειμένου στον τελικό χρήστη. Δεδομένου ότι η σύνηθης κύλιση σελίδες είναι προς τον κάθετο άξονα (scrolling) το ύψος της οθόνης είναι ενδεικτικό.
- **rec_date**: πρόκειται για την ημερομηνία εγγραφής του χρήστη στο σύστημα και είναι σε μορφή datatime.
- **app**: αναγνωριστικό για την εφαρμογή που χρησιμοποιεί ο χρήστης για περιήγηση στο διαδίκτυο.
- **only_rss**: flag που συμβολίζει αν ο χρήστης χρησιμοποιεί μόνο την RSS υπηρεσία του συστήματος ή και τις υπόλοιπες προσφερόμενες υπηρεσίες.

7.1.14. rss

Ο συγκεκριμένος πίνακας χρησιμοποιείται προκειμένουν να παρέχει στον mixed crawler πληροφορίες για το ποιες ιστοσελίδες πρέπει να προσπελάσει. Τα πεδία του πίνακα rss είναι τα ακόλουθα:

- **id**: είναι το αναγνωριστικό κάθε γραμμής του πίνακα και ταυτόχρονα αποτελεί και το πρωτεύον κλειδί (primary key).
- **name**: περιέχει το όνομα του κάθε rss feed. επιλέγεται έτσι ώστε να είναι σύντομο και περιγραφικό διότι είναι και η πληροφορία που εμφανίζεται και στις σελίδες του δικτυακού τόπου.
- **date**: η ημερομηνία και ακριβής ώρα στην οποία προστέθηκε το συγκεκριμένο rss στη βάση.
- **rss**: ο σύνδεσμος (link) από τον οποίο ο mixed crawler θα "διαβάζει" για να εντοπίσει τα νεότερα feeds των ειδησεογραφικών δικτυακών τόπων.

7.1.15. keywords

πρόκειται για έναν απο τους πιο σημαντικούς πίνακες της βάσης, ο οποίος περιέχει όλες τις λέξεις κλειδιά που έχουν εξαχθεί απο τα άρθρα και έχουν καταγραφεί στο μηχανισμό. Ο πίνακας περιέχει τρία πεδία:

- **kw_id**: το πρωτεύον κλειδί του πίνακα με το οποίο προσδιορίζεται μοναδικά κάθε λέξη κλειδί
- **kw_name**: η ρίζα (stem) της λέξης κλειδί.
- **lang_id**: το αναγνωριστικό της γλώσσας στην οποία ανήκει η λέξη κλειδί.

7.1.16. articles

Στον πίνακα αυτό περιέχονται όλα τα δεδομένα για άρθρα τα οποία έχουν προστεθεί στο σύστημα. Τα πεδία του πίνακα είναι τα ακόλουθα:

- **ar_id**: το πρωτεύον κλειδί του πίνακα και το μοναδικό αναγνωριστικό για κάθε άρθρο που έχει προστεθεί στη βάση.
- **ar_title**: στο πεδίο αυτό αποθηκεύεται ο τίτλος του άρθρου όπως αυτός αναγνωρίστηκε μέσω των σελιδών των rss feeds και όχι από την ανάλυση των σελίδων.
- **ar_body**: το κύριο σώμα του κειμένου ή όπως έχει ήδη αναφερθεί το Χρήσιμο Κείμενο.
- **ar_summary**: η γενική περίληψη του άρθρου όπως προκύπτει από το μηχανισμό αυτόματης εξαγωγής περίληψης. Στην περίπτωση της δυναμικής δημιουργίας περίληψης, το σύστημα τη συνθέτει σε πραγματικό χρόνο και δεν την ανακτά από τη βάση δεδομένων.
- **ar_html**: ο html κώδικά του άρθρου.
- **ar_url**: το url (Uniform Resource Locator) το οποίο οδηγεί στη σελίδα του άρθρου όπως αυτό ανακτήθηκε μέσα από το rss feed.
- **ar_date**: η ημερομηνία ανάκτησης του άρθρου. Το πεδίο αυτό είναι μορφής timestamp.
- **ar_lang**: το αναγνωριστικό της γλώσσας στην οποία είναι γραμμένο το άρθρο. Αποτελεί ξένο κλειδί (foreign key) στον πίνακα language.
- **rss_id**: το αναγνωριστικό του rss feed από το οποίο προήλθε το άρθρο. Αποτελεί ξένο κλειδί στον πίνακα rss.
- **flag_preprocessed**: πρόκειται για μια μεταβλητή αναγνώρισης για να εντοπίσουμε ποια άρθρα έχουν προεπεξεργαστεί και ποια όχι προκειμένου ο μηχανισμός κατηγοριοποίησης να μπορεί να αναγνωρίσει ποια άρθρα θα πρέπει να επεξεργαστούν.
- **flag_categorized**: πρόκειται για μια μεταβλητή αναγνώρισης για να εντοπίσουμε ποια άρθρα έχουν κατηγοριοποιηθεί και ποια όχι προκειμένου ο μηχανισμός κατηγοριοποίησης να μπορεί να αναγνωρίσει ποια άρθρα θα πρέπει να κατηγοριοποιηθούν.
- **flag_summarized**: πρόκειται για μια μεταβλητή αναγνώρισης για να εντοπίσουμε ποια άρθρα έχουν περάση από το μηχανισμό αυτόματης εξαγωγής περίληψης και ποια όχι. Με αυτή την πληροφορία, καθίσταται δυνατό για τον μηχανισμό αυτόματης εξαγωγής περίληψης να γνωρίζει για ποια άρθρα θα πρέπει να εκτελεστεί διαδικασία εξαγωγής περίληψης.
- **flag_training**: πρόκειται για μια μεταβλητή αναγνωρίσης ούτως ώστε το σύστημα να μπορεί να εντοπίσει ποια από τα άρθρα έχουν περάσει από το μηχανισμό training του συστήματος.

7.1.17. user_website_reading

Ο πίνακας χρησιμεύει για να καταμετρήσουμε το χρόνο που κάθε χρήστης περνά διαβάζοντας ένα άρθρο. Το χρησιμοποιούμε σαν μετρική προκειμένου να καταμετρήσουμε το ενδιαφέρον του χρήστη για συγκεκριμένα κείμενα έτσι ώστε στη συνέχεια να βελτιώσουμε και να κάνουμε περισσότερο προσωποποιημένα τα αποτελεσματα της αναζήτησης άρθρων στη βάση.

- **ar_id**: αναγνωριστικό για το άρθρο. Αποτελεί ξένο κλειδί στον πίνακα articles.
- **user_id**: αναγνωριστικό για τον χρήστη. Αποτελεί ξένο κλειδί στον πίνακα user_website.
- **total_time**: Ο συνολικός χρόνος που έχει σπαταλήσει ο χρήστης στο συγκεκριμένο άρθρο. Αποτελεί μέτρο για το ενδιαφέρον του χρήστη στο άρθρο.
- **hits**: αριθμός των διακεκριμένων φορών που έχει ανασύρει ο χρήστης το συγκεκριμένο άρθρο. Αποτελεί μέτρο για το ενδιαφέρον του χρήστη στο άρθρο.

- **last_date**: η ημερομηνία που ο χρήστης ανέσυρε τελευταία φορά το συγκεκριμένο άρθρο. Είναι σε μορφή datetime.

8

ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό περιγράφεται ο τρόπος ανάπτυξης του συστήματος συνολικά αλλά και κάθε μηχανισμού ξεχωριστά. Γίνεται ανάλυση των αλγορίθμων που χρησιμοποιούνται αλλά και ο τρόπος λειτουργίας κάθε υποσυστήματος.

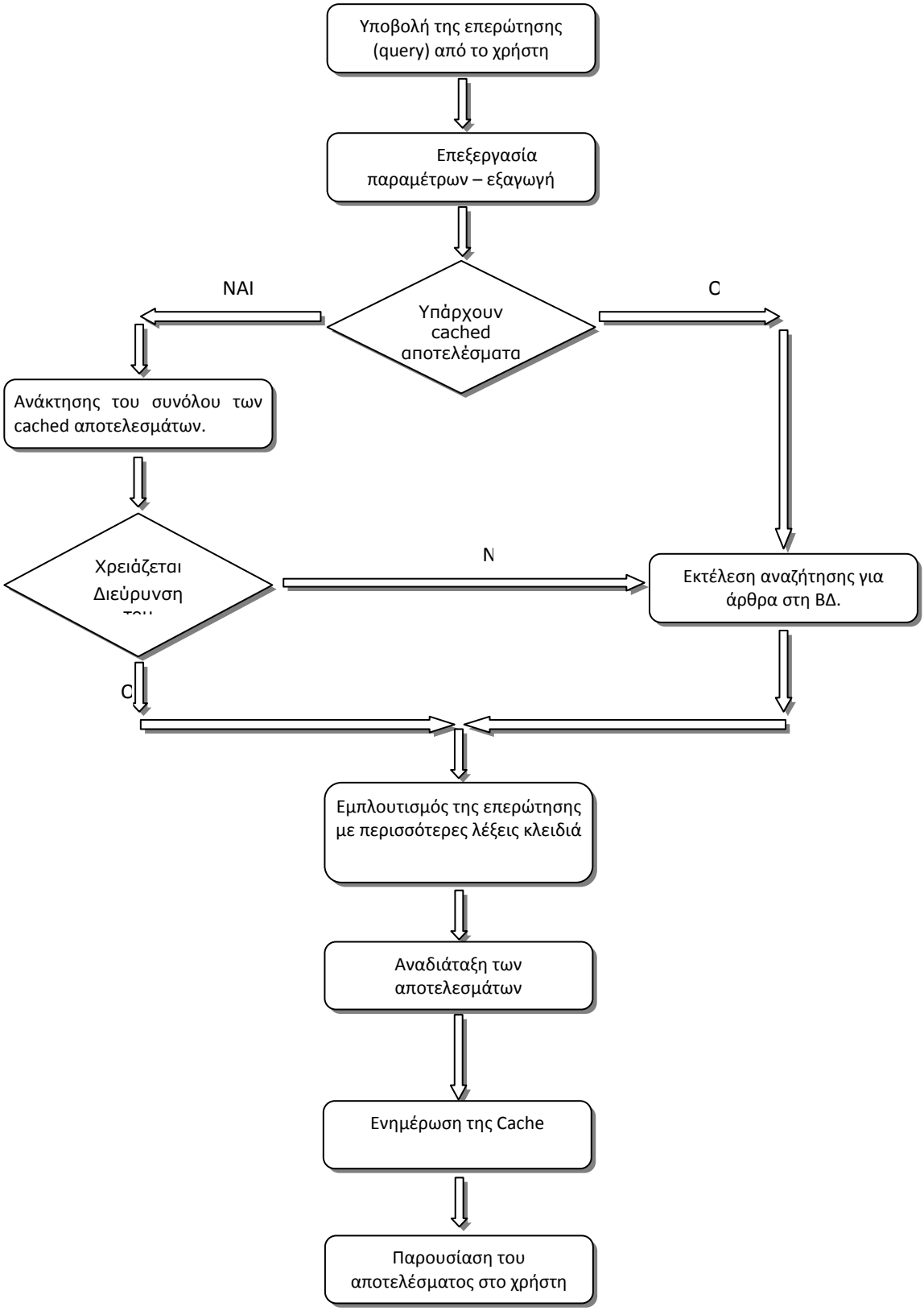
8. ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Μέχρι αυτή τη στιγμή έχουμε αναφερθεί στα βασικά συστήματα που αποτελούν τη βάση για το σύστημά μας αλλά και στους αλγορίθμους που χρησιμοποιούμε προκειμένου να υλοποιήσουμε το κάθε υποσύστημα. Σε αυτό το κεφάλαιο, θα εστιάσουμε την προσοχή μας στην υλοποίηση του συστήματος προσωποποιημένης αναζήτησης του χρήστη στη βάση δεδομένων που περιέχει τα άρθρα. Δεδομένου ότι οι γραμμές κώδικα που γράφτηκαν συνολικά για την κατασκευή του συστήματος είναι πολλές για να παρουσιαστούν συνολικά, θα εστιάσουμε την προσοχή μας στα πιο σημαντικά στοιχεία καθώς και στις τεχνικές με τις οποίες υλοποιήθηκε ο αλγόριθμος σε κάθε σημείο.

8.1. Αλγοριθμικά Θέματα

Στο επόμενο σχήμα παρουσιάζονται τα βασικά στάδια της εκτέλεσης του αλγόριθμου προσωποποιημένης αναζήτησης. Στο πρώτο στάδιο, ο χρήστης υποβάλλει την επερώτηση του. Το σύστημα αντιστοιχίζει τις λέξεις κλειδιά που έδωσε ο χρήστης σε λέξεις που υπάρχουν ήδη αποθηκευμένες στη ΒΔ από τη διαδικασία κατηγοριοποίησης των άρθρων που φτάνουν καθημερινά στο σύστημα. Στο επόμενο στάδιο και προτού γίνει εκκίνηση της διαδικασίας αναζήτησης άρθρων στη ΒΔ, ελέγχεται η cache μνήμη του συστήματος που περιέχει αποτελέσματα από παρελθοντικές επερωτήσεις που υπέβαλλαν χρήστες του συστήματος. Εάν εντοπιστεί επερώτηση στην cache που να έχει τις ίδιες παραμέτρους με την τρέχουσα επερώτηση και η οποία να έχει υποβληθεί από τον ίδιο χρήστη στο πρόσφατο παρελθόν τότε τα άρθρα-αποτελέσματα ανασύρονται γρήγορα από την cache χωρίς να έχει γίνει καμία αναζήτηση. Ορισμένες φορές, είναι απαραίτητο να εκτελεστεί μια περιορισμένη αναζήτηση αν το σύστημα διαπιστώσει ότι τα cached αποτελέσματα δεν επαρκούν για να καλύψουν όλο το χρονικό διάστημα για το οποίο ο χρήστης υπόβαλλε την επερώτηση. Αφού ανακτηθούν όλα τα άρθρα που απαντούν στην επερώτηση του χρήστη ξεκινάει η φάση της ταξινόμησης και επιλογής τους έτσι ώστε η διαδικασία να είναι προσωποποιημένη στο προφίλ και στις προτημίσεις του χρήστη. Από τις λέξεις-κλειδιά της επερώτησης με μια πολύπλοκη διαδικασία που θα παρουσιασθεί και θα αναλυθεί στη συνέχεια του κεφαλαίου, βρίσκονται άλλες λέξεις-κλειδιά που ήδη υπάρχουν στο σύστημα και που χαρακτηρίζονται ως σχετικές των λέξεων που έδωσε αρχικά ο χρήστης. Οι νέες αυτές λέξεις δεν χρησιμοποιούνται καθόλου στην διαδικασία ανάσυρσης άρθρων από τη βάση δεδομένων του συστήματος αλλά ο σκοπός τους είναι να βελτιώσουν την σειρά με την οποία θα εμφανιστούν τα άρθρα στον ίδιο το χρήστη. Αυτό για μάς αποτελεί μια αναβάθμιση της ποιότητας αναζήτησης, μιας και όσο πιο ψηλά στη λίστα των άρθρων του αποτελέσματος βρίσκονται τα άρθρα που ουσιαστικά επιθυμούσε ο χρήστης τότε μεγαλύτερη είναι και η επιτυχία της αναζήτησης καθώς και η απότελεσματικότητα του μηχανισμού μας. Με βάση αυτήν την «εμπλουτισμένη» επερώτηση που προκύπτει γίνεται η ταξινόμηση των άρθρων και η παρουσίασή τους στον τελικό χρήστη. Μετά από κάθε αναζήτηση ακολουθεί ενημέρωση της cache είτε υπό μορφή ενημέρωσης των cached αποτελεσμάτων για τις ήδη υπάρχουσες επερωτήσεις ή υπό μορφή προσθήκης των αποτελεσμάτων για νέες επερωτήσεις που δεν υπήρχαν στην cache.

Στη συνέχεια θα εξετασθεί λεπτομεριακά κάθε βήμα της διαδικασίας που περιγράφηκε με το βάρος να δίνεται στην διαδικασία της αναζήτησης καθώς και την προσωποποίησης μέσω της αναδιάταξης των άρθρων του αποτελέσματος. Για το λόγο αυτό, όπου κρίνεται αναγκαίο θα υπάρχουν διαφωτιστικά σχήματα, διαγράμματα ροής, κομμάτια κώδικα απευθείας μέσα από το σύστημα καθώς και ψευδοκώδικα για την απλούστευση της παρουσίασης ορισμένων σημείων.



Εικόνα 10. Σύστημα προσωποποίησης αρθρων αναζήτησης

8.2. Υποβολή και Επεξεργασία της επερώτησης του Χρήστη

Στην παράγραφο αυτή παρουσιάζονται τα πρώτα βήματα συλλογής και επεξεργασίας της επερώτησης του χρήστη. Περιγράφονται οι προσφερόμενες

παράμετροι καθώς και τα βήματα έως την δημιουργία της λίστας των λέξεων κλειδιών που θα τροφοδοτήσουν τον αλγόριθμο της αναζήτησης.

8.2.1. Παράμετροι της αναζήτησης

Το σύστημα αναζήτησης που υλοποιήθηκε δεν υλοποιεί μια απλή αναζήτηση με βάση μόνο κάποιες λέξεις κλειδιά αλλά προσφέρεται η δυνατότητα στο χρήστη να προσδιορίσει και άλλες παραμέτρους προκειμένου να περιγράψει με καλύτερη λεπτομέρια τα επιθυμητά από αυτόν αποτελέσματα. Οι παράμετροι οι οποίες καθορίζονται από το χρήστη στη φόρμα αναζήτησης χρησιμοποιούνται σε διάφορα στάδια της ανάκτησης των άρθρων του τελικού αποτελέσματος όπως κατά την αναζήτηση ήδη αποθηκευμένων αποτελεσμάτων στην μνήμη cache, στον εμπλουτισμό της επερώτησης του χρήστη με επιπλέον λέξεις που χαρακτηρίζονται από το σύστημα ως συναφείς καθώς και στην τελική αναδιάταξη των άρθρων του αποτελέσματος της αναζήτησης. Σε ένα πραγματικό σύστημα μια αναζήτηση θα μπορούσε να παραμετροποιείται από πολλές παραμέτρους ωστόσο για τις ανάγκες την ανάπτυξης και αξιολόγησης του συστήματος μας χρησιμοποιήσαμε τα ακόλουθα πεδία που μπορεί να ρυθμίσει ο χρήστης ούτως ώστε να προσδιορίσει το περιεχόμενο των άρθρων που επιθυμεί.

8.2.1.1. Λέξεις Κλειδιά

Αναφέρονται και ως όροι αναζήτησης (search terms) ή κωδικολέξεις (keywords). Αποτελούν την πιο κοινή και σημαντική παράμετρο για κάθε σύστημα αναζήτησης στο Διαδίκτυο. Εδώ ο χρήστης παρέχοντας λέξεις ή προτάσεις προσδιορίζει το θέμα γύρω από το οποίο επιθυμεί να εκτελεστεί η αναζήτηση των άρθρων.

8.2.1.2. Λογικός Τελεστής

Χρησιμοποιείται για να περιγράψει το πρότυπο με το οποίο θα γίνει η αναζήτηση βάσει των λέξεων-κλειδιών πάνω στα άρθρα της βάσης. Στο σύστημα μας δίνεται η δυνατότητα να χρησιμοποιηθεί ο λογικός τελεστής της σύζευξης (λογικό ΚΑΙ) ή ο λογικός τελεστής της διάζευξης (λογικό 'Η). Όταν ο χρήστης επιλέγει το «ΚΑΙ» τότε ένα άρθρο ταιριάζει με την επερώτηση του χρήστη μόνο εφόσον περιέχει όλες τις λέξεις κλειδιά που έδωσε ο χρήστης. Αντίστοιχα, όταν ο χρήστης επιλέγει το λογικό «'Η», τότε ένα άρθρο επιστρέφεται αν περιέχει οποιαδήποτε από τις λέξεις-κλειδιά του χρήστη. Προφανώς στην περίπτωση επιλογής του τελεστή «'Η» αναμένονται τουλάχιστον όσα αποτελέσματα θα επιστρέφονταν με τον τελεστή «ΚΑΙ».

8.2.1.3. Θεματική Ενότητα

ο χρήστης μπορεί να επιλέξει οποιδήποτε από τις κατηγορίες που έχουν οριστεί για το σύστημα και οι οποιές είναι οι εξής: Business (Επιχειρήσεις), Entertainment (Ψυχαγωγία), Health (Υγεία), Politics (Πολιτική), Science (Επιστήμες), Sports (Αθλητισμός), Education (Εκπαίδευση), Nature (Φύση & Περιβάλλον), Technology (Τεχνολογία). Η επιλογή αυτή του χρήστη είναι ιδιαίτερα σημαντική όπως θα αναλύσουμε και στη συνέχεια του κεφαλαίου για τον αλγόριθμο με τον οποίο θα εμπλουτιστεί η επερώτηση του χρήστη με νέες λέξεις-κλειδιά για την επίτευξη της προσωποποίησης του αποτελέσματος στα ενδιαφέροντα του χρήστη.

8.2.1.4. Χρονική Περίοδος

επίσης αρκετά σημαντική παράμετρος που προσδιορίζει το χρονικό διάστημα μέσα στο οποίο ο χρήστης επιθυμεί να ανήκουν τα άρθρα που θα επιστρέψει η αναζήτηση. Η παράμετρος αυτή εκτός της ευνόητης λειτουργικότητας σε αυτή καθ' αυτήν την διαδικασία αναζήτησης αρθρων στη βάση δεδομένων, αξιοποιείται και κατά την διεξαγωγή της αναζήτησης για ήδη υπάρχοντα cached αποτελέσματα από

παρόμοιες επερωτήσεις που υποβλήθηκαν από το χρήστη σε πρόσφατες επισκέψεις στο σύστημα. Βάσει της χρονικής περιόδου που έδωσε ο χρήστης είναι πιθανό να μην εκτελεστεί καθόλου αναζήτησης στου πίνακες με τα άρθρα, αλλά απλά να ανακτηθούν απευθείας τα cached αποτελέσματα σε πολύ μικρό χρόνο.

8.2.2. Προεπεξεργασία της Επερώτησης

Αφού ο χρήστης υποβάλει την επερώτηση ακολουθεί η φάση της επεξεργασίας της από το σύστημα προκειμένουν να προετοιμαστούν οι είδοδοι για το σύστημα αναζήτησης. Στη φάση αυτή λαμβάνουν χώρα οι ακόλουθες διαδικασιες:

1. Αφαίρεση των σημείων στίξης και των αριθμών: Τα σημεία στίξης (punctuation) στην πρόταση επερώτησης (query string) του χρήστη δεν προσδίδουν σημασιολογική πληροφορία σε αυτό και συνεπώς δεν είναι απαραίτητο να λάβουν μέρος και να υπολογιστούν στην διαδικασία αναζήτησης. Ο αλγόριθμος αφαιρεί κάθε σημείο στίξης από την πρόταση επερώτησης του χρήστη.
2. Αφαίρεση των stopwords: τα άρθρα, οι αντωνυμίες, οι προθέσεις καθώς επίσης και πολύ ευρέως χρησιμοποιούμενα ρήματα (is, have, are κτλ) και επίθετα επίσης δεν προσδίδουν σημασιολογική πληροφορία στις λέξεις-κλειδιά που απαρτίζουν την επερώτηση του χρήστη άρα μπορούν να αφαιρεθούν. Ο αλγόριθμος φιλτράρει κάθε λέξη κλειδί που έδωσε ο χρήστης και τις συγκρίνει με λέξεις τις οποίες ταυτοποιήσαμε ως stop words.
3. Μετατροπή των κεφαλαίων σε πεζά: η διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων αμελητaία μόνο πληροφορία μπορεί να δώσει για την πρόταση επερώτησης του χρήστη. Για το λόγο αυτό και για την ομοιομορφία των προς επεξεργασία λέξεων, όλα τα κεφαλαία γράμματα μετασχηματίζονται από το αλγόριθμο σε μικρά.
4. Εξαγωγή των ριζών των λέξεων κλειδιών (Stemming): Στη βάση δεδομένων του συστήματος οι λέξεις των άρθρων που έχουν εισαχθεί από διάφορα rss feeds αποθηκεύονται στο σύστημα αφού πρώτα επεξεργαστούν από αλγόριθμο που εκτελεί εξαγωγή της ρίζας τους. Αυτό γίνεται για να μειωθεί το πλήθος των λέξεων-κλειδιών που αποθηκεύονται ώστε να είναι πιο αποτελεσματική η μετέπειτα αναζήτηση. Αφετέρου, η ρίζα κρατάει το νόημα της αρχικής λέξης και επιτρέπει την διασταύρωση του άρθρου μέ άλλες λέξεις κλειδιά που προέρχονται από την ίδια ρίζα (π.χ. product, produce, production) αλλά δεν βρίσκονται στην ίδια μορφή. Ειδικά για ορισμένες γλώσσες για τις οποίες το τυπολογικό της γραμματικής τους ορίζεις μεγάλο πλήθος καταλήξεων, όπως για παράδειγμα τα Ελληνικά ή τα Γερμανικά) είναι δυνατό να μειωθεί δραστικά ο χώρος που χρειάζεται για την αποθήκευση των εξαγόμενων από τα άρθρα λέξεων και συνεπώς να επιταχύνεται η διαδικασία της αναζήτησης. Για τις ανάγκες του συστήματος μας χρησιμοποιήσαμε τόσο κατά την αρχική επεξεργασία των άρθρων που φτάνουν στο σύστημα όσο και κατά τη διεξεγωγή της αναζήσησης πάνω σε αυτά τον Porter Stemming αλγόριθμο τον οποίο και υλοποιήσαμε για τις ανάγκες του συστήματος.
5. Απομάκρυνση διπλών λέξεων (duplicate words elimination): προφανώς, αν στην πρόταση επερώτησης του χρήστη υπάρχει η ίδια λέξη ή παράγωγα της ίδιας ρίζας πάνω από μία φορά χρειάζεται να απομακρυνθούν αφού ούτως ή αλλως δεν προσδίδουν κάποια επιπλέον λειτουργικότητα στην αναζήτηση.

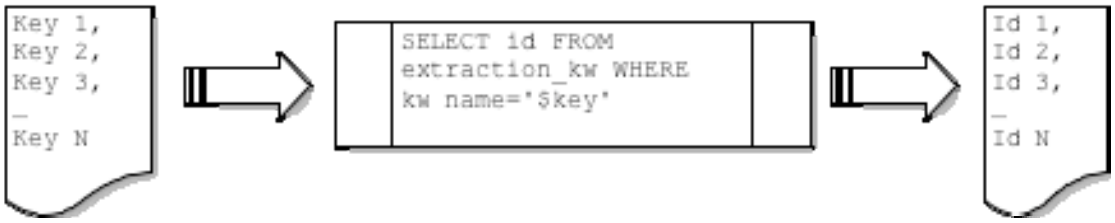
Τελικά, ακολουθώντας αυτά τα βήματα προκύπτει μια λίστα από λέξεις-κλειδιά σε μορφή ρίζας που θα τροφοδοτήσουν το κύριο κομμάτι του αλγόριθμου αναζήτησης το οποίο θα παρουσιαστεί στην επόμενη παράγραφο.

8.3. Αλγόριθμος Αναζήτησης

Στην παράγραφο αυτή, θα γίνει παρουσίαση των πιο βασικών σημείων του αλγόριθμου αναζήτησης άρθρων στην βάση δεμένων.

8.3.1. Αντιστοίχιση Κωδικολέξεων και Απόδοση Βαρών

Για κάθε λέξη-κλειδί που προέκυψε από την επεξεργασία της επερώτησης του χρήστη πρέπει να αναζητηθεί το αναγνωριστικό μέσα στον πίνακα extraction_kw της βάσης δεδομένων. Αυτό είναι απαραίτητο καθώς όλοι οι πίνακες της βάσης που συσχετίζουν τα άρθρα και τις θεματικές κατηγορίες με τις λέξεις-κλειδιά, οι οποίοι θα αξιοποιηθούν για την αναζήτηση, χρησιμοποιούν τα αναγνωριστικά των λέξεων και όχι τα ονόματα τους. Όπως έχει προαναφερθεί οι λέξεις-κλειδιά που υπάρχουν αποθηκευμένες στη βάση δεδομένων προκύπτουν από την προεπεξεργασία των άρθρων καθώς αυτά φτάνουν μέσω rss feeds στο σύστημα. Λόγω του μεγάλου όγκου των άρθρων που καταφθάνουν, κατηγοριοποιούνται και αποθηκεύονται καθημερινά, τα οποία χαρακτηρίζονται από μεγάλη ποικιλία ως προς το θεματικό τους περιεχόμενο και χάρη στην επιλογή να αποθηκεύονται οι λέξεις των άρθρων ως ρίζες, το πλήθος των λέξεων που υπάρχουν στη βάση είναι επαρκές ώστε με ικανοποιητικά μεγάλη πιθανότητα για κάθε λέξη που προσδιορίζει ο χρήστης στην πρόταση της επερώτησης να μπορεί να βρεθεί μια αντίστοιχη ρίζα στον πίνακα extraction_kw. Η διαδικασία παρουσιάζεται στο επόμενο σχήμα.



Εικόνα 11. αντιστοίχιση των λέξεων-κλειδιών σε κωδικολέξεις της ΒΔ

Αφού διεξαχθεί η αντιστοίχιση των λέξεων-κλειδιών στα αναγνωριστικά τους (IDs) ο αλγόριθμος συνεχίζει με την απόδοση συντελεστών βάρους στις λέξεις κλειδιά.

Οι συντελεστές βάρους για τις λέξεις κλειδιά που έδωσε στη φόρμα αναζήτησης ο χρήστης θα χρησιμοποιηθούν στη συνέχεια για την αναδιάταξη των άρθρων που θα επιστρέψει ο αλγόριθμος. Στον αλγόριθμο μας επιλέξαμε να δίνουμε στις κωδικολέξεις βάρη ανάλογα με τη σειρά με την οποία τις έγραψε ο χρήστης στην πρόταση της επερώτησης. Θεωρώντας ότι σε κάθε αναζήτηση φροντίζουμε να παρέχουμε στην μηχανή αναζήτησης τις λέξεις με τη σειρά που τις θεωρούμε σημαντικές για τα αποτελέσματα που θέλουμε να μας επιστραφούν χρησιμοποιήσαμε το ακόλουθο αλγόριθμο για την απόδοση βαρών:

```
W = 0.1
Foreach (key in Keyword_IDS) repeat
{
  Weight[i] = MAX(W, 0.06)
  W ← W – 0.01
}
```

Αλγόριθμος 1: απόδοση συντελεστών βάρους στις κωδικολέξεις

Στην πρώτη λέξη αποδώσαμε βάρος 0.1 και σε κάθε επόμενη λέξη έως και την πέμπτη κατά σειρά το βάρος μειώνεται κατά 0.01. Από την πέμπτη λέξη και μετά το βάρος παραμένει σταθερά στο 0.06. Αυτό θα έχει ως αποτέλεσμα η πρώτη κατά σειρά λέξη-κλειδί που παρείχε ο χρήστης να προσμετρηθεί περισσότερο στην όταν θα εκτελεστεί η αναδιάταξη των άρθρων στο τελικό αποτέλεσμα. Όπως θα δούμε και στη συνέχεια, οι συντελεστές βάρους για κάθε κωδικολέξη κληρονομούνται και στα «παιδιά» τους όταν θα γίνει ο εμπλουτισμός της

επερώτησης του χρήστη κατά τη διαδικασία προσωποποίησης του τελικού αποτελέσματος.

8.3.2. Ανάκτηση των άρθρων

Κατά την κατασκευή του επερωτήματος στη βάση για την ανάκτηση των άρθρων που περιέχουν τις λέξεις κλειδιά που έδωσε ο χρήστης στη φόρμα αναζήτησης λαμβάνονται υπόψιν όλες οι παράμετροι (λογικός τελεστής, θεματική κατηγορία, χρονική περίοδος). Από την βάση δεδομένων χρησιμοποιούνται οι ακόλουθοι πίνακες με τον τρόπο που περιγράφονται:

- **articles**: περιλαμβάνει όλα τα στοιχεία για τα άρθρα που είναι αποθηκευμένα στο σύστημα. Από αυτόν τον πίνακα μας ενδιαφέρουν τα πεδιά ar_id, ar_title και ar_date καθώς είναι αυτά που θα συμπεριλάβουμε στο τελικό αποτέλεσμα και τα οποία θα ενημερώσουν τον cache πίνακα της βάσης δεδομένων. Επιπλέον, το πεδίο ar_date χρησιμοποιείται για φιλτράρισμα των άρθρων σε περίπτωση που ο χρήστης έχει επιλέξει χρονική περίοδο στη φόρμα αναζήτησης.
- **extraction_kw2ar**: σε αυτόν τον πίνακα αποθηκεύονται οι σχέσεις μεταξύ των άρθρων και των λέξεων κλειδιών που υπάρχουν μέσα σε αυτά. Για κάθε άρθρο, όπως φαίνεται και στον ακόλουθο πίνακα μπορούν να υπάρχουν πολλαπλές καταχωρήσεις που το συνδέουν με τις κωδικολέξεις που εντοπίστηκαν σε αυτό κατά την προεπεξεργασία του και την κατηγοριοποίηση του όταν αποθηκεύτηκε στο σύστημα. Επίπρόσθετα, για κάθε κωδικολέξη που συνδέεται με ένα άρθρο έχει καταγραφεί η σχετική και η απόλυτη συχνότητα εμφάνισης της στο άρθρο. Η απόλυτη συχνότητα μιας κωδικολέξης σε ένα άρθρο ορίζεται ως το πλήθος των εμφανίσεων της μέσα σε αυτό ενώ η σχετική συχνότητα μιας κωδικολέξης ορίζεται ως το πηλίκο της απόλυτης συχνότητας της προς το άθροισμα των απολύτων συχνοτήτων όλων των κωδικολέξεων που υπάρχουν στο άρθρο μαζί με αυτή. Προφανώς μέτρο του κατά πόσο συχνά εμφανίζεται μια κωδικολέξη σε ένα άρθρο αποτελεί μόνο η σχετική συχνότητα της μέσα σε αυτό, αφού λαμβάνει υπόψιν και τις συχνότητες των υπολοίπων κωδικολέξεων.

Κωδικός Άρθρου	Κωδικός Λέξης	Σχετική Συχνότητα	Απόλυτη Συχνότητα
2929	570	0,0035337	1
2929	613	0,0141343	4
2929	650	0,0106007	3
2929	788	0,00706714	2
2929	791	0,0106007	3
2929	972	0,0176678	5
2929	1010	0,00353371	1
2929	1061	0,00706716	2

Πίνακας 1: Συχνότητες κωδικολέξεων στα άρθρα της βάσης

Σε αυτόν τον πίνακα είναι που αναζητούμε την συσχέτιση των κωδικολέξεων που παρέχει ο χρήστης με τα άρθρα της βάσης. Στη περίπτωση που ο λογικός τελεστής που επιλέχθηκε είναι το λογικό «Η», η επιλογή του άρθρου για να επιστραφεί ως τελικό αποτέλεσμα στο χρήστη γίνεται αν υπάρχει τουλάχιστον μια από τις κωδικολέξεις μέσα σε αυτό. Αυτή η περίπτωση μπορεί να υλοποιηθεί σχετικά εύκολα με ένα INNER JOIN του πίνακα articles και του πίνακα extraction_kw2ar.

Ωστόσο στην περίπτωση που ο λογικός τελεστής που επιλέχθηκε είναι το «ΚΑΙ» τότε θα πρέπει να εκτελείται LEFT JOIN του συγκεκριμένου πίνακα με τον

εαυτό του για τις κωδικολέξεις που έδωσε ο χρήστης. Ένα τέτοια LEFT JOIN για δύο κωδικολέξεις με αναγνωριστικά ID1 και ID2 είναι το ακόλουθο:

```
SELECT a1.ar_id, ar.ar_title
FROM articles ar, extraction_kw2ar a1
LEFT JOIN extraction_kw2ar a2 ON a1.ar_id=a2.ar_id
WHERE a1.kw_id=ID1 AND a2.kw_id=ID2 AND a1.ar_id=ar.ar_id
```

Αλγόριθμος 2: Επερώτηση για την ανάκτηση των άρθρων

Στην περίπτωση που οι κωδικολέξεις που εισήχθησαν στην πρόταση της επερώτησης είναι παραπάνω από δύο, το query στη βάση δεδομένων γίνεται αρκετά πιο πολύπλοκο διότι περιέχει παραπάνω του ενός LEFT JOINs του πίνακα extraction_kw2ar με τον εαυτό του.

- article2category: στην περίπτωση που ο χρήστης επίλεξε θεματική κατηγορία για την αναζήτηση του στην αρχική φόρμα αναζήτησης, χρησιμοποιείται και αυτός ο πίνακας ο οποίος συσχετίζει κάθε άρθρο που είναι αποθηκευμένο στο σύστημα με τις θεματικές κατηγορίες τις οποίες έχουμε ορίσει κατά την κατηγοριοποίηση των άρθρων. Στον επόμενο πίνακα παρουσιάζουμε ένα μέρος του πίνακα στο οποίο φαίνεται η συσχέτιση ενός άρθρου με κάθε κατηγορία του συστήματος.

Κωδικός άρθρου	Κωδικός Κατηγορίας	Συχνότητα
36728	1	0,0855691
36728	2	0,0454409
36728	3	0,0127697
36728	4	0,0151244
36728	5	0,0327046
36728	6	0,0160101
36728	7	0,00480065
36728	9	0,048447
36728	10	0,0087052
36728	11	0,00584141
36728	12	0,00406884

Πίνακας 2: Συχνότητας θεματικών ενοτήτων για ένα άρθρο.

Όπως ήταν αναμενόμενο για κάθε άρθρο σε αυτόν τον πίνακα υπάρχουν ακριβώς δώδεκα καταχωρήσεις, δηλαδή τόσες όσες και οι θεματικές κατηγορίες του συστήματος. Όπως είπαμε, αυτός ο πίνακας χρησιμοποιείται μόνο στην περίπτωση που ο χρήστης επέλεξε θεματική ενότητα για τα άρθρα που επιθυμεί να δει οπότε σε αυτή την περίπτωση ο αλγόριθμος αναζήτησης λειτουργεί ως εξής: αν κατά την αναζήτηση των προσδιορισμένων λέξεων κλειδιών στο άρθρο βρεθούν όλες (στην περίπτωση που ως λογικός τελεστής χρησιμοποιήθηκε είναι το «ΚΑΙ») ή ένα υποσύνολό τους (στην περίπτωση που ως λογικός τελεστής επιλέχθηκε το «Ή») τότε το άρθρο ανακτάται και αποθηκεύεται στην λίστα των άρθρων του αποτελέσματος που θα επιστραφεί στον τελικό χρήστη μόνο αν η κατηγορία που προσδιορίστηκε απο αυτόν είναι η κατηγορία η οποία εμφανίζεται με τη μεγαλύτερη συχνότητα στο άρθρο. Εν ολίγοις για κάθε άρθρο που ταιριάζει με τις λέξεις κλειδιά εκτελείται το ακόλουθο επερώτημα στη βάση δεδομένων:

```
SELECT cat_id, MAX(frequency)
FROM article2category
```

```
WHERE ar_id=ARTICLE_ID
```

Αλγόριθμος 3: Εύρεση της κατηγορίας με τη μεγαλύτερη συχνότητα για ένα άρθρο

Το αποτέλεσμα του επερωτήματος αποφασίζει ουσιαστικά αν το άρθρο «ανήκει» στη συγκεκριμένη κατηγορία. Ωστόσο, μπορεί να υπάρχουν άρθρα τα οποία να ανήκουν το ίδιο ή σχεδόν το ίδιο σε παραπάνω από μια κατηγορίες. Αυτό συμβαίνει προφανώς αν έχουν την ίδια ή περίπου την ίδια αντίστοιχα συχνότητα σε παραπάνω από μια κατηγορίες. Θα μπορούσε τότε να υιοθετηθεί μια παραλλαγή της υλοποίησης του αλγόριθμου η οποία να ελέγχει αν το άρθρο ανήκει με μεγάλη συσχέτιση σε μια κατηγορία. Αυτό είναι εφικτό εαν θεωρήσουμε ότι θα πρέπει η συχνότητα της κατηγορίας να υπερβαίνει ένα κατώφλι (π.χ. 0.5) που την κάθιστά θεματική κατηγορία του άρθρου.

Από την διαδικασία της ανάκτησης των άρθρων που απαντούν στην επερώτηση που υπέβαλλε ο χρήστης τα δεδομένα που συγκρατούμε για τις επόμενες φάσεις του αλγόριθμου είναι τα ακόλουθα:

- το αναγνωριστικό του άρθρου (article_id)
- τον τίτλο του άρθρου (ar_title)
- την ημερομηνία έκδοσης του άρθρου (ar_date)
- την απόλυτη συχνότητα εμφάνισης καθε κωδικολέξης μέσα στο άρθρο (abs_frequency). Αυτή θα χρησιμοποιηθεί στη συνέχεια για τον υπολογισμό του συντελεστή συσχέτισης του άρθρου (relevance factor) προς την επερώτηση του χρήστη.

8.3.3. Εμπλουτισμός Επερώτησης

Η διαδικασία του εμπλουτισμού της επερώτησης (query enrichment) είναι ουσιαστικής σημασίας για την αποτελεσματικότητα της προσωποποιημένης αναζήτησης αφού εγγυάται ότι τα τελικά αποτελέσματα που θα παρουσιαστούν στο χρήστη είναι σχετικά με το επερώτημα που τέθηκε. Ο στόχος του εμπλουτισμού της επερώτησης με επιπλέον κωδικολέξεις είναι να ανακτήσουμε άρθρα τα οποία είναι ακόμα πιο σχετικά με την κατηγορία καθώς και τις κωδικολέξεις τις οποίες παρέχει ο χρήστης.

8.3.3.1. Παράγοντας Αντιπροσωπευτικότητας

Για να επιτύχουμε τον εμπλουτισμό της επερώτησης, ορίζουμε για τον αλγόριθμο μας ένα ειδικό συντελεστή τον οποίο και ονομάζουμε παράγοντα αντιπροσωπευτικότητας ή σχετικότητας (relativity factor ή representativity factor). Ο αριθμός αυτός καθορίζει το ποσοστό το κατά πόσο είναι πιθανός ο εμπλουτισμός μιας επερώτησης με επιπλέον λέξεις κλειδιά. Όπως θα διαπιστώσουμε και στο επόμενο κεφάλαιο, όπου διεξάγονται τα πειράματα για την αξιολόγηση του αλγορίθμου και του συστήματός μας, ο παράγοντας αντιπροσωπευτικότητας μπορεί να παίξει σε αρκετές περιπτώσεις ιδιαίτερα σημαντικό ρόλο στην σειρά με την οποία εμφανίζονται τα άρθρα στον τελικό χρήστη. Αλλά τι καθορίζει ο αριθμός αυτός; Για τον αλγόριθμό μας, παράγοντας σχετικότητας με τιμή R σημαίνει ότι μια κωδικολέξη K θεωρείται ώς αντιπροσωπευτική μιας κατηγορίας C μόνο αν η απόλυτη συχνότητα εμφάνισης της στην κατηγορία C είναι R τουλάχιστον φορές μεγαλύτερη συγκρινόμενη με την απόλυτη συχνότητα εμφάνισης αυτής της κωδικολέξης στην θεματική κατηγορία στην οποία έχει τη δεύτερη μεγαλύτερη απόλυτη συχνότητα εμφάνισης. Υψηλότερες τιμές του παράγοντα αντιπροσωπευτικότητας μειώνουν την πιθανότητα οι κωδικολέξεις που επέλεξε ο χρήστης να θεωρηθούν ως αντιπροσωπευτικές μιας κατηγορίας, ενώ αν επιλέγουμε χαμηλότερες τιμές αυξάνουν την πιθανότητα οι κωδικολέξεις του χρήστη να θεωρηθούν αντιπροσωπευτικές και ως εκ τούτου το σύστημα να εμπλουτίσει την λίστα των όρων αναζήτησης με περισσότερες κωδικολέξεις. Για να γίνει καλύτερα κατανοητή η λειτουργικότητα του παράγοντα

αντιπροσωπευτικότητας ας εξετάσουμε τον επόμενο πίνακα όπου φαίνεται η απόλυτη συχνότητα 10 κωδικολέξεων για την κατηγορία της εκπαίδευσης (Education):

Κωδικολέξη	**Απόλυτη Συχνότητα στην κατηγορία Education**	**Απόλυτη συχνότητα στην αμέσως επόμενη κατηγορία**
school	4250	1819 (Politics)
student	1845	555 (Politics)
teacher	1112	396(Politics)
educ	1015	871(Politics)
univers	801	1826(Science)
children	611	1905(Health)
high	590	1695(Business)
parent	544	754(Health)
counti	528	284(Health)
class	402	172(Politics)

Πίνακας 3: Συχνότητες κωδικολέξεων στις δύο πιο αντιπροσωπευτικές κατηγορίες

Στην δεύτερη στήλη του πίνακα φαίνεται η απόλυτη συχνότητα δέκα τυχαίων κωδικολέξεων για την κατηγορία Εκπαίδευση ενώ στην δεύτερη στήλη του πίνακα βλέπουμε ποια είναι η δεύτερη κατηγορία στην οποία εμφανίζονται με επίσης υψηλή απόλυτη συχνότητα. Παρατηρούμε ότι για τις κωδικολέξεις school, student και teacher ακόμα και αν επιλέξουμε ως τιμή για τον παράγοντα αντιπροσωπευτικότητας την τιμή 2, οι κωδικολέξεις αυτές θα θεωρηθούν από το σύστημα ως αντιπροσωπευτικές της κατηγορίας Εκπαίδευση και αυτό θα οδηγήσει στον εμπλουτισμό της επερώτησης στην οποία τις επέλεξε ο χρήστης. Για τη ρίζα educ θα έπρεπε να μειώσουμε κατά πολύ τον παράγοντα αντιπροσωπευτικότητας (το πολύ στην τιμή 1.15) για να θεωρηθεί αντιπροσωπευτική κωδικολέξη της κατηγορίας. Στην περίπτωση της λέξης children μπορούμε να διαπιστώσουμε ότι ακόμα και με συντελεστή 2.5 θα μπορούσε να θεωρηθεί αντιπροσωπευτική της κατηγορίας Υγεία.

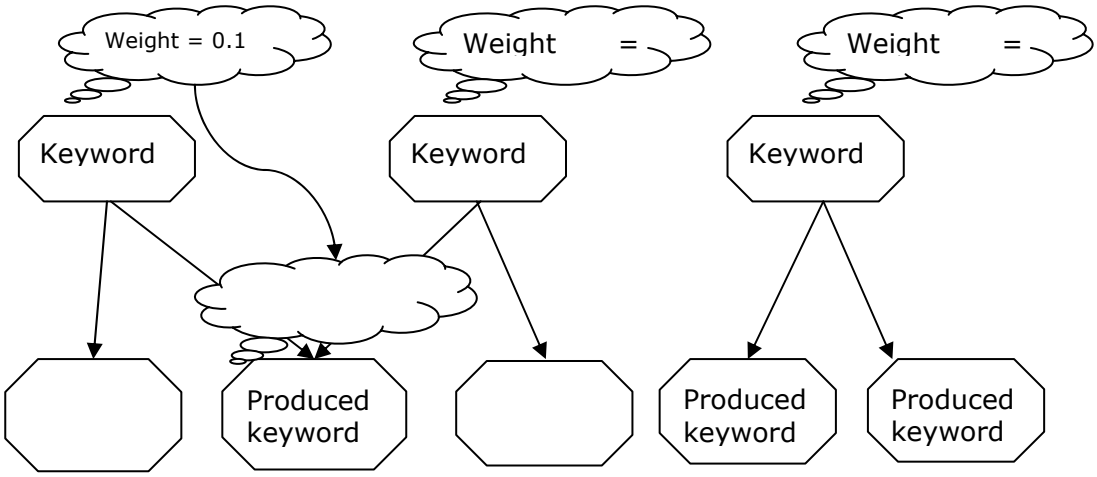
Αξίζει να παρατηρήσουμε ότι για να είναι μια κωδικολέξη αντιπροσωπευτική μιας κατηγορίας δεν απαιτείται (από τον αλγόριθμο) να έχει υπερβολικά μεγάλη απόλυτη συχνότητα σε αυτήν την κατηγορία απλά αρκεί να έχει τη μεγάλυτερη από τις απόλυτες συχνότητες με τις οποίες συναντάται σε άλλες κατηγορίες. Ως παράδειγμα ας φέρουμε τις ρίζες class και educ από τις οποίες η δεύτερη έχει αρκετά μεγάλη απόλυτη συχνότητα στην κατηγορία Εκπαίδευση (1015 έναντι 402 της ρίζας class) ωστόσο μόνο η class θα μπορούσε να θεωρηθεί αντιπροσωπευτική από το σύστημα για την κατηγορία Εκπαίδευση αφού έχει υπερδιπλάσια συχνότητα από την κατηγορία Πολιτική, στην οποία εμφανίζεται με τη δεύτερη μεγαλύτερη συχνότητα (402 έναντι 172).

Όταν τελικά μια κωδικολέξη θεωρηθεί αντιπροσωπευτική μιας θεματικής κατηγορίας, τότε ο αλγόριθμος προβαίνει στον εμπλουτισμό της επερώτησης με άλλες λέξεις από την ίδια κατηγορία. Η επιλογή των λέξεων με τις οποίες θα εμπλουτιστεί η λίστα των όρων αναζήτησης δεν είναι τυχαία. Ανακτώνται οι κωδικολέξεις που έχουν μεγαλύτερη απόλυτη συχνότητα εμφάνισης από την κωδικολέξη που ήδη υπάρχει στη επερώτηση του χρήστη και στην συνέχεια επιλέγονται δύο από αυτές που έχουν την αμέσως μεγαλύτερη απόλυτη συχνότητα στην κατηγορία. Για παράδειγμα αν ο χρήστη επέλεγε κωδικολέξη με ρίζα class

τότε η επερώτηση θα εμπλουτίζοταν με τις ρίζες parent και counti για την επίτευξη πιο εστιασμένου αποτελέσματος στην κατηγορία Εκπαίδευση.

8.3.3.2. Απόδοση Βαρών στις παραγόμενες κωδικολέξεις

Όπως αναφέρθηκε σε προηγούμενη παράγραφο, σε κάθε κωδικολέξη που έδωσε ο χρήστης στη φόρμα αναζήτησης, ο αλγόριθμος αναθέτει ένα συντελεστή βάρους ο οποίος χαρακτηρίζει τη σημαντικότητα της στην διαδικασία αναζήτησης των άρθρων της ΒΔ. Εαν κατά τον έλεγχο των απολύτων συχνοτήτων των κωδικολέξεων προκύψουν επιπλέον λέξεις-κλειδιά (παραγόμενες κωδικολέξεις) που θα εμπλουτίσουν την επερώτηση τότε και σε αυτές θα πρέπει να αποδοθούν συντελεστές βάρους ώστε να υπολογιστεί τελικά η σχετικότητα κάθε άρθρου προς την επερώτηση. Η λογική που ακολουθεί ο αλγόριθμος είναι σχετικά απλή. Σε κάθε παραγόμενη λέξη αποδίδεται ο συντελεστής βάρους της της αρχικής κωδικολέξης από την οποία προήλθε. Αν μια παραγόμενη λέξη προκύπτει από παραπάνω από μια αρχικές τότε σε αυτήν αποδίδεται το μεγαλύτερο βάρος. Η διαδικασία περιγράφεται στο επόμενο σχήμα για τρεις κωδικολέξεις keyword 1, keyword 2 και keyword 3 εκ των οποίων δύο οδηγούν σε εμπλουτισμό με την ίδια κωδικολέξη.



Εικόνα 12. Υπολογισμός βαρών για τις παραγόμενες κωδικολέξεις – εμπλουτισμός της επερώτησης

Τέλος, αν μια από τις παραγόμενες λέξεις υπάρχει ήδη στη λίστα των αρχικών κωδικολέξεων που έδωσε ο χρήστης τότε απλά παραλείπεται από την διαδικασία του εμπλουτισμού και η αρχική λέξη έχει το βάρος που της ανέθεσε ο αλγόριθμος αρχικά.

8.3.4. Προσωποποίηση του αποτελέσματος στο χρήστη

Αφού επιτευχθεί ο εμπλουτισμός της επερώτησης του χρήστη με επιπλέον κωδικολέξεις από τη βάση δεδομένων και ανατεθούν βάρη σε αυτές, όπως περιγράφηκε στην προηγούμενη παράγραφο, γίνεται ο υπολογισμός της σχετικότητας κάθε άρθρου. Για τον υπολογισμό της σχετικότητας του κάθε άρθρου που ανακτάται από την βάση δεδομένων χρησιμοποιείται ο ακόλουθος τύπος.

$$I_{relevance} = \frac{\sum_{i=1}^{N}(x_i y_i)}{\sqrt{\sum_{i=1}^{N}(x_i^2)}\sqrt{\sum_{i=1}^{M}(z_i^2)}}$$

Στον παραπάνω τύπο υπάρχουν οι ακόλουθοι συμβολισμοί: για μια επερώτηση του χρήστη η οποία αποτελείται από N κωδικολέξεις, συμπεριλαμβανομένων και αυτών με τις οποίες εμπλουτίστηκε από το σύστημα (όπως περιγράφηκε σε προηγούμενη παράγραφο), ορίζουμε ως x_i και y_i τα υπολογισμένα βάρη και τις πραγματικές συχνότητες εμφάνισης των κωδικολέξεων στην επερώτηση αντίστοιχα. Από την άλλη μεριά, ο παράγοντας z_i συμβολίζει την πραγματική συχνότητα εμφάνισης κάθε κωδικολέξης που υπάρχει στο άρθρο, για το οποίο υπολογίζεται ο βαθμός σχετικότητας του με την επερώτηση. Ως M συμβολίζεται το συνολικό πλήθος των κωδικολέξεων που υπάρχουν στο άρθρο. Ο παραπάνω τύπος μας δίνει μια κανονικοποιημένη μέτρηση της σχετικότητας κάθε άρθου. Σε περίπτωση που η εμπλουτισμένη επερώτηση του χρήστη δεν περιλαμβάνει καμία από τις κωδικολέξεις που υπάρχουν στο άρθρο τότε ο βαθμός σχετικότητας υπολογίζεται ως μηδέν ενώ στην ακραία περίπτωση που η εμπλουτισμένη επερώτηση περιλαμβάνει όλες τις κωδικολέξεις του άρθρου, ο βαθμός σχετικότητας είναι 1.

Στην περίπτωση της προσωποποιημένης αναζήτησης, η σχετικότητα κάθε άρθρου με την επερώτηση πρέπει να υπολογιστεί λαμβάνοντας υπόψιν το ατομικό προφίλ και τις προτιμήσεις του χρήστη που υπέβαλλε την επερώτηση. Για το προφίλ κάθε χρήστη, κατά την διαδικασία εγγραφής του στο σύστημα, αναθέτουμε διαφορέτικά βάρη σε διαφορετικές ομάδες κωδικολέξεων έτσι ώστε να περιγράψουμε με τον καλύτερο τρόπο τις διαφορετικές του προτιμήσεις. Η λογική που υπάρχει πίσω από την ανάθεση διαφορετικών βαρών σε διαφορετικές ομάδες κωδικολέξεων του συστήματος είναι η εξής: κωδικολέξεις με μεγαλύτερη σχετικότητα ως προς την θεματική κατηγορία που επέλεξε ο χρήστης παίρνουν θετικά βάρη ενώ κωδικολέξεις που ανήκουν σε κατηγορίες για τις οποίες ο χρήστης έχει δηλώσει μικρότερο ενδιαφέρον ή αδιαφορεί – έχει δώσει αρνητικό ενδιαφέρον κατά την εγγραφή – λαμβάνουν μικρότερα η αρνητικά βάρη αντιστοίχως.

Όπως αναφέραμε, το προφίλ του χρήστη αρχικοποιείται κατά τη διαδικασία εγγραφής του στο σύστημα, όπου παρέχει με ένα δείκτη (από -5 έως και 5) το βαθμολογημένο ενδιαφέρον του προς τις θεματικές κατηγορίες του συστήματος. Αυτό το προφίλ μπορεί να αλλάξει δυναμικά κατά την χρήση του συστήματος από το χρήστη σύμφωνα με τη συνολική του συμπεριφορά (άρθρα που επισκέπτεται, χρόνος παραμονής σε κάθε άρθρο, κλπ). Αυτό θα παρουσιαστεί πιο αναλυτικά στην επόμενη παράγραφο. Με βάση αυτά τα συνεχώς εξελισσόμενα προφίλ των χρηστών, μπορούμε να υπολογίσουμε ένα δείκτη σχετικότητας κάθε άρθρου του συστήματος για κάθε χρήστη. Ο δείκτης αυτός αποτελεί μέτρο του κατά πόσο ένας χρήστης θα ενδιαφερόταν να επισκεφτεί ένα άρθρο το οποίο ωστόσο μπορεί ποτέ να μην επισκεφτεί. Ο τύπος που χρησιμοποιούμε για τον υπολογισμό αυτής της σχετικότητας είναι ο ακόλουθος:

$$I_{personalized} = \frac{\sum_{i=1}^{N}(p_i w_i)}{\sqrt{\sum_{i=1}^{N}(p_i^2)}\sqrt{\sum_{i=1}^{M}(z_i^2)}}$$

Οι συμβολισμοί που υπάρχουν σε αυτόν το τύπο είναι οι ακόλουθοι: N είναι ο αριθμός των κωδικολέξεων για τις οποίες έχουν ανατεθεί βάρη (θετικά ή αρνητικά) στο προφίλ του χρήστη, P_i είναι το βάρος κάθε κωδικολέξης στο προφίλ του χρήστη και w_i είναι η πραγματική συχνότητας εμφάνισης κάθε μιας από αυτές τις κωδικολέξεις στο εν λόγω άρθρο. Με M συμβολίζουμε και πάλι τον συνολικό αριθμό κωδικολέξεων που υπάρχουν στο άρθρο και με Z_i συμβολίζεται η πραγματική συχνότητα εμφάνισης της κωδικολέξης στο άρθρο.

Από τον τύπο αυτό, μπορούμε να καταλάβουμε τον τρόπο με τον οποίο κωδικολέξεις με υψηλή βαρύτητα στο προφίλ ενός χρήστη μπορούν να αυξήσουν την σχετικότητα ενός άρθρου και συνεπώς τη τελική θέση με την οποία αυτό το άρθρο θα εμφανιστή στο τελικό αποτέλεσμα που θα παρουσιαστεί στο χρήστη αφού ολοκληρωθεί η αναζήτηση. Προφανώς επίσης, για διαφορετικούς χρήστες θα υπολογιστούν διαφορετικοί συντελεστές σχετικότητας των άρθρων ως προς τα προφίλ τους για τις ίδιες επερωτήσεις γεγονός που αναδεικνύει την ικανότητα του συστήματος να δώσει προσωποποιημένο αποτέλεσμα.

8.3.5. Διαχείρηση του Προφίλ Χρήστη

8.3.5.1. Διαμόρφωση και Εξέλιξη του προφίλ του Χρήστη

Η προσωποποίηση στο χρήστη είναι ένα από τα πιο σημαντικά κομμάτια του συστήματος καθώς σε αυτό το στάδιο διαμορφώνεται το δυναμικό προφίλ και προβάλλονται πίσω στο χρήστη όλα τα αποτελέσματα των προηγούμενων μηχανισμών.

Η προσωποποίηση στο χρήστη γίνεται σε επίπεδο διαδικτύου με τη συνεργασία PHP, AJAX και βάσης δεδομένων. Η προσωποποίηση βασίζεται σε συγκεκριμένες παραμέτρους προκειμένου να είναι πληρέστερη και να είναι εφικτή η καλύτερη δημιουργία προφίλ χρήστη. Οι παράμετροι που θέσαμε στο σύστημα για την προσωποποίηση είναι:

- Οι επιλογές του χρήστη που αφορούν τις κατηγορίες που έχει το σύστημα (μόλις κάνει εγγραφή)
  - Βαθμολόγηση των κατηγοριών ανάλογα με το πόσο ενδιαφέρουν τον χρήση
- Οι επιλογές του χρήστη μόλις του εμφανίζονται τα άρθρα του τελικού αποτελέσματος.
  - Επιλογή του χρήστη να διαβάσει ένα άρθρο.
  - Επιλογή του χρήστη να μη διαβάσει ένα άρθρο.
- Οι επιλογές του χρήστη κατά τη διάρκεια της ανάγνωσης ενός άρθρου.
  - Πόσο χρόνο καταναλώνει σε ένα άρθρο.
  - Πόσα άρθρα του ίδιου θέματος διαβάζει κατά τη διάρκεια μίας συνεδρίας
  - Η επιλογή του να στείλει ένα άρθρο σε ένα φίλο
  - Η επιλογή να τυπώσει ένα άρθρο
  - Η επιλογή να αναζητήσει παραπλήσια άρθρα σε όλη τη συλλογή.

Όλα τα παραπάνω αποτελούν παραμέτρους που διαμορφώνουν το προφίλ ενός χρήστη. Όμως ας δούμε τι εννοούμε όταν αναφερόμαστε στο προφίλ ενός χρήστη. *εδομένων των διαδικασιών με τις οποίες εξάγονται τα αποτελέσματα τόσο για την κατηγοριοποίηση (επιλογή σε ποια κατηγορία ανήκει ένα άρθρο που μόλις μπήκε στο σύστημα) όσο και για τις περιλήψεις έχουμε δει πως αυτό που έχει τη μεγαλύτερη σημασία είναι να εντοπίσουμε τις λέξεις κλειδιά. Έτσι, λοιπόν, και για το προφίλ του χρήστη αυτό που πραγματοποιούμε είναι να δημιουργήσουμε λίστες με λέξεις κλειδιά που έχουν κάποια βάρη. Σε αυτή την περίπτωση τα βάρη είναι θετικά και αρνητικά και προδίδουν το κατά πόσο ο χρήστης ενδιαφέρεται για κάποια λέξη κλειδί ή όχι καθώς και το μέγεθος ενδιαφέροντος.

8.3.5.2. Αλγόριθμος Διαμόρφωσης Αρχικού Προφίλ

Ως δεδομένα έχουμε στο σύστημά μας έχουμε 7 κατηγορίες τις οποίες τις χαρακτηρίζουν λέξεις κλειδιά με συγκεκριμένα βάρη. Ο παρακάτω πίνακας δείχνει ένα τέτοιο παράδειγμα για μία από τις κατηγορίες του συστήματός μας.

Θεματική Κατηγορία	Αναγνωριστικό Κωδικολέξης	Σχετική Συχνότητα	Απόλυτη Συχνότητα
1	42	0.00105974	298
1	43	0.000927275	201
1	44	0.00172208	201
1	41	0.0103325	188
1	37	0.00516625	150
1	228	0.0149689	148
1	45	0.00251689	141

Πίνακας 4: Κωδικολέξεις με τις μεγαλύτερες συχνότητες σε μια θεματική κατηγορία

Το σκεπτικό είναι πως κάθε χρήστης αντιπροσωπεύει μία υποκατηγορία ή πιο σωστά, μία σειρά από υποκατηγορίες. Αυτό σημαίνει πως εφόσον οι λίστες με τις λέξεις κλειδιά δύνανται να χαρακτηρίσουν μία κατηγορία αυτό συνεπάγεται και πως λίστες με λέξεις κλειδιά δύνανται να χαρακτηρίσουν τις επιλογές και τις προτιμήσεις ενός χρήστης. Αυτό που μας ενδιαφέρει συνεπώς είναι να μπορέσουμε από τις διαδικασίες που περιγράψαμε παραπάνω να καταλήγουμε σε λέξεις κλειδιά και συγκεκριμένα βάρη σε κάθε μία προκειμένου να χαρακτηρίσουμε το χρήστη. Σε πρώτη φάση αυτό που κάνουμε είναι να διαμορφώσουμε κάποιο αρχικό προφίλ για το χρήστη κατά τη διάρκεια που πραγματοποιεί εγγραφή στο σύστημα. *εδομένου ότι θέλουμε να κρατήσουμε τις διαδικασίες όσο το δυνατόν πιο διαφανείς προς τους χρήστες είναι ίσως το μόνο σημείο που μπορούμε ανώδυνα να βάλουμε το χρήστη στη διαδικασία του να συμπληρώσει κάποια στοιχεία για το προφίλ του.

Η διαδικασία εγγραφής και γενικά το περιβάλλον διεπαφής αποτελούν την βασική μονάδα επικοινωνίας του χρήστη με το σύστημα. Ένας χρήστης εγγράφεται στο σύστημα δίνοντας πληροφορίες για το μέγεθος της συσκευής που χρησιμοποιεί και δίνοντας πληροφορίες για τις κατηγορίες που θέλει να παρακολουθεί. Ένας χρήστης είναι δυνατόν να αλλάξει τα στοιχεία του μελλοντικά, κάτι που βέβαια δεν επηρεάζει άμεσα τα στοιχεία που έχουν ήδη συλλεγεί για το προφίλ του, εκτός κι αν ο ίδιος επιθυμεί δημιουργία από την αρχή του προφίλ που ήδη έχει. Οι πληροφορίες αποθηκεύονται στην κεντρικοποιημένη βάση δεδομένων και ανανεώνονται συνεχώς με το δυναμικό προφίλ του όπως θα δούμε στην επόμενη ενότητα. Όταν ο χρήστης βρίσκεται στη διαδικασία εγγραφής στο σύστημα του παρουσιάζονται όλες οι κατηγορίες του συστήματος και του ζητήται να δηλώσει την προτίμησή του για κάθε κατηγορία. Ο χρήστης καλείται να επιλέξει μία βαθμολογία για κάθε κατηγορία από -5 έως 5. Το -5 μεταφράζεται σαν η κατηγορία δε με αντιπροσωπεύει καθόλου ενώ το +5 σημαίνει πως η κατηγορία αντιπροσωπεύει απόλυτα το χρήστη. Η επιλογή του 0 σαν προτίμηση κατηγορίας μεταφράζεται σαν ουδέτερη στάση απέναντι στην κατηγορία. Εκμεταλλευόμενοι τις απαντήσεις των χρηστών μπορούμε να διαμορφώσουμε ένα αρχικό προφίλ για το χρήστη. Αυτό γίνεται ως εξής. Αρχικά δημιουργούμε εγγραφές για τις κατηγορίες που αρέσουν στο χρήστη και γι αυτές που ο χρήστης δεν προτιμά. Αυτό θα μας βοηθήσει να κάνουμε ένα πρώτο ξεκαθάρισμα των άρθρων ανάμεσα σε αυτά που ο χρήστης θέλει να δει και σε αυτά που δεν τον ενδιαφέρουν, ανάλογα με τις γενικές κατηγορίες που έχει επιλέξει. Ο χρήστης όμως δεν επιλέγει απλώς τι θέλει να βλέπει και τι δε θέλει. Έχει δώσει και κάποια βαθμολογία για κάθε κατηγορία. Χρησιμοποιώντας αυτά τα δεδομένα μπορούμε να δημιουργήσουμε μία πιο αναλυτική περιγραφή του προφίλ. Το αναλυτικό προφίλ όπως έχει ήδη αναφερθεί περιλαμβάνει λίστες με λέξεις κλειδιά όπως αυτές που υπάρχουν για τις κατηγορίες που δείχνουν ποιες λέξεις κλειδιά ενδιαφέρουν το χρήστη και ποιες δεν τον αφορούν. Σε αυτή την περίπτωση επιτρέπονται τόσο θετικά βάρη όσο και

αρνητικά. Ο υπολογισμός των βαρών για τις λέξεις κλειδιά του χρήστη υπολογίζονται από τον παρακάτω αλγόριθμο.

```
For each (selection s) {
If (s!=0) {
Keyword_name_usr = select 20*s keywords from category keywords
// the keywords used for categorization, summarization etc
Keyword_weight_usr = select (2*s*relative frequency) from category keywords
// the same list as above
}
else {
Keyword_name_usr = select 10 keywords from category keywords
Keyword_weight_usr = select relative_frequency from category. keywords
}
Insert into user profile keyword_name_usr, keyword_weight_usr
If exists
Update user profile set keyword_weight += keyword_weight_usr where
keyword_name = keyword_name_usr
}
```

Αλγόριθμος 4: Υπολογισμός βαρών για τις κωδικολέξεις

Υποθέτουμε ότι ο χρήστης κάνει κάποιες επιλογές για τις κατηγορίες και επιλέγει από -5 έως 5. Από αυτές τις επιλογές επιλέγουμε 20s λέξεις κλειδιά, όπου s είναι η επιλογή του χρήστη (s Є [-5..5]) από τη λίστα με τις λέξεις κλειδιά που αφορούν την κατηγορία, όπως ο πίνακας που είδαμε παραπάνω. Εν συνεχεία, επιλέγουμε τη σχετική συχνότητα κάθε λέξης και την πολλαπλασιάζουμε με 2s. Αν για παράδειγμα ο χρήστης έχει επιλέξει για μία κατηγορία την επιλογή -3 και μία συγκεκριμένη λέξη κλειδί για την κατηγορία έχει σχετική συχνότητα 0,12 τότε στον πίνακα του χρήστη η συγκεκριμένη λέξη θα πάρει σχετική συχνότητα -0,12. αυτός ο αριθμός μας δείχνει και το πόσο ο χρήστης ενδιαφέρεται για τη συγκεκριμένη λέξη κλειδί. Στο παράδειγμα που δείξαμε ο χρήστης δεν ενδιαφέρεται για τη συγκεκριμένη λέξη. Πραγματοποιώντας αυτή τη διαδικασία καταλήγουμε σε μία αρχική λίστα με λέξεις κλειδιά και σχετικές συχνότητες για το χρήστη οι οποίες μας δίνουν τα παρακάτω στοιχεία:

- Πολλές λέξεις κλειδιά από τις κατηγορίες που έχει επιλέξει ο χρήστης με μεγάλο σκόρ, είτε θετικό είτε αρνητικό και παράλληλα πολύ λίγες λέξεις από τις κατηγορίες που έχει δηλώσει ο χρήστης με χαμηλό σκορ. Πρόκειται για κατηγορίες που είναι αδιάφορες στο χρήστη και άρα, λέξεις κλειδιά από αυτές τις κατηγορίε δεν είναι απαραίτητες για το προφιλ του χρήστη.
- Μεγάλη θετική τιμή για τις σχετικές συχνότητες των λέξεων κλειδιών που ανήκουν στις κατηγορίες που έχει επιλέξει ο χρήστης με μεγάλο σκορ και μεγάλη απόλυτα αρνητική τιμή για τις σχετικές συχνότητες των λέξεων κλειδιών που ανήκουν σε κατηγορίες που έχει επιλέξει ο χρήστης με πολύ μικρό σκορ.

Αυτά τα στοιχεία μπορούν να μας δώσουν πληροφορίες για να εξάγουμε τις παρακάτω ενέργειες:

- Επιλογή κειμένων από τις κατηγορίες που ενδιαφέρουν το χρήστη
- Αποφυγή επιλογής κειμένων από τις κατηγορίες που δεν ενδιαφέρουν το χρήστη
- Επιλογή κειμένων από κατηγορίες που ενδιαφέρουν το χρήστη ενώ παράλληλα δεν ανήκουν σε κατηγορίες που δεν ενδιαφέρουν το χρήστη (ένα κείμενο μπορεί να ανήκει σε πολλές κατηγορίες)
- Ξεκαθάρισμα των αποτελεσμάτων του μηχανισμού αυτόματης εξαγωγής περίληψης προσθέτοντας τον παράγοντα της προσωποποίησης.

Η διαδικασία που περιγράφηκε, συμπεριλαμβανομένης και της κατασκευής της λίστας με τις λέξεις κλειδιά πραγματοποιήθηκε προκειμένου να έχουμε κάποια πρώτα στοιχεία για το αρχικό προφίλ του χρήστη. Στη συνέχεια θα περάσουμε στην κατασκευή του δυναμικού προφίλ χρήστη, το οποίο μεταβάλλεται με τη χρήση του δικτυακού τόπου. Είναι σημαντικό το γεγονός πως όσο περισσότερο χρησιμοποιεί ο χρήστης το δικτυακό τόπο, τόσο καλύτερα διαμορφώνεται το προφίλ του.

8.3.5.3. Δυναμική Διαμόρφωση Προφίλ Χρήστη

Όσο ο χρήστης χρησιμοποιεί το δικτυακό τόπο, τόσο καλύτερα διαμορφώνεται το προφίλ του από τα στοιχεία που συλλέγονται από τις επιλογές του. Όπως έχουμε ήδη αναφέρει τα στοιχεία που ελέγχονται για τη δυναμική διαμόρφωση του προφίλ του χρήστη είναι:

- Οι επιλογές του χρήστη μόλις του εμφανίζονται άρθρα είναι:
  - Επιλογή του χρήστη να διαβάσει ένα άρθρο
  - Επιλογή του χρήστη να μη διαβάσει ένα άρθρο
- Οι μετρήσεις/παρατηρήσεις που γίνονται τη στιγμή που ο χρήστης διαβάζει ένα άρθρο είναι:
  - Πόσο χρόνο καταναλώνει στην ανάγνωση του άρθρου.
  - Πόσα άρθρα της ίδιας θεματικής κατηγορίας διαβάζει κατά τη διάρκεια μιας συνεδρίας.
  - Η επιλογή αποστολής του άρθρου σε ένα φίλο
  - Η επιλογή της εκτύπωσης του άρθρου
  - Η επιλογής της αναζήτησης παραπλήσιων άρθρων σε όλη τη συλλογή.

Στη συνέχεια ας παρακολουθήσουμε πως συμπεριφέρεται ο μηχανισμός ανάλογα με τις επιλογές που κάνει ο χρήστης.

8.3.5.4. Επιλογές του χρήστη μόλις εμφανίζονται τα άρθρα

Από το χρήστη του συστήματός μας περιμένουμε όταν του εμφανιστούν τα τελευταία 20 άρθρα, κάποια από αυτά να τα διαβάσει και άλλα να μην τα δει καθόλου. Και οι δύο αυτές αντιδράσεις κάτι μπορεί να σημαίνουν όμως και γι αυτό κάθε τέτοιο στοιχεία είναι αντικείμενο μελέτης για το μηχανισμό μας. Αυτό που μπορούμε να καταλάβουν δημιουργώντας εικονικά προφίλ στο μηχανισμό μας είναι πως ο χρήστης θα επιλέξει να διαβάσει τα άρθρα που τον ενδιαφέρουν ενώ στα υπόλοιπα δε θα δώσει σημασία. Αυτή τη συμπεριφορά χρήστη την καταγράφουμε και την εκμεταλλευόμαστε προκειμένου να διαμορφώσουμε το προφίλ του. Από τα άρθρα που παρουσιάζουμε στο χρήστη επιλέγουμε τις λέξεις κλειδιά. Για κάθε άρθρο που επιλέγει ο χρήστης να διαβάσει προσθέτουμε τις συγκεκριμένες λέξεις κλειδιά στο προφίλ του βάση της σχετικής συχνότητας που παρουσιάζουν στο συγκεκριμένο άρθρο. Πρόκειται για μία πολύ μεγάλη σχετική συχνότητα κάτι που είναι επιθυμητό καθότι πρόκειται για λέξεις κλειδιά σε ένα αρθρο που ενδιαφέρει το χρήστη. Όσον αφορά τα άρθρα που δεν επέλεξε ο χρήστης. Σε αυτή την περίπτωση συγκεντρώνουμε όλες τις λέξεις κλειδιά από αυτά τα άρθρα και ανανεώνουμε τις λέξεις κλειδιά του προφίλ χρήστη με αρνητική σχετική συχνότητα. Σε αυτή την περίπτωση και προκειμένου να διατηρηθεί η ακεραιότητα του μηχανισμού δεν αφαιρούμε με την πολύ μεγάλη σχετική συχνότητα που έχουν οι λέξεις κλειδιά αλλά με το . αυτής. Έτσι σε περίπτωση που ένας χρήστης δεν ανάγνωσε ένα άρθρο που τον ενδιέφερε επειδή του διέφυγε δεν υπάρχει μεγάλη διαφορά στο προφίλ του. Αντίθετα, για τα άρθρα που επιλέγει ο χρήστης παρατηρείται μεγάλη αλλαγή στο προφίλ του.

8.3.5.5. Επιλογές του Χρήστη κατά την ανάγνωση ενός άρθρου

Την ώρα που ο χρήστης επιλέγει να διαβάσει ένα άρθρο, όπως ήδη είπαμε οι λέξεις κλειδιά αυτού του άρθρου προστίθενται στο προφίλ του. Αυτό που δεν είπαμε είναι πως υπάρχει μία δικλείδα ασφαλείας για την περίπτωση που ο χρήστης κάνει εσφαλμένη επιλογή. Έτσι, αν ο χρήστης ανοίξει ένα άρθρο για να το διαβάσει και το κλείσει μέσα σε 7 δευτερόλεπτα τότε αυτό δεν προσμετράται σε αυτά που έχει διαβάσει. Αντίθετα θεωρείται σε αυτά που δεν έχει διαβάσει. Για τον υπολογισμό του χρόνου αυτού, χρησιμοποιείται τεχνολογία AJAX. Επιπλέον υπάρχει μία δικλείδα ασφαλείας για την περίπτωση που ο χρήστης «ξεχάσει» για οποιονδήποτε λόγο ανοιχτό το παράθυρο που περιέχει το σώμα του άρθρου. Ο χρόνος αυτός έχει οριστεί στα 2 λεπτά, κάτι το οποίο σημαίνει πως μετά την πάροδο δύο λεπτών που ο χρήστης έχει ανοιχτό ένα άρθρο, θεωρείται πως το έχει ξεχάσει ανοιχτό και έτσι αυτός ο χρόνος δεν είναι αντιπροσωπευτικός για το χρόνο που δαπάνησε ο χρήστης στο συγκεκριμένο άρθρο.

Ο χρόνος που καταναλώνει ο χρήστης σε ένα άρθρο είναι φυσικά ευθέως ανάλογος με το μέγεθος του άρθρου. Ας υπενθυμίσουμε σε αυτό το σημείο πως στο χρήστη προβάλλεται η εξής πληροφορία μόλις διαβάζει ένα άρθρο:

- Ο τίτλος του άρθρου
- Η ημερομηνία που το άρθρο καταγράφηκε στο σύστημα
- Οι πιθανές κατηγορίες στις οποίες ανήκει
- Η περίληψη του άρθρου
- Το σώμα του άρθρου, όπως αυτό έχει εξαχθεί από το μηχανισμό εξαγωγής χρήσιμου κειμένου

Ο χρήστης βλέπει τα 4 πρώτα στοιχεία άμεσα ενώ το σώμα του άρθρου μπορεί να το δει επιλέγοντας ένα σύνδεσμο και αποτελεί και αυτό στοιχείο που καταγράφεται για την προσωποποίηση. Τα στοιχεία που μπορεί να επιλέξει ο χρήστης μόλις βλέπει ένα άρθρο και μπορεί να αναγνωρίσει ο μηχανισμός είναι:

- Να διαβάσει το κύριο σώμα του άρθρου
- Να τυπώσει το άρθρο (περίληψη ή κύριο σώμα)
- Να ακολουθήσει το σύνδεσμο προς τη δικτυακή τοποθεσία στην οποία φιλοξενείται το άρθρο.
- Να στείλει το άρθρο σε ένα φίλο
- Να διαβάσει ταυτόσημα άρθρα που υπάρχουν σε άλλους δικτυακού τόπους.
- Να διαβάσει παρόμοια άρθρα του τελευταίου 24ώρου.
- Να διαβάσει παρόμοια άρθρα των τελευταίων 3 ημερών

Κάθε μία από αυτές τις ενέργειες έχει άμεση επίδραση στον τρόπο με τον οποίο ανανεώνονται οι λέξεις κλειδιά στο προφίλ του χρήστη. Κάποιες ενέργειες θεωρούνται πιο σημαντικές από άλλες και έτσι δεν αυξάνονται ομοιόμορφα οι συχνότητες των λέξεων κλειδιών του προφίλ του χρήστη. Όπως έχουμε ήδη δει η διαμόρφωση του προφίλ συνίσταται στην καταγραφή λέξεων κλειδιών με κάποιο βάρος. Η αρχική τιμή αυτού του βάρους συλλέγεται από τις λέξεις κλειδιών των κατηγοριών ενώ στην πορεία από τον πίνακα που περιέχει τις λέξεις κλειδιά του συγκεκριμένου άρθρου μαζί με τα βάρη τους.

Σύμφωνα με τα παραπάνω και βάση των ενεργειών του χρήστη, οι λέξεις κλειδιά στο προφίλ του χρήστη διαμορφώνονται βάσει του παρακάτω πίνακα.

Ενέργεια του χρήστη	**Επίδραση**	**Πολλαπλασιαστής επί της σχετικής συχνότητας**
Μη επιλογή του άρθρου	Αρνητική	-0.25
Επιλογή του άρθρου	Θετική	+1

Ενέργεια του χρήστη	Επίδραση	Πολλαπλασιαστής επί της σχετικής συχνότητας
Ανάγνωση του κυρίου σώματος	Θετική	+0.25
Σύνδεσμος στο δικτυακό τόπο που φιλοξενεί το άρθρο	Θετική	+0.25
Εκτύπωση του άρθρου	Θετική	+0.15
Αποστολή σε φίλο	Θετική	+0.15
Ταυτόσημα άρθρα	Θετική	+0.20
Παρόμοια (παραπλήσιουπεριεχομένου) άρθρα	Θετική	+0.17
Σχετικά άρθρα	Θετική	+0.15

Πίνακας 5: Ενέργειες του χρήστη και επιδράσεις στο προφίλ του

Σύμφωνα με τον παραπάνω πίνακα, αν ένας χρήστης επιλέξει ένα άρθρα, διαβάσει το κύριο σώμα του (εκτός από την περίληψη), το τυπώσει, το στείλει σε ένα φίλο του, μεταβεί στη σελίδα με τα ταυτόσημα άρθρα και ακολουθήσει το σύνδεσμο προς το δικτυακό τόπο που φιλοξενεί το άρθρο τότε συνολικά για κάθε λέξη κλειδί που είναι στο άρθρο θα έχουμε μια ανανέωση/προσθήκη στο προφίλ του χρήστη της τάξης του 2x (relative frequency).

8.4. Βελτίωση της αναζήτησης με χρήση τεχνικής Caching

Πριν την εκκίνηση της διαδικασίας αναζήτησης των άρθρων που ικανοποιούν την επερώτηση του χρήστη, το σύστημα αναζητά cached δεδομένα από προηγούμενες συνεδρίες των χρηστών του συστήματος. Όλα τα cached δεδομένα αποθηκεύονται στον αποθηκευτικό χώρο του server που φιλοξενεί το σύστημα και επίσης ο αλγόριθμος λειτουργεί στη μνήμη του εξυπηρετητή. Η διαδικασία που θα περιγραφεί στη συνέχεια ωφελεί τόσο του εγγεγραμμένους χρήστες του συστήματος (χρήστες-μέλη) όσο και του μη εγγεγραμένους χρήστες (φιλοξενούμενοι χρήστες – guests) χωρίς να δημιουργείται καθόλου επιπλέον υπολογιστικό φορτίο για αυτούς (server-side caching).

Για κάθε επερώτηση που υποβάλλεται στο σύστημα, σώζουμε σε ένα ξεχωριστό πίνακα της βάσης δεδομένων πληροφορίες για την διαμόρφωση της επερώτησης. Οι πληροφορίες αυτές περιλαμβάνουν το αναγνωριστικό του χρήστη που υπέβαλλε την επερώτηση, την ακριβή χρονική στιγμή που υποβλήθηκε (αποθηκεύεται ως timestamp), τις κωδικολέξεις που χρησιμοποιήθηκαν (αποθηκεύονται ως λίστα διαχωρισμένη με κόμματα) καθώς επίσης και ένα αλφαριθμητικό (string) που περιέχει πληροφορίες για άλλες παραμέτρους της αναζήτησης όπως τη θεματική κατηγορία εφόσον αυτή προσδιορίστηκε κατά τον καθορισμό της επερώτησης και τον λογικό τελεστή που χρησιμοποιήθηκε για το σχηματισμό της επερώτησης από τις κωδικολέξεις της. Για τα παραπάνω δεδομένα, ο αλγόριθμος που χρησιμοποιήσαμε για την εύρεση cached αποτελεσμάτων λειτουργεί «στατικά» ως προς το ταίριασμα των κωδικολέξεων της τρέχουσας επερώτησης με προγενέστερες. Ας το εξηγήσουμε κάπως καλύτερα αυτό. Αν ένας χρήστης υποβάλλει μια επερώτηση που περιλαμβάνει τις λέξεις «nuclear technology», επιλέγοντας ως επιθυμητή θεματική ενότητα του αποτελέσματος την κατηγορία «science», η επερώτηση του δεν θα βρεί αντιστοίχιση σε μια προγενέστερη cached επερώτηση που περιλαμβάνει ακριβώς τις ίδιες κωδικολέξεις αλλά αναφέρεται σε άλλη θεματική ενότητα. Επιπρόσθετα, αν υποβληθεί μια επερώτηση που περιλαμβάνει παραπάνω από μια κωδικολέξεις δεν θα βρεί αντιστοίχιση μέσω του αλγόριθμου σε cached επερωτήσεις που να περιέχουν

υπερσύνολο ή υποσύνολο των κωδικολέξεων της επερώτησης. Για παράδειγμα, αν η επερώτηση περιέχει τις λέξεις «Monaco Circuit Formula», πιθανώς αναφερόμενη στο διάσημο αγώνα ταχύτητας του Μονακό, δε θα θεωρηθεί αντίστοιχη με μια cached επερώτηση που περιέχει τις κωδικολέξεις «circuit formula», που πιθανότατα αναφέρεται σε ένα τύπο φυσικής για ηλεκτρονικά κυκλώματα. Η απόφαση για αυτό το χαρακτηριστικό της υλοποίησης έγινε ούτως ώστε να αποφευχθούν σημασιολογικές αμφισημίες στη διαδικασία ταιριάσματος των κωδικολέξεων με άρθρα του συστήματος αφού μια λέξη μπορεί να έχει αρκετά διαφορετικά νοήματα ανάλογα με τα συμφραζόμενα της πρότασης.

Η δυναμική λογική του caching αλγορίθμου φαίνεται στον τρόπο με τον οποίο χρησιμοποιούνται τα διαστήματα DATE FROM και DATE TO τα οποία χρησιμοποιεί και επιλέγει ο χρήστης κατά την αρχική διαμόρφωση της επερώτησης στη φόρμα αναζήτησης του συστήματος. Αυτό η δυναμικότητα του αλγόριθμου επιλέχτηκε λαμβάνοντας υπόψιν το γεγονός ότι είναι πολύ πιθανόν για πολλούς διαδικτυακούς χρήστες να υποβάλλουν πανομοιότυπες επερωτήσεις επαναλαμβανόμενα μέσα στην ίδια συνεδρία, την ίδια μέρα ή γενικά σε μικρές χρονικές περιόδους (για παράδειγμα, επερωτήσεις που αφορούν θέματα επικαιρότητας). Για το λόγο αυτό, ο αλγόριθμος που σχεδιάστηκε λαμβάνει υπόψιν τις ακόλουθες 4 περιπτώσεις για τα cached πεδία DATE FROM και DATE TO καθώς και για αντίστοιχα πεδία της υποβαλλόμενης επερώτησης.

- **Περίπτωση 1:** οι τιμές των DATE FROM – DATE TO της υποβαλλόμενης επερώτησης ορίζουν μια χρονική περίοδο που είναι υποσύνολο της περιόδου που ορίζεται από τα αντίστοιχα πεδία στην cached επερώτηση. Σε αυτήν την περίπτωση, το σύνολο των άρθρων που είναι το επιθυμητό αποτέλεσμα της επερώτησης που υπέβαλλε ο χρήστης υπάρχει αποθηκευμένο στον πίνακα με τα cached αποτελέσματα. Φυσικά, είναι πιθανόν τα cached αποτελέσματα να περιέχουν και ορισμένα παραπάνω άρθρα που δεν ανήκουν στο χρονικο διάστημα που ζήτησε ο χρήστης στην αρχική επερώτηση. Στην περίπτωση αυτή, ο αλγόριθμος φέρνει σε μικρό χρονικό διάστημα, χωρίς να εκτελεστεί αναζήτηση, όλα τα cached αποτελέσματα και στη συνέχεια φιλτράρει εκείνα τα οποία είναι εντός του ζητούμενου χρονικού διαστήματος. Κάτι τέτοιο είναι εφικτό, καθώς για κάθε αποθηκευμένη cached επερώτηση, τα άρθρα του αποτελέσματος υπάρχουν μαζί με μεταδεδομένα σε μορφή XML και αρκεί ένα απλό parsing του XML κειμένου για την εξαγωγή εκείνων των άρθρων που χρειάζονται. Για αυτή την περίπτωση αντιστοίχισης της υποβαλλόμενης επερώτησης με τα cached δεδομένα της βάσης δεδομένων του server δεν χρειάζεται καμία ενημέρωση καθώς δεν εκτελείται καθόλου αναζήτηση για την εύρεση νέων άρθρων.
- **Περίπτωση 2:** το πεδίο με την ημερομηνία DATE FROM της υποβαλλόμενης επερώτησης προηγείται χρονικά του πεδίου DATE FROM της cached επερώτησης ενώ το πεδίο με την ημερομηνία DATE TO της υποβαλλόμενης επερώτησης έπεται της ημερομηνίας DATE TO της cached επερώτησης. Στην περίπτωση αυτή, που είναι και η χειρότερη που μπορεί να αντιμετωπίσει ο cached αλγόριθμος, τα επιθυμητά άρθρα του αποτελέσματος είναι υπερσύνολο των άρθρων που υπάρχουν αποθηκευμένα στον cache πίνακα της βάσης δεδομένων. Ως συνέπεια αυτού, ο αλγόριθμος που υλοποιήθηκε ανακτά από την cache γρήγορα όλα τα άρθρα της cache τα οποία θα υπάρχουν στο τελικό αποτέλεσμα που ζήτησε ο χρήστης. Ωστόσο, θα διεξαχθούν και δύο επιπλέον διαδικασίες αναζήτησης για άρθρα που υπάρχουν στα διαστήματα πρίν και μετά τη χρονική περίοδο στην οποία αναφέρονται τα cached δεδομένα. Όταν η διαδικασία αναζήτησης ολοκληρωθεί και τότε το τελικό αποτέλεσμα παρουσιάζεται στο χρήστη ενώ θα πρέπει παράλληλα να ενημερωθεί και η cache της βάσης δεδομένων με τα καινούργια άρθρα που προέκυψαν από τις δύο εκτελέσεις της αναζήτησης. Προφανών και οι τιμές των πεδίων DATE FROM και DATE TO της cache θα χρειαστεί να τροποποιηθούν ώστε να ανταποκρίνονται στην καινούργια χρονική περίοδο που προέκυψε και να

χρησιμοποιηθούν σε μελλοντικές αντιστοιχίσεις υποβαλλόμενων επερωτήσεων με cached επερωτήσεις.

- **Περίπτωση 3:** η ημερομηνία DATE FROM της υποβαλλόμενης επερώτησης προηγείται χρονικά της ημερομηνίας DATE FROM της cached επερώτησης ενώ η ημερομηνία DATE TO της υποβαλλόμενης επερώτησης είναι χρονικά ανάμεσα στην DATE FROM και DATE TO της cached επερώτησης. Στην περίπτωση αυτή, μέρος των επιθυμητών άρθρων του αποτελέσματος υπάρχει ήδη έτοιμο στην cache οπότε για αυτό το κομμάτι δεν χρειάζεται να εκτελεστεί εκ νέου αναζήτηση. Συνεπώς, ο αλγόριθμος ανακτά από τον cache πίνακα όλα τα αποτελέσματα και φιλτράρει μόνο όσα είναι στο διάστημα από την ημερομηνία DATE FROM της cached επερώτησης μέχρι την ημερομηνία DATE TO της υποβαλλόμενης επερώτησης. Επιπρόσθετα, μια καινούργια διαδικασία αναζήτησης θα πρέπει να εκτελεστεί για όλα τα νέα άρθρα που δεν υπάρχουν στην cache και αφορούν την χρονική περίοδο από την ημερομηνία DATE FROM της υποβαλλόμενης επερώτησης έως και πριν την ημερομηνία DATE FROM της cached επερώτησης. Παράλληλα με την παρουσίαση του τελικού αποτελέσματος στο χρήστη, χρειάζεται να ενημερωθεί και η ημερομηνία DATE FROM στην cache για να ανταποκρίνεται στην καινούργια χρονική περίοδο που προέκυψε.
- **Περίπτωση 4:** η περίπτωση αυτή είναι παρόμοια με την προηγούμενη μόνο που είναι η αντίστροφη. Για το τελικό αποτέλεσμα θα χρειαστεί και πάλι φιλτράρισμα των δεδομένων που υπάρχουν στην cache καθώς και επιπλέον αναζήτηση για νέα άρθρα στο διάστημα DATE TO της cached επερώτησης έως την ημερομηνία DATE TO της υποβαλλόμενης επερώτησης. Προφανώς, με παρόμοιο αλλά αντίστροφο τρόπο με την τρίτη περίπτωση θα χρειαστεί και ενημέρωση της ημερομηνίας DATE TO στην cache.

Στο σημείο αυτό θα πρέπει να παρατηρήσουμε ότι για τα άρθρα που είναι αποθηκευμένα στην cache από προγενέστερες επερωτήσεις χρησιμοποιείται ένας μηχανισμός «λήξης» τους (expiration mechanism). Κάθε cached επερώτηση είναι έγκυρη και χρησιμοποιείται από τον αλγόριθμο για ένα σχετικά μικρό χρονικό διάστημα (λίγες ημέρες συνήθως), έτσι ώστε η έξοδος του συστήματος προς το χρήστη, τα τελικά άρθρα του αποτελέσματος δηλαδή, να παραμένουν όσο το δυνατόν πιο «φρέσκα» και έγκυρα γίνεται. Αυτό υλοποιείται ως εξής: οποτεδήποτε ο αλγόριθμος εκτελεί αναζήτηση στην cache για αποτελέσματα από προηγούμενες επερωτήσεις που είναι πανομοιότυπες με την υποβαλλόμενη επερώτηση, σβήνονται όσες επερωτήσεις έχουν «λήξει» και αντικαθίστανται από νέες. Λόγω αυτού του τρόπου που υλοποιείται ο μηχανισμός «λήξης» των cached αποτελεσμάτων, είναι πιθανό για την ίδια επερώτηση να υπάρχουν στην cache αρκετές εγγραφές, για μη επικαλυπτόμενα χρονικά διαστήματα φυσικά, εφόσον κανένα από αυτά δεν έχει λήξει. Η επιλογή της χρονικής περιόδου μετά την παρέλευση της οποίας «λήγουν» και διαγράφονται τα cached δεδομένα θα αναλυθεί πειραματικά στο επόμενο κεφάλαιο. Στη συνέχεια παρουσιάζεται με ψευδοκώδικα η λειτουργία του αλγόριθμου που περιγράψαμε και ο οποίος υλοποιεί το caching.

```
Algorithm Search_Cached(query)
match = Search_In_Cache(query);
If(Is_Found(match))
   expired = Check_Expiration(match);
   If(expired==true)
          Delete_From_Cache(match);
          results[] = Execute_New_Search(query);
          Insert_In_Cache(results[]);
   Else
          Check_Case(1):
                 results[] = Fetch_Results(match);
```

```
                    results[] = Filter(results[]);
                    Check_Case(2):
                            results[] = Fetch_Results(match);
                            results.append(Execute_New_Search_Before());
                            results.append(Execute_New_Search_After());
                    Update_Cache(results[]);
            Check_Case(3):
                    results[] = Fetch_Results(match);
                    results = Filter(results[]);
                            results.append(Execute_New_Search_Before());
                            Update_Cache(results[]);
                    Check_Case(4):
                    results[] = Fetch_Results(match);
                    results = Filter(results[]);
                            results.append(Execute_New_Search_After());
                            Update_Cache(results[]);
      Endif
   Else
      results[] = Execute_New_Search(query);
      Insert_In_Cache(results[]);
   Endif
   End Algorithm
```

Αλγόριθμος 5: Λειτουργία του αλγόριθμου για αναζήτηση σε cached αποτελέσματα

Εξετάζοντας αυτόν το αλγόριθμο που λειτουργεί στον εξυπηρετητή, μπορούμε να διαπιστώσουμε ότι και στις τέσσερις περιπτώσεις που αναλύθηκαν προηγουμένως, πετυχαίνουμε να μειώσουμε σε μεγάλο βαθμό το χρόνο που χρειάζεται για την αναζήτηση των άρθρων σε μια επερώτηση του χρήστη αντικαθιστώντας την χρονοβόρα διαδικασία της αναζήτησης με αυτή της ανάκτησης των cached αποτελεσμάτων και του φιλτραρίσματος όσων ανήκουν στην επιθυμητή χρονική περίοδο. Όπως αναφέρθηκε το φιλτράρισμα αυτό επιτυγχάνεται εύκολα αφού τα cached αποτελέσματα μαζί με μεταδεδομένα υπάρχουν σε μοφρή XML και η διαδικασία του parsing αυτής της XML δεν μπορεί να θεωρηθεί υπολογιστικά απαιτητική για τον εξυπηρετητή.

Η πιο σημαντική βελτίωση υπάρχει στην πρώτη περίπτωση, όπου δεν εκτελείται καμία νέα αναζήτηση και όλα τα αποτελέσματα του τελικού συνόλου των άρθρων ανακτώνται απευθείας από την cache. Αυτό, είναι ένα ιδιαίτερα σημαντικό ώφελος για την υλοποίηση μας καθώς και είναι και η πιο συνηθησμένη περίπτωση, όπου ο χρήστης μπορεί να υποβάλλει την ίδια ακριβώς επερώτηση επαναλαμβανόμενα χωρίς να αλλάζει καθόλου τις ημερομηνίες DATE FROM και DATE TO στην επερώτηση που υποβάλλει. Η χειρότερη περίπτωση είναι η δεύτερη, όπου ο χρήστη επεκτείνει τη χρονική περίοδο της αναζήτησης (και προς τις δύο κατευθύνσεις – πριν και μετά τα cached δεδομένα) με αποτέλεσμα να χρειάζεται να γίνουν επιπλέον δύο διαφορετικές αναζητήσεις ακολουθούμενες από ενημέρωση των δεδομένων που υπάρχουν στο πίνακα της βάσης με τα cached δεδομένα. Ωστόσο, αυτή είναι μάλλον η πιο σπάνια περίπτωση αφού τις περισσότερες φορές ο χρήστης έχει την τάση να συρικνώνει την χρονική περίοδο του ζητούμενου αποτελέσματος προκειμένου να εστιάσει καλύτερα χρονικά στο αποτέλεσμα του. Στις δύο τελευταίες περιπτώσεις, μια διαδικασία αναζήτησης εκτελείται κάθε φορά και μια ενημέρωση συμβαίνει στη βάση δεδομένων. Αυτό συνεπάγεται ότι μπορούμε να εξοικονομήσουμε παραπάνω από το 50% του υπολογιστικού φορτίου όταν η επέκταση των χρονικών ορίων της επερώτησης δεν είναι μεγαλύτερη από το ίδιο το χρονικό διάστημα για το οποίο υπάρχουν cached δεδομένα.

9

ΤΟ ΣΥΣΤΗΜΑ ΣΕ ΠΛΗΡΗ ΛΕΙΤΟΥΡΓΙΑ

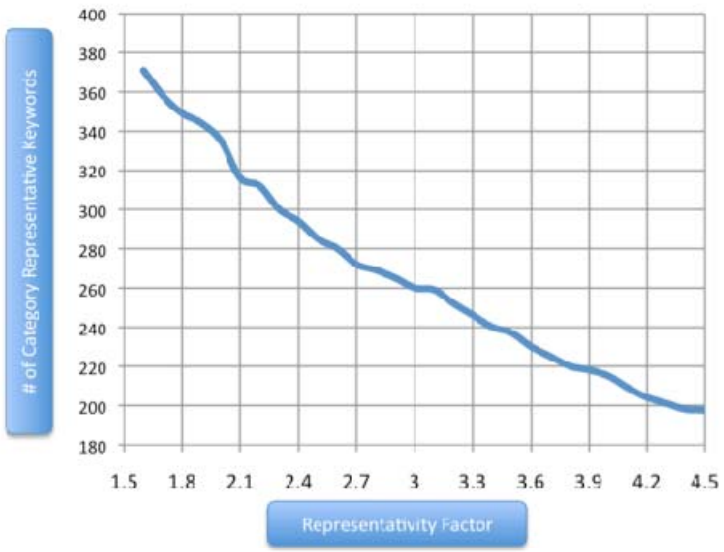
Στο κεφάλαιο αυτό περιγράφονται τα πειράματα που έγιναν στο σύστημα καθώς αυτό είναι σε πλήρη λειτουργία.

9. ΤΟ ΣΥΣΤΗΜΑ ΣΕ ΠΛΗΡΗ ΛΕΙΤΟΥΡΓΙΑ

Η ανάπτυξη του συστήματος που έγινε στα πλαίσια της παρούσας εργασίας έγινε τμηματικά με κάθε module αυτού να αναπτύσσεται ξεχωριστά από τα υπόλοιπα. Την ανάπτυξη του καθενός τμήματος ακολουθούσε και μια διαδικασία αξιολόγησης του ώστε: α) να εντοπισθεί η αποτελεσματικότητά του ως ξεχωριστή οντότητα και β)να προσδιοριστούν οι απαραίτητες παράμετροι που πρέπει να χρησιμοποιηθούν σε κάθε βήμα ώστε ο μηχανισμός, ως σύνολο, να παράγει το βέλτιστο αποτέλεσμα. Ακολουθεί μια αναλυτική παρουσίαση των πειραματικών διαδικασιών και αξιολογήσεων που έλαβαν μέρος και που αφορούν στα βασικά υποσυστήματα του μηχανισμού: τη διαδικασία επιλογής της θέσης των άρθρων του αποτελέσματος, την προσωποποίηση στο χρήστη, την βελτιστοποίηση του υποσυστήματος αναζήτησης με χρήση caching μηχανισμού καθώς και τις μεταξύ τους αλληλεπιδράσεις.

9.1. Συντελεστής Αντιπροσωπευτικότητας

Κατά τη διαδικασία επέκτασης της επερώτησης που υπέβαλλε ο χρήστης, καταφέραμε να εμπλουτίσουμε την επερώτηση χρησιμοποιώντας έναν παράγοντα που ονομάσαμε συντελεστή αντιπροσωπευτικότητας. Αυτός ο αριθμός, όπως περιγράφηκε στους αλγόριθμους του προηγούμενου κεφαλαίου, αναπαριστά την πιθανότητα που έχει ο χρήστης να χρησιμοποιήσει κωδικολέξεις οι οποίες είναι αρκετά αντιπροσωπευτικές της θεματικής κατηγορίας που έχει επιλέξει για τα άρθρα του τελικού αποτελέσματος κατά την διαμόρφωση της αρχικής επερώτησης. Όταν ο χρήστης επιλέξει τέτοιες αντιπροσωπευτικές κωδικολέξεις στην επερώτηση του, το σύστημα επεκτείνει την επερώτηση, εμπλουτίζοντάς την με περισσότερες κωδικολέξεις από την θεματική κατηγορία στην οποία διαπιστώθηκε ότι υπήρχαν αντιπροσωπευτικές κωδικολέξεις. Στην επόμενη γραφική παράσταση παρουσιάζουμε τη σχέση που υπάρχει ανάμεσα στον συντελεστή αντιπροσωπευτικότητας και τον αριθμό των κωδικολέξεων που είναι αποθηκευμένες στη βάση δεδομένων του συστήματος και οι οποίες καθίστανται αντιπροσωπευτικές μιας κατηγορίας.



Εικόνα 13: Συντελεστής Αντιπροσωπευτικότητας και πλήθος κωδικολέξεων που είναι αντιπροσωπευτικές μιας θεματικής ενότητας.

Για την κατασκευή αυτού του γραφήματος αρχικά ανακτήσαμε για κάθε μία από τις 7 θεματικές κατηγορίες του συστήματος τις 100 κωδικολέξεις που παρουσίαζαν την μεγαλύτερη συχνότητα εμφάνισης σε αυτήν την κατηγορία. Στους παρακάτω πίνακες φαίνονται ενδεικτικά για δύο κατηγορίες (Business και Sports) οι 10 κωδικολέξεις με τις μεγαλύτερες συχνότητες εμφάνισης.

Αναγνωριστικό κωδικολέξης	Ρίζα κωδικολέξης	Σχετική Συχνότητα	Απόλυτη Συχνότητα
410	percent	0.0214501	10724
349	price	0.0176078	8803
279	compani	0.0130293	6514
344	market	0.0128853	6442
1761	bank	0.0121472	6073
1926	share	0.0105051	5252
1616	quarter	0.00784278	3921
342	cent	0.00742874	3714
1526	rate	0.00736674	3683
1711	sale	0.00668667	3343

Πίνακας 6: Κωδικολέξεις με μεγαλύτερες συχνότητες για την κατηγορία business

Αναγνωριστικό κωδικολέξης	Ρίζα κωδικολέξης	Σχετική Συχνότητα	Απόλυτη Συχνότητα
146	game	0.0209057	4408
91	play	0.0136826	2885
650	season	0.0129143	2723
57	team	0.0105382	2222
115	inning	0.00933356	1968
52	player	0.00893992	1885
90	pitch	0.0073037	1540
899	wood	0.00679624	1433
207	round	0.00679149	1432
205	shot	0.0065259	1376

Πίνακας 7: Κωδικολέξεις με μεγαλύτερες συχνότητες για την κατηγορία sports

Από τις κωδικολέξεις που συλλέξαμε με αυτόν τον τρόπο, απομακρύναμε αυτές που εμφανίζονταν δύο ή περισσότερες φορές και καταλήξαμε τελικά με 490 διαφορετικές κωδικολέξεις. Όπως βλέπουμε από τη γραφική παράσταση η πιθανότητα να διαλέγει ο χρήστης κωδικολέξεις αντιπροσωπευτικές κάποιας θεματικής κατηγορίας είναι σχετικά υψηλή, σχεδόν 70%, όταν ο συντελεστής αντιπροσωπευτικότητας λαμβάνει τιμές μικρότερες του 2, ενώ μειώνεται στο 50% όταν ο συντελεστής αντιπροσωπευτικότητας για τιμές άνω του 3.5. Αυτό έχει ως αποτέλεσμα να είναι κάπως δυσκολότερο για το χρήστη να επιλέγει αντιπροσωπευτικές κωδικολέξεις και συνεπώς λιγότερες επερωτήσεις εμπλουτίζονται με επιπλέον κωδικολέξεις ανάμεσα σε αυτές που υποβάλλονται στη μηχανή αναζήτησης.

Να υπενθυμίσουμε σε αυτό το σημείο ότι οι επιπλέον κωδικολέξεις με τις οποίες εμπλουτίζεται μια επερώτηση δε συμμετέχουν στη απόκτηση των άρθρων του τελικού αποτελέσματος που θα παρουσιαστεί στον τελικό χρήστη αλλά ο ρόλος τους είναι να βοηθήσουν στην διαμόρφωση της σειρά με την οποία θα εμφανιστούν τα τελικά άρθρα στο χρήστη. Να σημειώσουμε επίσης ότι για τον συντελεστή αντιπροσωπευτικότητας του πραγματικού συστήματος κάνεαμε τις ακόλουθες επιλογές

- δεν χρησιμοποιήσαμε ιδιαίτερα χαμηλές τιμές - κάτω του 2.5 διότι κάτι τέτοιο θα καθιστούσε την επέκταση και τον εμπλουτισμό των

υποβαλλόμενων επερωτήσεων αρκετά συχνό φαινόμενο με αποτέλεσμα τα άρθρα στο τελικό αποτέλεσμα να διατάσσονται βάσει κωδικολέξεων που ο χρήστης δεν είχε καν σκεφτεί ή πιθανόν να μην αντιπροσωπεύουν σε μεγάλο βαθμό το επιθυμητό για αυτόν αποτέλεσμα.

- δεν χρησιμοποιήσαμε υψηλές τιμές άνω του 3.5 καθώς η επέκταση των υποβαλλόμενων επερωτήσεων θα ήταν αρκετά σπάνια με αποτέλεσμα τα άρθρα της εξόδου να μην εστιάζουν σε συγκεκριμένες θεματικές ενότητες αλλά να υπάρχει μεγάλη διασπορά σε πολλές από τις θεματικές κατηγορίες του συστήματος.

9.2. Προσωποποιημένη & Μη προσωποποιημένη Αναζήτηση – Πειράματα & Αξιολόγηση

Για να συγκρίνουμε τα αποτελέσματα μιας προσωποποιημένης με μια μη προσωποποιημένη αναζήτηση εκτελέσαμε πειράματα για τρεις διαφορετικούς εικονικούς χρήστες που δημιουργήσαμε στο σύστημα. Κατα τη διάρκεια της εγγραφής αυτών των εικονικών χρηστών στο σύστημα, αποδώσαμε σε κάθε ένα από αυτούς διαφορετικά χαρακτηριστικά και προτιμήσεις. Σε κάθε χρήστη αποδόθηκε μια θετική προτίμηση για μια θεματική κατηγορία και αρνητική προτίμηση για όλες τις υπόλοιπες κατηγορίες. Σκοπός ήταν τα αποτελέσματα που θα λάβουμε να είναι πιο εύκολα στην ανάλυση και παρακολούθηση ενώ παράλληλα να αποτελούν και μια ρεαλιστική απεικόνιση των διαφορετικών ομάδων χρηστών που είναι εγγαγραμμένοι στο σύστημα μας. Στους πίνακες και στις γραφικές παραστάσεις που ακολουθούν θεωρούμε τους παρακάτω χρήστες:

- Χρήστης Α: θετική προτίμηση στην κατηγορία των αθλητικών νέων (sports) και αρνητική προτίμηση σε όλες τις άλλες κατηγορίες
- Χρήστης Β: θετική προτίμηση στην κατηγορία των επιχειρηματικών νέων (business) και αρνητική προτίμηση σε όλες τις υπόλοιπες κατηγορίες
- Χρήστης C: θετική προτίμηση στην κατηγορία των τεχνολογικών νέων και αρνητική προτίμηση σε όλες τις υπόλοιπες κατηγορίες.

Να σημειώσουμε εδώ ότι το γεγονός ότι ένας χρήστης έχει θετική προτίμηση σε μια συγκεκριμένη κατηγορία και αρνητική προτίμηση στις υπόλοιπες κατηγορίες δεν τον αποκλείει από το να του παρουσιαστούν στο τελικό αποτέλεσμα άρθρα από τις κατηγορίες με την αρνητική προτίμηση, απλά η μηχανή αναζήτησης έχει την "τάση" να επιλέγει για αυτόν άρθρα που να είναι πιο κοντά στην κατηγορία της θετικής προτίμησης, προσπαθώντας με αυτόν τον τρόπο να προσαρμοστεί όσο το δυνατόν περισσότερο στο προφίλ του. Φυσικά, όπως αναφέραμε και στο προηγούμενο κεφάλαιο, είναι δυνατόν το προφίλ ενός χρήστη να αλλάζει δυναμικά με την πάροδο των συνεδριών του (user sessions) μέσα στο σύστημα πράγμα το οποίο με τη σειρά του αναδιαμορφώνει και την προτίμηση του χρήστη για τις διαφορετικές θεματικές κατηγορίες που προτιμά. Αν δηλαδή, ο εικονικός χρήστης Α που δημιουργήσαμε πιο πάνω παρόλη την αρχική δήλωση προτίμησης για την κατηγορία των αθλητικών νέων, μέσα από τις συνεδρίες του αλλάξει το ενδιαφέρον του στα νέα που αφορούν την υγεία (health) τότε σταδιακά και το ίδιο το σύστημα θα τον αντιμετωπίζει σαν ένα χρήστη ο οποίος κατά την αρχική διαμόρφωση του προφίλ του είχε επιλέξει θετική προτίμηση για τον τομέα της υγείας.

Για τα πειράματά μας, χρησιμοποιήσαμε επερωτήσεις που αποτελούνται από κωδικολέξεις που είναι περισσότερο γενικές και δεν αντιπροσωπεύουν ή δεν πλησιάζουν περισσότερο μια κατηγορία από ότι πλησιάζουν άλλες κατηγορίες. Παραδείγματα τέτοιων κωδικολέξεων που επιλέξαμε είναι "Sunday", "New York", "red", "people".

Στην πρώτη φάση των πειραμάτων, εκτελούμε μια μη προσωποποιημένη αναζήτηση για έναν ανώνυμο χρήστη σε μια ουδέτερη επερώτηση. Τα πρώτα 60

άρθρα που επιστρέφονται από τη μηχανή αναζήτησης παρουσιάζονται στον επόμενο πίνακα.

Θέση Άρθρου	Κατηγορία	Σχετικότητα	Θέση Άρθρου	Category	Relevance	Art. Rank	Category	Relevance
1	4	0.836	21	4	0.597	41	2	0.340
2	1	0.822	22	1	0.562	42	3	0.332
3	1	0.807	23	2	0.559	43	3	0.284
4	4	0.782	24	3	0.556	44	4	0.284
5	2	0.766	25	2	0.545	45	1	0.269
6	5	0.762	26	2	0.533	46	6	0.262
7	3	0.760	27	5	0.532	47	1	0.213
8	2	0.743	28	1	0.530	48	2	0.199
9	3	0.735	29	6	0.528	49	7	0.178
10	4	0.732	30	5	0.525	50	4	0.166
11	6	0.707	31	3	0.519	51	1	0.156
12	1	0.706	32	4	0.515	52	2	0.117
13	6	0.702	33	2	0.487	53	1	0.111
14	3	0.632	34	6	0.476	54	1	0.091
15	2	0.624	35	2	0.474	55	3	0.085
16	2	0.622	36	1	0.468	56	3	0.067
17	4	0.617	37	2	0.427	57	1	0.065
18	7	0.609	38	3	0.411	58	5	0.056
19	1	0.604	39	3	0.410	59	5	0.033
20	2	0.602	40	1	0.369	60	7	0.008

Πίνακας 8: Μη προσωποποιημένη Αναζήτηση – Θέσεις, Κατηγορίες και Βαθμοί Σχετικότητας των ανακτημένων άρθρων

Για λόγους απλότητας, δεν εμφανίζεται το αναγνωριστικό κάθε άρθρου που υπαρχει στο πίνακα articles της βάσης δεδομένων αλλά τη σειρά με την οποία επέστρεψε από την αναζήτηση. Η σειρά αυτή θα χρησιμεύσει για τη σύγκριση της αυτού του είδους αναζήτησης με την προσωποποιημένη αναζήτηση που θα εκτελέσουμε στη συνέχεια. Για κάθε άρθρο του αποτελέσματος επίσης παρουσιάζουμε την κατηγορία στην οποία ανήκει με τη μεγαλύτερη συχνότητα καθώς και τη σχετικότητα του με την επερώτηση όπως αυτή υπολογίζεται από τον τύπο ... του προηγούμενου κεφαλαίου.

Όπως βλέπουμε από τα αποτελέσματα του πίνακα, εξαιτίας της γενικότητας των επερωτήσεων που επιλέξαμε, υπάρχει διασπορά των άρθρων του αποτελέσματος στις περισσότερες από τις κατηγορίες του συστήματος. Αυτό είναι και λογικό, μιας και επιτρέπει σε διαφορετικούς χρήστες να βρίσκουν άρθρα της αρεσκείας τους. Για το παραπάνω πείραμα, τα άρθρα που οι τρεις εικονικοί χρήστες A,B και C επέλεξαν να δουν συνοψίζονται στον ακόλουθο πίνακα, όπου δικαιολογημένα και με βάση τη διαμόρφωση του αρχικού του προφίλ, ο κάθε χρήστης επιλέγει άρθρα που είναι πιο σχετικά με την κατηγορία στην οποία και δήλωσε θετική προτίμηση.

Χρήστης	Αναγνωριστικά άρθρων από κάθε κατηγορία
A	Sports(8,20,33,48) Business(12,54)
B	Business(12,19,40,45,57) Entertainment(18)
C	Technology(31,32,44) Health(7) Science(27) Business(19)

Πίνακας 9: Πειραματικοί Χρήστες για την αξιολόγηση της προσωποποιημένης Αναζήτησης.

Στη δεύτερη φάση του πειράματος, εκτελούμε μια προσωποποιημένη αναζήτηση για κάθε ένα από τους τρεις χρήστες διαφορετικά αυτή τη φορά, χρησιμοποιώντας το ίδιο σετ επερωτήσεων που χρησιμοποιήσαμε και στη μη προσωποποιημένη αναζήτηση. Αυτή τη φορά επιλέγουμε για κάθε χρήστη τα πρώτα 15 από τα άρθρα που του παρουσιάστηκαν στο τελικό αποτέλεσμα με τη σειρά με την οποία του παρουσιάστηκαν. Για την αξιολόγηση των αποτελεσμάτων του πίνακα που ακολουθεί παρουσιάζουμε και πάλι τη σειρά με την οποία παρουσιάστηκε κάθε άρθρο (Personalized Rank) καθώς επίσης και τη σειρά με την οποία το συγκεκριμένο άρθρο εμφανίστηκε στην μη προσωποποιημένη αναζήτηση του προηγούμενου πειράματος (Article Non Personalized (NP) Rank). Τα άρθρα που δεν παρουσιάστηκαν καθόλου στη μη προσωποποιημένη αναζήτηση εμφανίζονται ως "New" σε αυτόν τον πίνακα. Τα άρθρα τα οποία είναι τονισμένα με διαφορετικό χρώμα είναι αυτά που ο κάθε χρήστης είχε επιλέξει από το τελικό αποτέλεσμα της μη προσωποποιημένης αναζήτησης πριν.

Σειρά στην προσωπ/μένη αναζήτηση	Σειρά στη μη προσωπ/μένη αναζήτηση	Σχετικότητα στην προσωπ/μένη αναζήτηση	Σχετικότητα στη μη προσωπ/μένη αναζήτηση	Κατηγορία Άρθρου
1	5	0.846	0.766	Sports
2	8	0.839	0.743	Sports
3	12	0.833	0.706	Business
4	14	0.824	0.632	Health
5	3	0.779	0.807	Business
6	33	0.776	0.487	Sports
7	New	0.763	0.000	Sports
8	39	0.757	0.410	Health
9	35	0.738	0.474	Sports
10	7	0.738	0.000	Health
11	28	0.735	0.530	Business
12	New	0.734	0.000	Sports
13	18	0.730	0.609	Entertainment
14	20	0.729	0.602	Sports
15	6	0.698	0.762	Science

Πίνακας 10: Χρήστης Α – Σειρά άρθρων στην προσωποποιημένη αναζήτηση συγκρινόμενη με αυτή της μη προσωποποιημένης αναζήτησης

Σειρά στην προσωπ/μένη αναζήτηση	Σειρά στη μη προσωπ/μένη αναζήτηση	Σχετικότητα στην προσωπ/μένη αναζήτηση	Σχετικότητα στη μη προσωπ/μένη αναζήτηση	Κατηγορία Άρθρου
1	22	0.901	0.562	Business
2	40	0.871	0.369	Business
3	21	0.824	0.597	Technology
4	35	0.801	0.474	Sports
5	12	0.796	0.706	Business
6	28	0.788	0.530	Business
7	1	0.751	0.836	Technology
8	18	0.723	0.609	Entertainment
9	New	0.717	0.000	Business
10	New	0.715	0.000	Sports
11	6	0.699	0.762	Science
12	18	0.680	0.609	Entertainment
13	New	0.658	0.000	Technology
14	45	0.649	0.269	Business

Σειρά στην προσωπ/μένη αναζήτηση	Σειρά στη μη προσωπ/μένη αναζήτηση	Σχετικότητα στην προσωπ/μένη αναζήτηση	Σχετικότητα στη μη προσωπ/μένη αναζήτηση	Κατηγορία Άρθρου
15	New	0.640	0.000	Sports

Πίνακας 11: Χρήστης Β – Σειρά άρθρων στην προσωποποιημένη αναζήτηση συγκρινόμενη με αυτή της μη προσωποποιημένης αναζήτησης

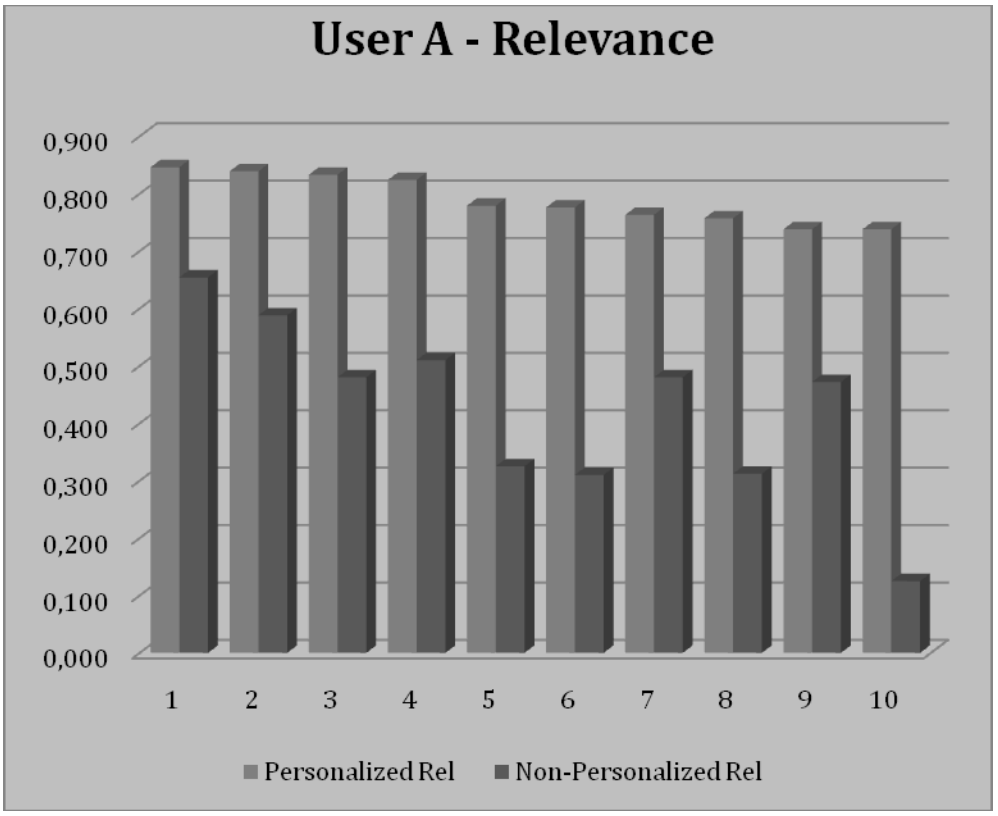
Σειρά στην προσωπ/μένη αναζήτηση	Σειρά στη μη προσωπ/μένη αναζήτηση	Σχετικότητα στην προσωπ/μένη αναζήτηση	Σχετικότητα στη μη προσωπ/μένη αναζήτηση	Κατηγορία Άρθρου
1	21	0.928	0.597	Technology
2	7	0.894	0.760	Health
3	New	0.877	0.000	Technology
4	19	0.830	0.604	Business
5	44	0.821	0.284	Technology
6	49	0.819	0.178	Entertainment
7	17	0.782	0.617	Business
8	30	0.760	0.525	Science
9	32	0.754	0.515	Technology
10	56	0.753	0.067	Health
11	10	0.741	0.732	Technology
12	36	0.722	0.468	Business
13	31	0.716	0.836	Technology
14	3	0.691	0.807	Technology
15	New	0.680	0.000	Technology

Πίνακας 12: Χρήστης Γ – Σειρά άρθρων στην προσωποποιημένη αναζήτηση συγκρινόμενη με αυτή της μη προσωποποιημένης αναζήτησης

Με μια γρήγορη ματιά στους παραπάνω πίνακες, μπορούμε να διαπιστώσουμε ότι τα περισσότερα από τα άρθρα που οι τρεις εικονικοί χρήστες είχαν επιλέξει αρχικά, στη μη προσωποποιημένη αναζήτηση, παρουσιάζονται στην περίπτωση της προσωποποιημένης αναζήτησης με καλύτερη σειρά (υψηλότερη θέση) και με μεγαλύτερη σχετικότητα ως προς την επερώτηση του χρήστη. Στις δύο επόμενες γραφικές παραστάσεις φαίνεται η σύγκριση των αποτελεσμάτων των δύο πειραμάτων για τον χρήστη Α.

Παρατηρούμε ότι 4 από τα 6 άρθρα που είχε επιλέξει στη μη προσωποποιημένη αναζήτηση από τα πρώτα 60 αποτελέσματα έχουν μετακινηθεί αρκετές θέσεις παραπάνω στην περίπτωση της προσωποποιημένης αναζήτησης ενώ τρία από αυτά έχουν καταλήξει στα πρώτα 10 αποτελέσματα.

Ένα σημαντικό σημείο που πρέπει να τονιστεί είναι ότι με την πάροδο του χρόνου, για χρήστες οι οποίοι παρουσιάζουν μια "συνέπεια" στην επιλογή της θεματικής κατηγορίας των άρθρων που επισκέπτονται, τα αποτελέσματα είναι ακόμα πιο εστιασμένα στις κατηγορίες αυτές καθώς οι κωδικολέξεις των εν λόγω κατηγοριών αποκτούν αρκετά μεγάλη πόλωση στον πίνακα "user_website_keyword" της βάσης δεδομένων. Στην επόμενη γραφική παράσταση , μπορούμε να δούμε τη σχετικότητα με την οποία τα πρώτα 10 άρθρα επιστρέφονται τόσο στη μη προσωποποιημένη όσο και στην προσωποποιημένη αναζήτηση. Διαφαίνεται ότι στην περίπτωση της προσωποποιημένης αναζήτησης ο χρήστης ανακτά άρθρα με πολύ μεγαλύτερο βαθμό σχετικότητας με την επερώτηση την οποία υπέβαλλε στο σύστημα στις πρώτες θέσεις του αποτελέσματος.



Εικόνα 14. Σχετικότητα των 10 πρώτων άρθρων του αποτελέσματος σε προσωποποιημένη και μη προσωποποιημένη αναζήτηση

Για την περίπτωση ενός ανώνυμου χρήστη, ο αλγόριθμος του πραγματικού συστήματος εξομοιώνει την διαδικασία προσωποποιημένης αναζήτησης. Η μηχανή αναζήτησης μπορεί να επιλέξει να του παρουσιάσει αποτελέσματα τέτοια σαν να εκτελούσε μια προσωποποιημένη αναζήτηση με το να προσπαθεί να ταιριάξει την υποβαλλόμενη επερώτηση με μια συγκεκριμένη κατηγορία, οποτεδήποτε αυτό είναι εφικτό. Στη συνέχεια, εκτελείται αναζήτηση όπως αν ο χρήστης είχε ένα συγκεκριμένο προφίλ με προτίμηση στη συγκεκριμένη κατηγορία που υπολογίστηκε αμέσως πριν. Φυσικά σε αυτήν την περίπτωση συμπεριφοράς του συστήματος, δεν είναι δυνατή η εκπαίδευση της μηχανής από το χρήστη μιας και δεν υπάρχει στην πραγματικότητα ένα προφί με προτιμήσεις οι οποίες μπορούν να εξελιχθούν, ωστόσο, μπορούμε ακόμα να επωφεληθούμε καθώς τα αποτελέσματα είναι περισσότερο εστιασμένα στον θεματικό προσανατολισμό της επερώτησης και προκύπτει μεγαλύτερη πιθανότητα ο χρήστης να ικανοποιηθεί με το τελικό αποτέλεσμα.

9.3. Πειραματική Αξιολόγηση του αλγόριθμου Caching

Στο προηγούμενο κεφάλαιο έγινε η περιγραφή και η ανάλυση του αλγόριθμου που χρησιμοποιήσαμε στο σύστημα μας για αποθήκευση (caching) των αποτελεσμάτων από αναζητήσεις των χρηστών με σκοπό τη μελλοντική τους χρησιμοποίηση σε παρόμοιες αναζητήσεις για βελτίωση της ταχύτητας και της απόδοσης του συστήματος. Σε αυτήν την παράγραφο θα περιγράψουμε το πείραμα που διεξάγαμε προκειμένου να αξιολογήσουμε τη συμπεριφορά του αλγόριθμου που υλοποιήσαμε και να αναλύσουμε τη βελτίωση που επιφέρει στην απόδοση του συστήματος.

Στο πείραμά μας, κατασκευάσαμε έναν εικονικό χρήστη ο οποίος υποβάλλει επερωτήσεις στο σύστημα για την αναζήτηση άρθρων. Οι επερωτήσεις που υποβάλλονται αποτελούνται από κωδικολέξεις που αντιπροσωπεύουν διαφορετικές θεματικές κατηγορίες των άρθρω της βάσης δεδομένων του Perssonal (αθλητικά, επιστήμη, πολιτική κλπ). Επιλέξαμε να αξιολογήσουμε την απόδοση του

αλγόριθμου του caching σε ερωτήσεις που να περιέχουν όχι παραπάνω από τρεις κωδικολέξεις υπό το σκεπτικό το αποτέλεσμα να περιέχει ένα σχετικά μεγάλο αριθμό άρθρων, τέτοιο ώστε η διαδικασία αναζήτησης να διαρκέσει τόσο όσο χρειάζεται ώστε οι μετρήσεις στα πειράματά μας είναι επαρκείς και ικανές να μας οδηγήσουνε σε χρήσιμα συμπεράσματα. Στη συνέχεια θα επικεντρωθούμε στα ακόλουθα σημεία:

- Την απόδοση του αλγόριθμου που εκτελείται στην μνήμη του εξυπηρετητή (server) για τα διαφορετικά σενάρια αντιστοίχησης των χρονικών ορίων που θέτει ο χρήστης στην υποβαλλόμενη επερώτηση με αυτά που είναι αποθηκευμένα στον πίνακα της βάσης που συλλέγει τα cached δεδομένα.
- Τον τρόπο με τον οποίο το πλήθος των αποθηκευμένων άρθρων στην cache επηρρεάζει την ταχύτητα του αλγόριθμου καθώς και τις απαιτήσεις σε μνήμη στον εξυπηρετητή.
- Τον τρόπο με τον οποίο η επιλογή της χρονικής περιόδου ισχύος (ή λήξης) των δεδομένων της cache επηρρεάζει την ποιότητα και την ακρίβεια της τελικής εξόδου του συστήματος στο χρήστη.

9.3.1. Συμπεριφορά και απόδοση του αλγόριθμου

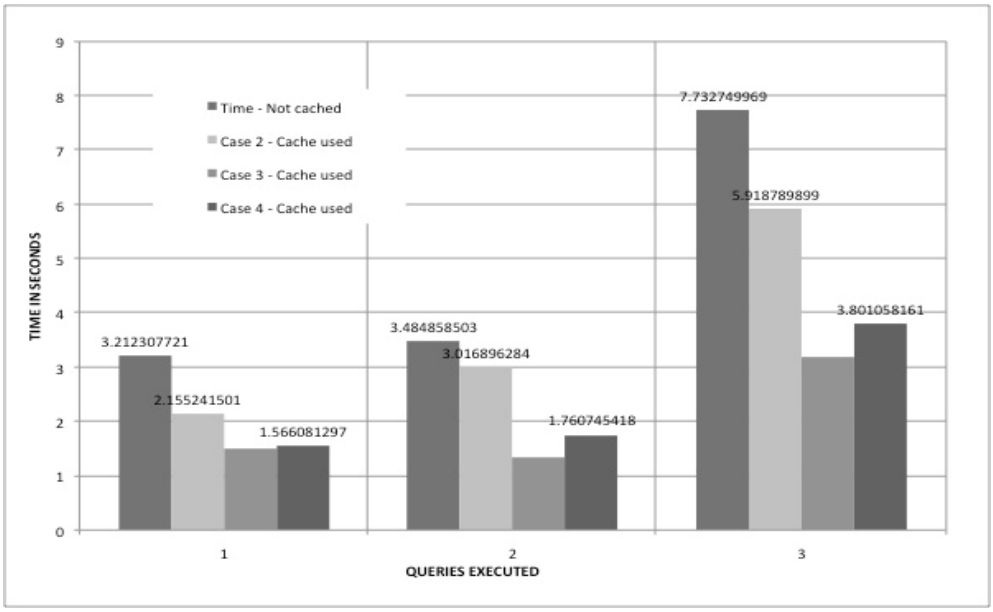
Κατά την διάρκεια των πειραμάτων για την αξιολόγηση της απόδοσης και της συμπεριφοράς του αλγορίθμου του caching, κάναμε δοκιμές με διάφορες επερωτήσεις αναζητώντας άρθρα από διαφορετικές θεματικές κατηγορίες για χρονικό διάστημα έξι μηνών. Στην πρώτη φάση, χρησιμοποιήσαμε κενή μνήμη για την αποθήκευση των αποτελεσμάτων των επερωτήσεων και διαμορφώσαμε τη μηχανή αναζήτησης με τέτοιο τρόπο ώστε να μην εκτελείται αναζήτηση cached αποτελεσμάτων στη βάση δεδομένων. Καθώς ήταν αναμενόμενο, οι επερωτήσεις που αποτελούνταν από αρκετά εξειδικευμένες κωδικολέξεις επεξεργάστηκαν αρκετά γρήγορα. Αυτές οι επερωτήσεις δεν είναι απλά υψηλού ενδιαφέροντος σε ότι αφορά την ανάλυση του αλγορίθμου, καθώς το πλήθος των άρθρων που περιλαμβάνουν τις συγκεκριμένες κωδικολέξεις είναι σχετικώς περιορισμένο και χρειάζεται μικρό υπολογιστικό χρόνο και πόρους για να ανακτηθεί από τη βάση δεδομένων.

Το μεγαλύτερο πρόβλημα υπάρχει στις επερωτήσεις οι οποίες αποτελούνται από κωδικολέξεις μη εξειδικευμένες που μπορούν να απαντώνται σε μεγάλο πλήθος άρθρων από διαφορετικές θεματικές κατηγορίες. Αυτή η κατηγορία επερωτήσεων επιβαρύνει από θέμα υπολογιστικών πόρων και χρόνου περισσότερο το σύστημα και μπορούμε να τη θεωρήσουμε ως μια καλή αφετηρία για την αξιολόγιση της μεθόδου caching που χρησιμοποιήσαμε. Στο σχήμα ...., μπορούμε να δούμε τα αποτελέσματα του αλγόριθμου σε ότι αφορά την επιτάχυνση της διαδικασίας αναζήτησης για τρεις γενικές επερωτήσεις, που αποτελούνται από τις κωδικολέξεις "sports", "computers" και "health body" και οι οποίες επιστρέφουν πάνω από 5000 άρθρα. Αυτή η γραφική παράσταση απεικονίζει το χρόνο σε δευτερόλεπτα που χρειάστηκε η μηχανή αναζήτησης να επιστρέψει τα σχετικά άρθρα από τη βάση δεδομένων. Από τις περιπτώσεις που αναλύσαμε στο προηγούμενο κεφάλαιο κατά την περιγραφή του αλγόριθμου, αυτές που παρουσιάζονται σε αυτή τη γραφική παράσταση είναι οι περιπτώσεις 2, 3 και 4. Να υπενθυμίσουμε ότι σε αυτές τις περιπτώσεις μόνο ένα υποσύνολο των άθρων του αποτελέσματος υπήρχε αποθηκευμένο στην cache μνήμη από προγενέστερες αναζητήσεις, με αποτέλεσμα η μηχανή αναζήτησης να πρέπει σε όλες τις περιπτώσεις να εκκινήσει νέα διαδικασία αναζήτησης ώστε να ανακτήσει τα άρθρα των χρονικών περιόδων που δεν είνα αποθηκευμένα στην cache. Στην περίπτωση 1, όπως αναφέρθηκε, λόγω του ότι όλα τα επιθυμητά άρθρα του αποτελέσματος υπάρχουν στην cache δεν παρουσιάζει ενδιαφέρον στην ανάλυση αφού η μηχανή αναζήτησης δεν χρειάζεται καν να εκτελέσει καινούργια αναζήτηση απλά χρειάζεται να φιλτράρει τα δεδομένα της cache για να κρατήσει μόνο τα άρθρα που εντάσσονται στην χρονική περίδο που ζήτησε ο χρήστης. Στο πείραμα μας, η

χρονική περιόδος, για την οποία τα αποτελέσματα είχαν αρχικά αποθηκευθεί στην cache προτού υποβληθούν οι υπό συζήτηση επερωτήσεις, ήταν ένας τυχαίος αριθμός ημερών ανάμεσα στις 60 και 90 ημέρες. Η πραγματική επερώτηση, με βάση την οποία θα κάνουμε την αξιολόγηση, αναζητά άρθρα που έχουν δημοσιευθεί τις τελευταίες 180 ημέρες. Η επιλογή αυτών των διαστημάτων δεν ήταν τυχαία. Έγινε για να εξασφαλισθεί ότι το σύστημα θα έπρεπε ούτως ή άλλως να αναζητήσεις περισσότερα άρθρα από όσα ήταν ήδη αποθηκευμένα στην cache.

Στα αποτελέσματα που παρουσιάζονται, παρατηρούμε ότι σε ορισμένες περιπτώσεις, το κέρδος έφτασε σχεδόν το μισό του χρόνου που θα χρειαζόταν αν έπρεπε να επεξεργαστούμε τις επερωτήσεις χωρίς να έχουμε αποθηκευμένα και γρήγορα προσβάσιμα δεδομένα από προγενέστερες επερωτήσεις (uncached). Όπως ήταν αναμενόμενο, η χειρότερη απόδοση εντοπίζεται στην δεύτερη περίπτωση, όπου δύο νέες διαδικασίες αναζήτησης πρέπει να εκτελεστούν, μια για το χρονικό διάστημα που προηγείται αυτού για το οποίο άρθρα υπάρχουν στην cache και μία για το χρονικό διάστημα που έπεται. Μετά από αυτή τη διαδικασία προκύπτουν τρία διαφορετικά σετ άρθρων. Πριν τα παρουσιάσουμε στον τελικό χρήστη, πρέπει να τα επαναταξινομήσουμε με βάση το βαθμό σχετικότητας τους με την επερώτηση που υποβλήθηκε κατατάσσοντας στις πρώτες θέσεις τα άρθρα με το μεγαλύτερο βαθμό σχετικότητας. Για τις περιπτώσεις 3 και 4, τα αποτελέσματα παρουσιάζουν, όπως είναι φυσικό και επόμενο, αρκετές ομοιότητες και αναλογίες. Οι ελαφρώς υψηλότεροι χρόνοι που παρατηρούνται στην περίπτωση 4 ενδεχομένως είναι συνέπεια μιας σχετικά υψηλής συγκέντρωσης των επιθυμητών άρθρων στη συγκεκριμένη χρονική περίοδο για την οποία εκτελέστηκε αναζήτηση εκ νέου, μιας και δεν υπήρχαν αποθηκευμένα στην cache άρθρα για την περίδο αυτή σε συνδυασμό με μικρή συγκέντρωση των άρθρων του αποτελέσματος για την χρονική περίοδο που υπάρχουν cached δεδομένα στην βάση δεδομένων του εξυπηρετητή.

Στους χρόνους εκτέλεσης που μετρήθηκαν στα πειράματα, ένας μέσος όρος 0.1 δευτερολέπτων χρειάστηκε για να ανακτηθούν άρθρα από την cache. Αυτός ο χρόνος είναι κατά μέσο όρο σχεδόν 3% του συνολικού χρόνου που χρειάζεται. Ένα άλλο 2% του χρόνου καταναλώνεται σε διαδικασίες επαναταξινόμησης των δύο ή τριών σετ άρθρων που προέκυψαν από τις αναζητήσεις. Η επαναταξινόμηση των άρθρων γίνεται με βάση τη σχετικότητα τους με την επερώτηση με σκοπό την παρουσίαση τους στον τελικό χρήστη από το πιο σχετικό άρθρο στο λιγότερο σχετικό. Λαμβάνοντας υπόψιν τα παραπάνω, μπορούμε να θεωρούμε αναμενόμενο για την 1η περίπτωση του αλγόριθμου του caching να επιτυγχάνει μια επιτάχυνση της τάξης του 95% στην διαδικασία της αναζήτησης. Μετά την πρώτη εκτέλεση αυτών των επερωτήσεων, κάθε επόμενη υποβολή μιας ίδιας επερώτησης διεκπεραιώνεται σε κάτω από 0.1 δευτερόλεπτα. Οποτεδήποτε υπάρχουν προαποθηκευμένα αποτελέσματα στην cache για μια επερώτηση, κάθε επακόλουθη ταυτόσημη επερώτηση η οποία αναζητά άρθρα μέσα στη χρονική περίοδο του προαποθηκευμένου αποτελέσματος θα επεξεργασθεί σε σχεδόν μηδενικό χρόνο. Αυτό προκύπτει από το γεγονός ότι ο μόνος χρόνος που χρειάζεται είναι αυτός για την ανάκτηση των αποτελεσμάτων από την μνήμη προαποθήκευσης αφού ακόμα και για την ταξινόμηση των άρθρων του αποτελέσματος σε αυτήν την περίπτωση δε θα επιβαρυνθεί το σύστημα μιας και υπάρχει ένα μοναδικό σετ άρθρων τα οποία είναι ήδη ταξινομημένα από την πρώτη αναζήτηση που δημιούργησε την καταχώρηση στη μνήμη προαποθήκευσης. Με τον τρόπο αυτό μπορούμε να περιορίσουμε κατά πολύ τους υπολογιστικούς πόρους που απαιτούνται στον εξυπηρετητή για χρονοβόρες επερωτήσεις στο χρόνο που χρειάζεται για την εκτέλεση της επερώτησης μία μόνο φορά, την πρώτη δηλαδή φορά που εκτελούνται. Κάθε επόμενη φορά οι επερωτήσεις επεξεργάζονται και απαντώνται με πρόσβαση στη μνήμη προαποθήκευσης από τον αλγόριθμο που σχεδιάσαμε, διαδικασία η οποία είναι αρκετά γρήγορη και που δε δημιουργεί ιδιαίτερο φορτίο για τον εξηπηρετητή.

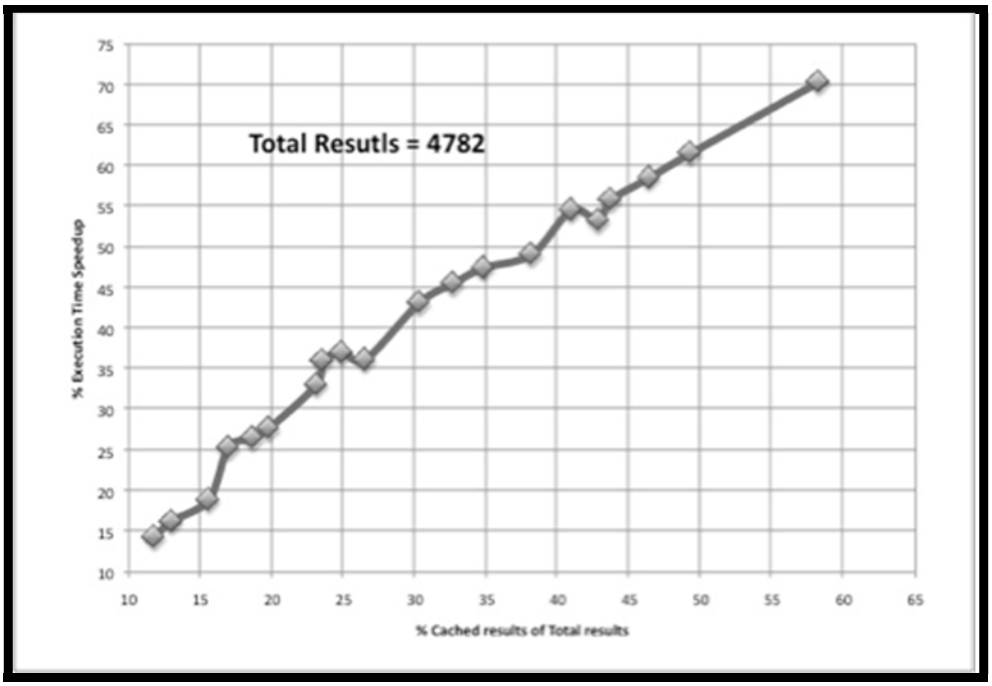


Εικόνα 15: Χρόνος για την ανάκτηση των cached αποτελεσμάτων για κάθε σενάριο επιλογής χρονικού διαστήματος αναζήτησης

9.3.2. Μέγεθος Μνήμης Cache

Το δεύτερο μας μέλημα ήταν η εξέταση του τρόπου με τον οποίο ο αριθμός των άρθρων για κάθε επερώτηση στη μνήμη προαποθήκευσης επηρρεάζει την συνολική απόδοση του αλγόριθμου και το μέγεθος του πίνακα που χρησιμοποιήσαμε για την προαποθήκευση της πληροφορίας στη βάση δεδεομένων. εκτελέσαμε μια μη εξειδικευμένη επερώτηση για αρκετούς αριθμούς προαποθυκευμένων άρθρων-αποτελεσμάτων, αυξάνοντας κάθε φορά το τη χρονική περίοδο για την οποία είχαμε τα προαποθηκευμένα αποτελέσματα. Ο συνολικός αριθμός των άθρων για την επερώτηση που εκτελέσαμε ήταν 4782 άρθρα για μια χρονική περίδιο 4 μηνών. Για το πείραμα αυτό, εξετάσαμε τις περιπτώσεις 2,3 και 4 του αλγόιρθμου μας, έτσι ώστε για κάθε υποβαλλόμενη επερώτηση να μην περιέχονται όλα τα ζητούμενα άρθρα στη μνήμη προαποθήκευσης και η μηχανή αναζήτησης να μη βασίζεται αποκλειστικά και μόνο στην cache και στον αλγόριθμο αλλά να χρειάζεται να εκτελέσει και νέες αναζητήσεις για άρθρα που δεν υπάρχουν σε καταχωρήσεις της μνήμης προαποθήκευσης. Από τη γραφική αναπαράσταση του σχήματος ... που σχετίζει το επι τοις εκατό ποσοστό της επιτάχυνσης του χρόνου με το επι τοις εκατό ποσοστό των προαποθηκευμένων άρθρων στο σύνολο των επιθυμητών άρθρων, μπορούμε να παρατηρήσουμε ότι ο χρόνος εκτέλεσης της αναζήτησης μειώνεται κατά μέσο όρο 50% όταν ελαφρώς λιγότερο ποσοστό απο το 40% των άρθρων της εξόδου υπάρχει στη μνήμη προαποθήκευσης. Καθώς ο συνολικός αριθμός των άρθρων σε αυτό το πείραμα κάλυπτε μια περίοδο περίπου 4 μηνών, μπορούμε να πούμε ότι στατιστικά, το 40% των αποτελεσμάτων θα μπορούσε να ανακτηθεί από μια αναζήτηση σε μια περίοδο μικρότερη των δύο μηνών, η οποία μάλλον είναι μικρότερη χρονική περίοδος από αυτή μιας κοινής αναζήτησης από το μέσο χρήστη. Με αυτό εννοούμε ότι αν ο χρήστης υπέβαλλε αρχικά μια επερώτηση ζητώντας άρθρα για μια περίοδο μεγαλύτερη των δύο μηνών, τότε κάθε επόμενη φορά που θα υπέβαλλε την ίδια επερώτηση μέσα στις επόμενες ημέρες, η μηχανή αναζήτησης θα χρειαζόταν στη χειρότερη περίπτωση το μισό του χρόνου για την εκτέλεση της από αυτό που θα χρειαζόταν αν δεν χρησιμοποιούσαμε μνήμη προαποθήκευσης. Αν στο παραπάνω προσθέσουμε και το γεγονός ότι ο αλγόριθμος ενημερώνει την μνήμη προαποθήκευσης με νέα αποτελέσματα κάθε φορά που μια επεκτεταμένη χρονικά εκδοχή μιας ήδη υπάρχουσας επερώτησης υποβάλλεται στο σύστημα, τότε μπορούμε να καταλάβουμε ότι το ποσοστό των προαποθηκευμένων αποτελεσμάτων όχι μόνο δεν μειώνεται αλλά αυξάνεται γρήγορα και τα

αποτελέσματα του αλγόριθμου στο σύστημα μας συνεχώς βελτιώνονται οδηγώντας σε μικρότερους χρόνου εκτέλεσης επερωτήσεων.



Εικόνα 16: Επίδραση του ποσοστό των cached άρθρων στην επιτάχυνση της αναζήτησης

Όπως παρουσιάστηκε και αναλύθηκε σε προηγούμενο κεφάλαιο, ο αλγόριθμος αποθηκεύει για κάθε επερώτηση στην μνήμη προαποθήκευσης ένα περιορισμένο όγκο πληροφορίας σε σχέση με το πλήθος των άρθρων που ανακτώνται. Η πληροφορία αυτή περιλαμβάνει τα αναγνωριστικά (ids) των άρθρων χωρίς προφανώς το περιεχόμενο τους, τις ημερομηνίες δημοσίευσης τους και τους συντελεστές σχετικότητας με την επερώτηση από την οποία ανακτήθηκαν αρχικά. Αυτό έχει ως συνέπεια το μέγεθος της μνήμης προαποθήκευσης για κάθε καταχώρηση να διατηρείται αρκετά μικρό. Για να δώσουμε ένα πραγματικό παράδειγμα από τη λειτουργία του συστήματος, για την αποθήκευση των σχεδόν 5000 άρθρων της επερώτησης που αναφέρθηκε νωρίτερα και η οποία πρόκειται για μια μη εξειδικευμένη επερώτηση που επιστρέφει ένα σχετικά μεγάλο πλήθος αποτελεσμάτων αφού αρκετά άρθρα κρίνονται συναφή, υπολογίσαμε ότι το μέγεθος της καταχώρησης στη βάση ήταν μικρότερο από 150KB. Αν συνδυάσουμε το μικρό μέγεθος της γραμμής του πίνακα search_caching με την περιοδική διαγραφή καταχωρήσεων που έχουν λήξει σύμφωνα με το μηχανισμό που περιγράψαμε στην παράγραφο ... του προηγούμενου κεφαλαίου, η τεχνική μας μπορεί να εγγυηθεί ικανοποιητικά μικρά μεγέθη μνήμης προαποθήκευσης των επερωτήσεων στο χώρο του εξυπηρετητή. Η επιλογή και η επιρροή του διαστήματος που χρειάζεται για να λήξει μια καταχώρηση στην μνήμη προαποθήκευσης αναλύεται στην επόμενη παράγραφο.

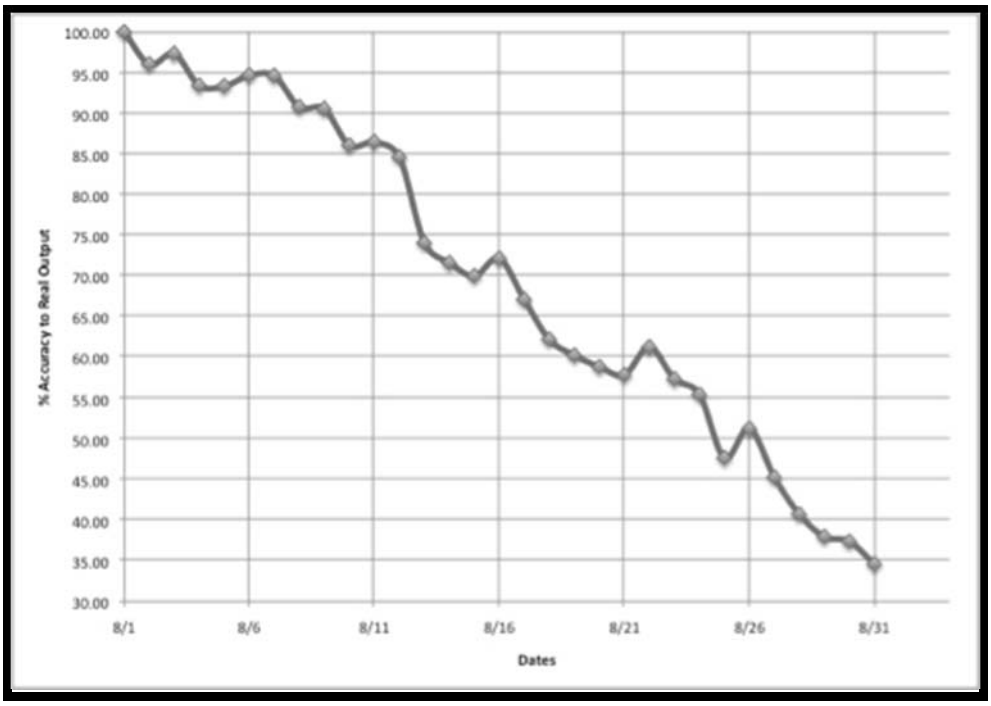
9.3.3. Χρονικό Διάστημα Λήξης & Ακρίβεια στο αποτέλεσμα

Στο τελευταίο στάδιο των πειραμάτων μας, θα εξετάσουμε τον τρόπο με τον οποίο επηρρεάζεται η ποιότητα και η ακρίβεια των αποτελεσμάτων της αναζήτησης από την επιλογή του χρονικού διαστήματος που περνάει από τη στιγμή που καταχωρούνται στη μνήμη προαποθήκευσης τα αποτελέσματα μιας αναζήτησης μέχρι τη στιγμή που λήγουν και διαγράφονται. Όπως αναφέρθηκε στην προηγούμενη παράγραφο, ο αλγόριθμος που υλοποιήσαμε περιοδικά διαγράφει καταχωρήσεις της μνήμης προαποθήκευσης που υπάρχει στη βάση δεδομένων. Η υλοποίηση ενός τέτοιου μηχανισμού περιοδικής διαγραφής (λήξης) είναι

ουσιαστική όχι μόνο διότι συμβάλλει στο να διατηρείται το μέγεθος του πίνακα στη βάση μικρό αλλά επιπλέον, που είναι και το πιο σημαντικό, βελτιώνει κατά πολύ τη ακρίβεια και την ποιότητα του τελικού αποτελέσματος που φτάνει στο χρήστη.

Ο σκοπός μας σε αυτό το τελικό στάδιο των πειραμάτων μας είναι να εξετάσουμε πως η επέκταση του χρονικού διαστήματος λήξης των καταχωρήσεων της μνήμης προαποθήκευσης υποβαθμίζει την ακρίβεια των άρθρων στο τελικό αποτέλεσμα. Για αυτό το λόγο, κατασκευάσαμε έναν εικονικό χρήστη και του αποδώσαμε ένα προφίλ με θεματικές κατηγορίες και κωδικολέξεις στις οποίες να έχει μεγαλύτερη προτίμηση. Αρχικά για τον χρήστη αυτό δεν υπήρχαν καταχωρήσεις στη μνήμη προαποθήκευσης. Στη συνέχεια υποβλήθηκαν από το χρήστη αυτό αρκετές επερωτήσεις στο σύστημα και θέσαμε τον αλγόριθμο προαποθήκευσης σε λειτουργία ώστε να αποθηκεύσει αποτέλεσματα για ορισμένες από τις επερωτήσεις που υποβλήθηκαν. Για τα επόμενα στάδια του πειράματος που απεικονίζουν τις επόμενες εικονικές μέρες του χρήστη στο σύστημα, υποβάλλαμε εκ μέρους του και άλλες επερωτήσεις, αυτή τη φορά χωρίς να αποθηκευουμε κανένα από τα αποτελέσματα στον πίνακα search_caching της βάσης δεδομένων και χωρίς να τροποποιούμε και να επεκτείνουμε καμία από τις υπάρχουσες καταχωρήσεις για το χρήστη αυτό. Ανάμεσα στις επερωτήσεις που υποβλήθηκαν, συμπεριλάβαμε και επερωτήσεις πανομοιότυπες με τις προαποθηκευμένες της πρώτης φάσης έτσι ώστε να μπορούμε να κάνουμε στη συνέχεια της απαραίτητες συγκρίσεις.

Ο μηχανισμός προσωποποιημένης αναζήτησης του συστήματος μας, όπως έχει περιγραφεί σε προηγούμενο κεφάλαιο, λαμβάνει υποψιν του την καθημερινή συμπεριφορά κάθε εγγεγραμένου χρήστη (άρθρα που διαβάζει ή που απορρίπτει, χρόνος που καταναλώνεται σε κάθε άρθρο κλπ) και δυναμικά εξελίσσει το προφίλ του χρήστη ώστε να απεικονίζει τις προτιμήσεις του ανά πάσα χρονική στιγμή περιήγησης στο σύστημά μας. Για παράδειγμα, είναι πιθανό για ένα χρήστη να έχει δηλώσει κατά την εγγραφή του σύστημα ως αγαπημένη θεματική του ενότητα τα αθλητικά αλλά μπορεί κατά περιπτώσεις να παρουσιάζει ένα αυξημένο ενδιαφέρον για άρθρα που ανήκουν θεματικά στη κατηγορία της τεχνολογίας. Το σύστημα προσωποποίσησης τότε εξελίσσει το προφίλ του και ξεκινά να τον τροφοδοτεί με τεχνολογικά νέα ανάμεσα στα αθλητικά που τον ενδιαφέρουν άμεσα και αυτή η εξέλιξη έχει μια προφανή επιρροή τις συνεδρίες του εντός του συστήματος.



Εικόνα 17: υποβάθμιση της ακρίβειας του αποτελέσματος αναζήτησης με την πάροδο των ημερών σε σχέση με μια «φρέσκια» αναζήτηση.

Στην εικόνα 17 μπορούμε να παρατηρήσουμε πως η ακρίβεια του αποτελέσματος της αναζήτησης υποβαθμίζεται με την πάροδο των ημερών όταν συγκρίνουμε τα άρθρα που προκύπτουν από τη μνήμη προαποθήκευσης με τα άρθρα που θα προέκυπταν από μια "φρέσκια" αναζήτηση. Για τον εικονικό μας χρήστη, τη πρώτη ημέρα, η μέση ακρίβεια των αποτελεσμάτων ήταν όπως ήταν και αναμενόμενο στο 100% καθώς είναι η μέρα που γίνεται η αποθήκευση των πραγματικών αποτελεσμάτων στην cache και τα άρθρα που επιστρέφονται είναι αυτά που προκύπτουν από κανονική αναζήτησης στον πίνακα articles και δεν έχουν καθόλου να κάνουν με τη μνήμη προαποθήκευσης. Κάθε επόμενη μέρα παίρνουμε τα αποτελέσματα επερωτήσεων από κανονικές αναζητήσεις (χωρίς να χρησιμοποιήσουμε τη μνήμη και τον αλγόριθμο προαποθήκευσης) που έχουν συντελεστή σχετικότητας άνω του 35% με τις επερωτήσεις που υποβάλλονται. Για τα άρθρα του αποτελέσματος υπολογίζουμε κατά μέσο όρο πόσα από αυτά υπήρχαν στη μνήμη προαποθήκευσης για τη συγκεκριμένη επερώτηση. Καθώς περνάει ο χρόνος, η έξοδος των αναζητήσεων που δεν χρησιμοποιούν τη μνήμη προαποθήκευσης αλλάζει καθώς νέα άρθρα προστίθενται στη βάση δεδομένων και φυσικά το προφίλ του χρήστη εξελίσσεται για να αντικατοπτρίσει τις αλλαγές στις προτιμήσεις του. Το αποτέλεσμα είναι ότι το ποσοστό των άρθρων που υπάρχουν στην μνήμη προαποθήκευσης (από την πρώτη κιόλας ημέρα) μειώνεται και από το σχήμα παρατηρούμε ότι μέχρι τη 10η ημέρα του πειράματος, βλέπουμε ότι η ακρίβεια είναι ακόμα κοντά στο 90%, ποσοστό το οποίο είναι αρκετά ικανοποιητικό αν λάβουμε υπόψιν και το ώφελος που έχουμε σε υπολογιστικούς πόρους και χρόνο που καταναλώνεται από τον εξυπηρετητή.

Ωστόσο, αφού περάσουν δύο εβδομάδες, η ακρίβεια του αποτελέσματος που προκύπτει από τη μνήμη προαποθήκευσης υποβαθμίζεται στο 70% και ώς το τέλος της τρίτης εβδομάδας είναι κοντά στο 55%. Με άλλα λόγια, αν παρουσιάζαμε στο χρήστη σε αυτό το σημείο τα άρθρα που προκύπτουν από τη μνήμη προαποθήκευσης της πρώτης μέρας αντί για τα αποτελέσματα μιας νέας αναζήτησης, ο χρήστης θα έβλεπε λίγα παραπάνω από μισά άρθρα από αυτά που ανταποκρίνονται στο προφίλ και τις προτιμήσεις του, πράγμα το οποίο συνεπάγεται την υποβάθμιση της εξόδου του συστήματος από τη σκοπιά της ακρίβειας και της ποιότητας των αποτελεσμάτων

Ως συμπέρασμα, η προαποθήκευση των αποτελεσμάτων μιας αναζήτησης και η χρήση αυτού του αποτελέσματος για πάνω από δύο εβδομάδες δεν είναι μια ικανοποιητική λύση για ένα εγγεγραμμένο χρήστη καθώς αφήνει πολλές πιθανότητες για υποβάθμιση της ποιότητας του αποτελέσματος και την παραγωγή εσφαλμένης εξόδου που δε συμφωνεί με το εξελισσόμενο προφίλ του χρήστη. Ωστόσο, για μη εγγεγραμμένους χρήστες, για τους οποίους δεν υπάρχει διαμορφωμένο και αποθηκευμένο προφίλ, θα μπορούσε να γίνει χρήση ενός επεκτεταμένου χρονικού διαστήματος για τη λήξη των καταχωρήσεων της μνήμης προαποθήκευσης. Στην υλοποίηση μας, υπάρχει μια διαφοροποίηση για εγγεγραμμένους και μη εγγεγραμμένους χρήστες κατά τη διάρκει του ελέγχου για προαποθηκευμένα αποτελέσματα για μια επερώτηση.

10

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο κεφάλαιο αυτό περιγράφονται τα συμπεράσματα από τη χρήση του μηχανισμού

10. ΣΥΜΠΕΡΑΣΜΑΤΑ & ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Λόγω της εγγενούς δυναμικότητας του Διαδικτύου, το περιεχόμενο των ιστοσελίδων αλλάζει συνεχώς, ειδικά όταν συζητούμε για ένα μηχανισμό που ανακτά παραπάνω από 1500 άρθρα σε καθημερινή βάση και τα παρουσιάζει προσωποποιημένα πίσω στον τελικό χρήστη. Οι προσωποποιημένες διαδικτυακές πύλες δίνουν τη δυνατότητα για εστιασμένα στους χρήστες αποτελέσματα ωστόσο είναι απαραίτητο να δημιουργούνται ακριβή προφίλ για τους χρήστες. Με βάση τα προφίλ που δημιουργεί ο μηχανισμός του PeRSSonal κατασκευάσαμε μια προσωποποιημένη μηχανή αναζήτησης για το υποσύστημα αναζήτησης της διαδικτυακής πύλης του PeRSSonal. Σκοπός ήταν η βελτίωση της διαδικασίας αναζήτησης τόσο για εγγεγραμένους όσο και για μη εγγεγραμένους χρήστες.

Δίνοντας βάρος κυρίως στου εγγεγραμένους χρήστες, εννοώντας ότι κατέχουμε πληροφορία για τις προτιμήσεις τους, παρουσιάσαμε ένα σύστημα το οποίο είναι ικανό να προσωποποιεί τα αποτελέσματα της αναζήτησης στο προφίλ κάθε χρήστη και να διατηρεί μια συνέπει στη διαδικασία αναζήτησης ανεξάρτητα από τις αλλαγές που μπορούν να υποστούν τα προφιλ των χρηστών. Παρουσιάσαμε και αναλύσαμε αλγορίθμους και τύπους που οδηγούν στην προσωποποίηση των αποτελεσμάτων και πιο συγκεκριμένα στην αναβάθμιση της διαδικασίας ταξινόμησης των αποτελεσμάτων έτσι ώστε να ωθήσουμε τα σχετικά άρθρα στην κορυφή των αποτελεσμάτων. Συγκρίνοντας τα αποτελέσματα σε αυτά μιας γενικής και μη προσωποποιημένης αναζήτησης, είναι προφανές ότι το σύστημα είναι σε θέση να βελτιώσει τη διαδικασία αναζήτησης και να διευκολύνει τους χρήστες να εντοπίσουν τα επιθυμητά άρθρα πιο εύκολα και γρήγορα.

Επιπλέον περιγράψαμε έναν αλγόριθμο προαποθήκευσης (caching) που επίσης χρησιμοποιείται στο υποσύστημα αναζήτησης του PeRSSonal. Παρουσιάσαμε τηςδιαδικασία με την οποία τα αποτελέσματα από διάφορες επερωτήσεις των χρηστών αποθηκεύονται με εξειδικευμένο τρόπο στη βάση δεδομένων ώστε να μπορούν να ανακτηθούν γρήγορα και χωρίς αυξημένη κατανάλωση υπολογιστικών πόρων όταν παρόμοιες επερωτήσεις υποβάλλονται. Τέλος περιγράψαμε τεχνικές με τις οποίες μπορούμε να αποδείξουμε την βελτίωση που επιτυγχάνεται στην ταχύτητα της διαδικασίας αναζήτησης του συστήματος μας.

Συγκρίνοντας τα αποτελέσματα μιας μη εξειδικευμένης αναζήτησης με αυτά της δικής μας υλοποίησης, είναι προφανές ότι καταφέραμε να βελτιώσουμε τη διαδικασία αναζήτησης και να διευκολύνουμε τους χρήστες να εντοπίσουν πιο γρήγορα τα επιιθυμητά άρθρα. Το σύστημα, όπως κάθε διαδικτυακό σύστημα, εξάγει τα αποτελέσματα σε μορφή XML έτσι ώστε να εξασφαλίσει τη συμβατότητα με τα γενικά πρότυπα. Για το μέλλον, αυτό που θα μπορούσε να επιτευχθεί είναι η περεταίρω βελτίωση της διαδικασίας αναζήτησης με έναν πιο ακριβή αλγόριθμο προσωποποίησης έτσι ώστε να καταστεί η συνολική διαδικασία γρηγορότερη και ώστε να μην παρουσιάζονται στον τελικό χρήστη άρθρα και αποτελέσματα που έχουν πολύ μικρό ή καθόλου ενδιαφέρον για αυτόν με βάση το προφίλ και τις προτιμήσεις του. Τέλος ένας μηχανισμός ομαδοποίησης των χρηστών θα μπορούσε να προστεθεί στο προσωποποιημένο σύστημα αναζήτησης με σκοπο την πρόταση ορισμένων άρθρων σε ίδιες ή παρόμοιες επερωτήσεις που επιλέχθησαν από χρήστης με παρόμοιο προφίλ και προτιμήσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΒΙΒΛΙΑ / ΔΗΜΟΣΙΕΥΣΕΙΣ

[1] Mooers, C. N. 1952. Information Retrieval Viewed as Temporal Signaling. In Proceedings of the International Conference of Mathematicians, Cambridge, Massachusetts. American Mathematical Society, σελίδες 572-573.

[2] Doyle L. B. 1961. Semantic Road Maps for Literature Searchers. In Journal of the Association for Computing Machinery, 8, σελίδες 553-578.

[3] Salton, G. 1968. Automatic Information Organization and Retrieval. New York: McGraw-Hill.

[4] Shneiderman, B., Byrd, D. and Croft, B. 1998. Sorting out Searching: a User-Interface Framework for Text Searches. In Communications of the ACM, 41(4), σελίδες 95-98.

[5] Salton, G. and Buckley, C. 1988. Improving Retrieval Performance by Relevance Feedback. In Journal of the American Society for Information Science, 41, σελίδες 288-297.

[6] Cleverdon, C. W. 1972. The Cranfield Tests on Index Language Devices. In Aslib Proceedings, 19, σελίδες 173-192.

[7] Belkin, N. J., and Croft, W. B. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? In Communications of the ACM, 35(12), σελίδες 29-38.

[8] Bernard J. Jansen , Amanda Spink , Tefko Saracevic, Real life, real users, and real needs: a study and analysis of user queries on the web, Information Processing and Management: an International Journal, v.36 n.2, p.207-227, Jan.1.2000

[9] Robert Krovetz , W. Bruce Croft, Lexical ambiguity and information retrieval, ACM Transactions on Information Systems (TOIS), v.10 n.2, p.115-141, April 1992

[10] F. Qiu and J. Cho. Automatic identification of user preferences for personalized search. Technical report, UCLA Computer Science Department, 2005.

[11] Alexander Pretschner , Susan Gauch, Ontology Based Personalized Search, Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, p.391, November 08-10, 1999

[12] P. Ferragina , A. Gulli, A personalized search engine based on Web-snippet hierarchical clustering, Software—Practice & Experience, v.38 n.2, p.189-225, February 2008

[13] Georg Buscher , Andreas Dengel , Ludger van Elst, Query expansion using gaze-based feedback on the subdocument level, Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, July 20-24, 2008, Singapore, Singapore

[14] Bouras, C., and Konidaris, A. 2001. Web Components: A Concept for Improving Personalization and Reducing User Perceived Latency on the World Wide Web. The 2nd International Conference on Internet Computing (IC2001),Las Vegas, Nevada, USA, Vol. 2, σελίδες 238-244.

[15] Bouras, C., Kapoulas, V., and Misedakis, I. 2004. Web Page Fragmentation for Personalized Portal Construction. IEEE International Conference on Information Technology: Coding and Computing - ITCC 2004 (Web/IR Track), The Orleans, Las Vegas, Nevada, USA, σελίδες 332 – 336.

[16] Πανεπιστημιακό Σύγγραμμα «Ανάκτηση Πληροφορίας». Δρ. Χρήστος Μακρής, Ε. Θεοδωρίδης, Ι. Παναγής, Α. Περδικούρη, Ε. Χριστοπούλου.

[17] Πανεπιστημιακές Παραδόσεις «Προηγμένα Πληροφοριακά Συστήματα». Α. Τσακαλίδης, Β. Βασιλειάδης, Ε. Σακκόπουλος.

[18] Mandar Rahurkar , Silviu Cucerzan, Predicting when browsing context is relevant to search, Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, July 20-24, 2008, Singapore, Singapore

[19] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. Communications of the ACM, 43(8):142--151, 2000.

[20] T. Joachims, D. Freitag, and T. Mitchell. 1997. Webwatcher: A tour guide for the World Wide Web. In Proc. IJCAI-97. http://citeseer.ist.psu.edu/joachims96webwatcher.html

[21] Dick Hardt. How SXIP Works (whitepaper). https://sxip.org/docs/specs/how-sxip-works.pdf 2004.

[22] Proposal for an Open Profiling Standard. W3C Note – 02 June 1997. http://www.w3.org/TR/NOTE-OPS-FrameWork

[23] PIDL - Personalized Information Description Language. W3C Note - 09 Feb 1999. http://www.w3.org/TR/1999/NOTE-PIDL-19990209

[24] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004. http://www.w3.org/TR/2004/REC-CCPP-structvocab-20040115/

[25] The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation 16 April 2002.

http://www.w3.org/TR/P3P/

[26] Project Liberty. Introduction To The Liberty Alliance Identity Architecture (whitepaper). 2003. Available from https://www.projectliberty.org/resources/whitepapers/LAP%20Identity%20Architecture%20Whitepaper%20Final.pdf

[27] Gary Ellison, Jeff Hodges, Susan Landau (2002) Security and Privacy Concerns of Internet Single Sign-On: Risks and Issues as They Pertain to Liberty AllianceVersion 1.0 Specifications. Technical Report.

[28] Dick Hardt. How SXIP Works (whitepaper). https://sxip.org/docs/specs/how-sxip-works.pdf 2004.

[29] Jude Shavlik et al : An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web. Proceedings of the 1999 International Conference on Intelligent User Interfaces, pp. 157 - 160, Redondo Beach, CA.

[30] Dwi H. Widyantoro, Thomas R.Ioerger and John Yen, Learning User Interest Dynamics with a Three-Descriptor Representation. Journal of the American Society for Information Science, 52(3):212-225.

[31] Philip Chan: Constructing Web User Profiles: A Non-invasive Learning Approach. KDD-99 Workshop on Web Usage Analysis and User Profiling, pp. 7-12, 1999.

[32] Michael Pazzani et al : Syskill & Webert : Identifying interesting Web sites. M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting Web sites," in Proceedings of the 13th National Conference on Artificial Intelligence (AAA196), 1996, pp. 54--61.

[33] Jeremy Goecks, Jude Shavlik: Automatically Labeling Web Pages Based on Normal User Actions. In Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, July 1999.

[34] Dongling Chen , Daling Wang , Ge Yu , Fang Yu, A PLSA-based approach for building user profile and implementing personalized recommendation, Proceedings of the joint 9th Asia-Pacific web and 8th international conference on web-age information management conference on Advances in data and web management, June 16-18, 2007, Huang Shan, China

[35] Jingfang Xu , Chuanliang Chen , Gu Xu , Hang Li , Elbio Renato Torres Abib, Improving quality of training data for learning to rank using click-through data, Proceedings of the third ACM international conference on Web search and data mining, February 04-06, 2010, New York, New York, USA

[36] Dhruba J. Baishya, Enhanced visual experience and archival reusability in personalized search based on modified spider graph, Proceedings of the 3rd international conference on Advances in visual computing, November 26-28, 2007, Lake Tahoe, NV, USA

[37] H. Arimura, A.Wataki, R. Fujino, and S. Arikawa. A fast algorithm for discovering optimal string patterns in large text databases. Proc. the 8th International Workshop on Algorithmic Learning Theory, 1501:247–261.

[38] M. Montes-y G´omez, A. Gelbukh, and A. Lopez-Lopez. Mining the News: Trends, Associations, and Deviations. Computation y Sistemas, 5(1):14–24, 2001.

[39] K. Hoang and P. Do. Discovering Motiv Based Association Rules in a Set of DNA sequences. RSCTC, pages 386–390, 2000.

[40] Colleen E. Crangle. Text summarization in data mining. In Soft-Ware 2002: Proceedings of the First International Conference on Computing in an Imperfect World, pages 332–347, London, UK, 2002. Springer-Verlag.

[41] N. Collier, C. Nobata, and J. Tsujii. Extracting the names of genes and gene products with a hidden Markov model. Proceedings of the 18th conference on Computational linguistics-Volume 1, pages 201–207, 2000.

[42] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. Tuning Support Vector Machines for Biomedical Named Entity Recognition. Proc. of the Workshop on Natural Language Processing in the Biomedical Domain (at ACL'2002), pages 1–8, 2002.

[43] C. Nobata, N. Collier, and J. Tsujii. Automatic term identification and classification in biology texts. Proc. of the 5th NLPRS, pages 369–374, 1999.

[44] K.S. Jones. Exhaustivity and specificity. Journal of Documentation, 28(1):11–21, 1972.

[45] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. Proceedings of the Fourteenth International Conference on Machine Learning, 97, 1997.

[46] D. Mladenic and M. Grobelnik. Word sequences as features in text-learning. Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference, pages 145–148, 1998.

[47] J.C. French, A.L. Powell, J. Callan, C.L. Viles, T. Emmitt, K.J. Prey, and Y. Mou. Comparing the performance of database selection algorithms. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 238–245, 1999.

[48] M. Lennon. Pierce. D., Tarry, B.. & Willett, P.(198 1). An evaluation of the stemming algorithms.

[49] J.B. Lovins. Development of a Stemming Algorithm. 1968.

[50] M. Porter. The Porter Stemming Algorithm. Accessible at http://www. tartarus. org/martin/PorterStemmer.

[51] C.D. Paice. Another stemmer. ACM SIGIR Forum, 24(3):56–61, 1990.

[52] R. Krovetz. Viewing morphology as an inference process. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 191–202, 1993.

[53] W.B. Frakes and R. Baeza-Yates. Information retrieval: data structures and algorithms. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[54] R. Krovetz. Viewing morphology as an inference process. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 191–202, 1993.

[55] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. Proceedings of the seventh international conference on Information and knowledge management, pages 148–155, 1998.

[56] Y. Yang and C.G. Chute. An example-based mapping method for text categorization and retrieval. ACM Transactions on Information Systems (TOIS), 12(3):252–277, 1994.

[57] B. Masand. Lino, G., &Waltz, D.(1992). Classifying news stories using memory based reasoning. Proceedings of 506 15th ACM SIGIR international conference on research and development in information retrieval, pages 59–65.

[58] Y. Yang. Expert network: effective and efficient learning from human decisions in text categorization and retrieval. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 13–22, 1994.

[59] K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 22–35, 1993.

[60] D.D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. Third Annual Symposium on Document Analysis and Information Retrieval, pages 81–93, 1994.

[61] C. Apt´e, F. Damerau, and S.M. Weiss. Towards language independent automated learning of text categorization models. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[62] W.W. Cohen. Text categorization and relational learning. Proceedings of ICML-95, 12th International Conference on Machine Learning, pages 124–132, 1995.

[63] H.T. Ng, W.B. Goh, and K.L. Low. Feature selection, perception learning, and a usability case study for text categorization. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pages 67–73, 1997.

[64] T. Joachims. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. Springer-Verlag London, UK, 1998.

[65] PC Reghu Raj and S. Raman. Content identification and semantic indexing of text documents. Proc. Of the Indo European Conference on Multilingual Communication Technologies (IEMCT-02), pages 203–217, 2002.

[66] M.F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. Text Databases and Document Management: Theory and Practice, pages 78–102, 2001.

[67] J. Furnkranz, T. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. Learning for Text Categorization: Proceedings of the 1998 AAAI/ICML Workshop, pages 98–05, 1998.

[68] E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pages 117–124, 1997.

[69] C. Jacquemin. Spotting and Discovering Terms Through Natural Language Processing. MIT Press, 2001.

[70] H. Berger and D. Merkl. A Comparison of Text-Categorization Methods applied to N-Gram Frequency Statistics. Proc. of the 17th Australian Joint Conf. on Artificial Intelligence, 2004.

[71] B. Sankaran. Tamil Search Engine.

[72] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence, 139(1):91–107, 2002.

[73] R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. Proceedings of HLT-NAACL 2004, pages 113–120, 2004.

[74] G. Salton, J. Allan, C. Buckley, and A. Singhal. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. Science, 264(5164):1421, 1994.

[75] M. Saravanan, Pc Reghu Raj, and S. Raman. Summarization and Categorization of text data in high-level data cleaning for information retrieval. Applied Artificial Intelligence, 17(5):461–474, 2003.

[76] M. Saravanan and S. Raman. The term distribution model for summarization of multiple documents. Proceedings of the Indo European Conference on Multilingual Communication Technologies (IEMCT 2002), pages 182–192, 2002.

[77] R. Evans, R. Gaizauskas, L. Cahill, J.Walker, J. Richardson, and A. Dixon. POETIC: a system for gathering and disseminating traffic information. Journal of Natural Language Engineering, 1(4), 1995.

[78] D. Marcu. The rhetorical parsing of natural language texts. Proceedings of the 35th annual meeting on Association for Computational Linguistics, pages 96–103, 1997.

[79] R.C. Schank. Reading and Understanding: Teaching from the Perspective of Artificial Intelligence. Lawrence Erlbaum Associates, 1982.

[80] WA Woods and JG Schmolze. The KL-ONE family. Semantic Networks in Artificial Intelligence, Pp133-178, 1992.

[81] D. Fum, G. Guida, and C. Tasso. Forward and backward reasoning in automatic abstracting. Proceedings of the 9th conference on Computational linguistics-Volume 1, pages 83–88, 1982.

[82] PS Jacobs and L.F. Rau. SCISOR: extracting information from on-line news. Communications of the ACM, 33(11):88–97, 1990.

[83] U. Hahn and U. Reimer. Semantic Parsing and Summarizing of Technical Texts in the TOPIC System. Informations linguistik, pages 153–193, 1986.

[84] L.A. Mather and J. Note. Discovering Encyclopedic Structure and Topics in Text. Sixth ACM SIGKDD.

[85] I. Mani and G. Wilson. Robust temporal processing of news. Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pages 69–76, 2000.

[86] J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection: 1999 summer workshop at CLSP, final report, 1999.

[87] R. Swan and J. Allan. Automatic generation of overview timelines. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 49–56, 2000.

[88] PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries. Data and Knowledge Engineering Journal, Elsevier Science, 2007, C. Bouras, V. Poulopoulos, V. Tsogkas, 2007

[89] Efficient Summarization Based On Categorized Keywords. The 2007 International Conference on Data Mining (DMIN07), Las Vegas, Nevada, USA, C. Bouras, V. Poulopoulos, V. Tsogkas, 25 - 28 June 2007

[90] Personalizing text summarization based on sentence weighting. IADIS European First International Conference Data Mining (ECDM 2007), Lisbon, Portugal, C. Bouras, V. Poulopoulos, V. Tsogkas, 3 - 8 July 2007

[91] The importance of the difference in text types to keyword extraction: Evaluating a mechanism. 7th International Conference on Internet Computing 2006 (ICOMP 2006), Las Vegas, Nevada, USA, C. Bouras, C. Dimitriou, V. Poulopoulos, V. Tsogkas, 26 - 29 June 2006, pp. 43 - 49

[92] Scalability of text classification. 2nd International Conference on Web Information Systems and Technologies (WEBIST 2006), Setubal, Portugal, I. Antonellis, C. Bouras, V. Poulopoulos, A. Zouzias, 19 - 22 April 2006, pp. 408 - 413

[93] Personalized News Categorization through Scalable Text Classification. The Eight Asia Pacific Web Conference (APWeb – 06), Harbin, China, I. Antonellis, C. Bouras, V. Poulopoulos, 16 - 18 January 2006, pp. 391 - 401

[94] E. Casaola. Pro Fusion Personal Assistant: An agent for personalized information filtering on the WWW. Master's Thesis, The University of Kansas, Lawrence, KS, 1998.

[95] B.J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: A study and analysis of user queries on the Web. Information Processing and Management, , 2000, 36(2): pp. 207 – 227

[96] Robert Krovetz and W. Bruce Croft. Lexical ambiguity and information retrieval. Information Systems, 1992, 10(2): pp. 115-141

[97] S. Lawrence, Context in Web Search. IEEE Data Engineering Bulletin. 2000, 23: pp. 25 – 32

[98] J. Xu and W.B. Croft. Query Expansion using local and global document analysis. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Zurich, Switzerland, 1996, pp. 4-11

[99] Glover, E., Lawrence, s., Brimingham, W.,Andgiles, c. L. Architecture of a meta search engine that supports user information needs. In Proceedings of the 8th International Conference on Information Knowledge Management. Kansas City, MO, 1999, pp. 210–216.

[100] Leory, G., Lally, a. M.,Andchen. The use of dynamic contexts to improve casual internetsearching.ACM Trans. Inform. Syst. 2003, 21(3): pp. 229–253.

[101] Oyama, S., Kokubo, T.,Andishida, T. Domain-specific Web search with keyword spices..IEEE Transformation Knowledge and Data Engineering. 2004, 16(1): pp. 17–27.

[102] Teevan, J., Dumais, S. T.,Andhorvitz, E. Personalizing search via automated analysis of interests and activities. In Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Salvador, Brazil, 2005, pp. 449–456.

[103] Shen, X., Tan, B.,Andzhai, C. X. Context-sensitive information retrieval using implicit feedback. In Proceedings of the 28th Annual International ACM

SIGIR Conference on Research and Development in Information Retrieval. Salvador, Brazil, 2005, pp. 43–50.

[104] Shen, X., Tan, B.,Andzhai, C. X. Implicit user modeling for personalized search. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management. Bremen, Germany, 2005, pp. 824–831.

[105] Markatos, E.P., 2001. On caching search engine query results. Computer Communications 24, pp. 137–143.

[106] Xie, Y., O'Hallaron, D.R., 2002. Locality in search engine queries and its implications for caching. IEEE Infocom 2002, pp. 1238 – 1247.

[107] Lempel, R., Moran, S., 2003. Predictive caching and prefetching of query results in search engines. Proceedings of the 12th WWW Conference, pp. 19–28

[108] Fagni, T., Perego, R., Silvestri, F., Orlando, S., 2006. Boosting the performance of Web search engines: Caching and prefetching query results by exploiting historical usage data. ACM Transactions on Information Systems 24, pp. 51–78.

[109] Jansen B. J., Spink A., 2006. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing and Management, 42(1) ,pp. 248-263.

[110] Teevan J., Adar E., Jones R. and Pott M., 2006. History repeats itself: Re-peat queries in Yahoo's logs. ACM SIGIR, pp. 703-704.

Δικτυακοι Τοποι

[111] Μηχανή αναζήτησης Google. http://www.google.com

[112] Μηχανή αναζήτησης Altavista. http://www.altavista.com

[113] Portal Yahoo! http://www.yahoo.com

[114] Διεθνές ειδησεογραφικό πρακτορείο CNN. http://www.cnn.com

[115] Διεθνές ειδησεογραφικό πρακτορείο BBC. http://www.bbc.co.uk

[116] Διεθνές ειδησεογραφικό πρακτορείο Reuters. http://www.reuters.com

[117] Διεθνές ειδησεογραφικό πρακτορείο FoxNews http://www.foxnews.com

[118] MySQL, Βάση Δεδομένων ανοιχτού κώδικα. http://www.mysql.com

[119] PostgreSQL, Βάση Δεδομένων ανοιχτού κώδικα. http://www.postrgresql.org

[120] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα C++ (C Plus Plus). http://en.wikipedia.org/wiki/C_Plus_Plus

[121] Το χρονικό της Java. http://ils.unc.edu/blaze/java/javahist.html.

[122] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα Perl. http://en.wikipedia.org/wiki/Perl

[123] Επίσημος Δικτυακός τόπος της PHP. http://www.php.net/

[124] Ελεύθερη εγκυκλοπαίδεια WikiPedia. Θέμα JSP. http://en.wikipedia.org/wiki/JavaServer_Pages

[125] http://www.amazon.com

[126] http://www.passport.net

[127] http://www.gartner.com