ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ

ΚΑΤΕΥΘΥΝΣΗ: ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗ ΑΝΑΚΑΛΥΨΗ ΟΜΑΔΑΣ ΕΙΔΙΚΩΝ ΓΙΑ ΤΗΝ ΚΑΛΥΨΗ ΣΥΝΟΛΟΥ ΤΟΜΕΩΝ ΕΝΔΙΑΦΕΡΟΝΤΟΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΦΡΑΓΚΟΣ ΝΙΚΟΛΑΟΣ

Επιβλέπων: Χρήστος Δουλκερίδης

Πειραιάς 2015

Πρόλογος

Πολύ πριν την έλευση των υπολογιστών υπήρχε η ανάγκη για εύρεση εξειδικευμένων ατόμων. Πλέον ο τεράστιος όγκος δεδομένων συνάμα με την ανάγκη για αξιοποίηση της πληροφορίας αυτής κάνουν ιδιαίτερα χρήσιμη την δυνατότητα αυτοματοποιημένης αναζήτησης ειδικών. Η συγκρότηση επιτροπών για αξιολόγηση δημοσιεύσεων αλλά και η επιλογή κατάλληλων υπαλλήλων για πρόσληψη αποτελούν μόνο δύο από τα πολλά παραδείγματα.

Στα πλαίσια της παρούσας διπλωματικής υλοποιήθηκε μία μέθοδος για την επίλυση του προβλήματος της εύρεσης ειδικών. Η είσοδος αποτελείται από τομείς εξειδίκευσης q και έναν αριθμό που προσδιορίζει το επιθυμητό πλήθος των αποτελεσμάτων έστω K. Κάθε ειδικός συνοδεύεται από ένα σύνολο πληροφορίας το οποίο προσδιορίζει την έρευνα και τα ενδιαφέροντα του έστω d. Συνδυάζοντας την είσοδο με το εκάστοτε σύνολο d προκύπτει ένα προφίλ για κάθε ειδικό. Το προφίλ αποτελεί ένα ποσοτικό μέτρο σύγκρισης για το πόσο σχετικός είναι κάποιος και σε ποιους τομείς. Από εκεί και πέρα το πρόβλημα μοντελοποιείται ως μέγιστη κάλυψη συνόλου (Maximum Set Cover) και χρησιμοποιείται η προσέγγιση με έναν άπληστο αλγόριθμο. Στην συνέχεια, με την εκτέλεση του εν λόγω αλγορίθμου επιστρέφονται οι καλύτεροι K ειδικοί που καλύπτουν όσο το δυνατό περισσότερους, από τους τομείς q της εισόδου. Οι πειραματικές μετρήσεις αποδεικνύουν την εγκυρότητα των αποτελεσμάτων με βάση πραγματικούς ακαδημαϊκούς από συνέδρια αλλά και την υπεροχή της μεθόδου σε σχέση με την απλοϊκότερη λύση του αλγόριθμου Cosine Similarity.

Περιεχόμενα

Κατάλογος σχημάτων

1 Εισαγωγή

Πολλές φορές χρειάζεται ένας ή περισσότεροι ειδικοί για να λυθεί ένα πρόβλημα. Παρόλη την πληροφορία που υπάρχει διαθέσιμη πλέον στον παγκόσμιο ιστό, για οτιδήποτε, τίποτα δε μπορεί να αντικαταστήσει την εμπειρία ενός ανθρώπου πάνω σε ένα ζήτημα. Η γνώση που απαιτείται δεν είναι πάντα ελεύθερα προσβάσιμη ή είναι δύσκολο να εκφραστεί στο χαρτί έτσι ώστε να είναι κατανοητή και άμεσα εφαρμόσιμη. Οι ειδικοί μπορεί έχουν ζήτηση όχι μόνο για να δώσουν απαντήσεις, αλλά και για να αναλάβουν οργανωτικούς και καθοδηγητικούς ρόλους.

Στην παρούσα εργασία μελετάται η αυτοματοποιημένη εύρεση ενός συνόλου ατόμων των οποίων η συνδυασμένη εξειδίκευση καλύπτει πλήρως ένα δεύτερο σύνολο από λέξεις κλειδιά.

1.1 Ορισμός του προβλήματος

Σε μεγάλους οργανισμούς ή ακόμα και στην έκταση του παγκόσμιου ιστού πολλοί είναι εκείνοι που αναζητούν πρόσωπα και δη εξειδικευμένα, αντί για κείμενα. Σε πολλές περιπτώσεις μοναδική λύση αποτελεί η γνώμη άλλων ανθρώπων. Μια κλασσική μηχανή αναζήτησης μπορεί να βοηθήσει αρκετά στην αναζήτησης προσώπων αλλά δεν έχει την δυνατότητα να αυτοματοποιήσει την διαδικασία. Η συνηθισμένη αντιμετώπιση του προβλήματος είναι η κατασκευή ενός συστήματος στα πλαίσια μιας εταιρίας για να βοηθήσει στην αναζήτηση ατόμων ή τμημάτων που κατέχουν συγκεκριμένες γνώσεις και δεξιότητες. Έτσι θα εξοικονομηθεί χρόνος και χρήμα για την πρόσληψη ενός συμβούλου/υπαλλήλου όταν το υπάρχον προσωπικό δεν είναι αρκετό. Αντίστοιχα με μια μηχανή αναζήτησης το αυτόματο σύστημα ειδικών λαμβάνει ως είσοδο κάποιες λέξεις κλειδιά και επιστρέφει μια λίστα με άτομα ταξινομημένα με το επίπεδο γνώσεων τους σε σχέση με το θέμα της αναζήτησης [1].

Παρόλα αυτά η εύρεση ενός και μόνο ατόμου είναι ένα θέμα εκτενώς μελετημένο στην βιβλιογραφία. Πληθώρα δημοσιεύσεων αναφέρεται σε τεχνικές σύμφωνα με τις οποίες μπορεί να εξαχθεί γνώση από μια πηγή, συνήθως κείμενο, και να βγει συμπέρασμα για τις γνώσεις ενός ατόμου. Μεγαλύτερη πρόκληση αποτελεί το ερώτημα εάν μπορεί να βρεθεί μια ομάδα ειδικών που να καλύπτει ορισμένους τομείς ενδιαφέροντος. Το ερώτημα αυτό συχνά καλούνται να απαντήσουν οργανωτές συνεδρίων, αρχισυντάκτες περιοδικών ή διαχειριστές επιχορηγήσεων [2]. Η εργασία τους περιλαμβάνει την εύρεση συγκεκριμένου αριθμού ατόμων στους οποίους θα μοιράσουν τις δημοσιεύσεις προς αξιολόγηση. Τα άτομα αυτά θα πρέπει να γνωρίζουν πολύ καλά το θέμα το όποιο πραγματεύεται το κείμενο ή τα κείμενα που τους δόθηκαν. Τα βασικά κριτήρια του διαμοιρασμού είναι δύο. Το πρώτο είναι το περιεχόμενο της δημοσίευσής και το δεύτερο οι γνώσεις ενός αξιολογητή. Καθώς η προαναφερθείσα διαδικασία είναι χειροκίνητη και άρα στηρίζεται στην ανθρώπινη εμπειρία, δεν μπορεί αντικειμενικά μια επιτροπή συνέδριού να γνωρίζει επακριβώς το εύρος και το επίπεδο γνώσεων όλων των συμμετεχόντων. Άρα το έργο αυτό όχι μόνο δεν είναι εύκολο αλλά προσθέτει και τον παράγοντα τις δίκαιης κατανομής. Δηλαδή το κατά πόσο είναι εφικτό σε κάθε αξιολογητή να ανατεθούν κείμενα σχετικά με το αντικείμενό του. Κάποια συστήματα συνεδρίων δίνουν την δυνατότητα στους ειδικούς να εκδηλώσουν ενδιαφέρον για συγκεκριμένες δημοσιεύσεις με το σύστημα της δημοπρασίας. Ακόμα και με αυτό όμως

παραμένει μια χρονοβόρα χειροκίνητη διαδικασία. Ένα άλλο μειονέκτημα είναι ότι η χειροκίνητη ανάθεση είναι μεροληπτική. Για παράδειγμα στις δημοπρασίες μια δημοφιλής δημοσίευση θα δεχτεί περισσότερες προσφορές από μια άλλη. Επίσης το ταίριασμα της εξειδίκευσης θα είναι ακριβέστερο για έναν αξιολογητή που η εξειδίκευσή του είναι ευρέως γνωστή στον συντάκτη ή επιτροπή του συνεδρίου και λιγότερο ακριβής για κάποιον δεν είναι τόσο οικείος. Για τους παραπάνω λόγους είναι επιθυμητό να αναπτυχθούν τεχνικές για την αυτοματοποίηση της διαδικασίας αυτής.

Κοιτώντας αφαιρετικά το πρόβλημα τα κείμενα μπορούν να περιγραφούν ως οντότητες που αναπαριστώνται από λέξεις κλειδιά που αντιστοιχούν σε επιστημονικούς τομείς. Στην συνέχεια ζητείται να βρεθεί ένα σύνολο ειδικών των οποίων η συνδυασμένη γνώση να καλύπτει πλήρως τις λέξεις αυτές. Έχοντας το σύνολο των ειδικών ο εκάστοτε ενδιαφερόμενος μπορεί να τους κατανείμει ανάλογα με τους σκοπούς του. Η αυτοματοποίηση συνεπάγεται την κατασκευή ενός συστήματος με καθορισμένη είσοδο (πληροφορίες, κείμενο), έξοδο (λίστα ειδικών) και διαδικασία επεξεργασίας των δεδομένων (αλγόριθμοι). Επιπλέον παράμετρος, θεωρείται ο στόχος βελτιστοποίησης που στην προκειμένη περίπτωση είναι να μεγιστοποιηθεί η κάλυψη του συνόλου των ειδικοτήτων. Είναι δυνατό να βρεθεί ένα τέτοιο σύστημα; Και εάν ναι είναι αποδοτικό; Η μελέτη για την ύπαρξη και την αξιολόγηση ενός τέτοιου συστήματος αποτελεί το αντικείμενο έρευνας της παρούσας εργασίας.

1.2 Διάρθρωση Εργασίας

Η εργασία χωρίζεται σε οκτώ (8) κεφάλαια συμπεριλαμβανομένης και της εισαγωγής. Τα πρώτα κεφάλαια παρουσιάζουν το θεωρητικό κομμάτι και τα υπόλοιπα αφορούν την εφαρμογή που υλοποιήθηκε καθώς και τις πειραματικές μετρήσεις.

Αρχικά στο κεφάλαιο 2 γίνεται παρουσίαση της βασικής ορολογίας μαζί με την ανασκόπηση της υπάρχουσας βιβλιογραφίας. Στην συνέχεια στο κεφάλαιο 3 παρουσιάζονται οι ορισμοί και τα μαθηματικά μοντέλα που συνθέτουν το πρόβλημα της αυτοματοποιημένης εύρεσης ειδικών. Περαιτέρω ανάλυση της μεθόδου γίνεται στο κεφάλαιο 4 όπου υπάρχει λεπτομερής περιγραφή για κάθε στάδιο επίλυσης. Η αρχιτεκτονική της εφαρμογής που αναπτύχθηκε για να δοκιμαστεί η μέθοδος επίλυσης παρουσιάζεται στο κεφάλαιο 5. Χρησιμοποιώντας την μέθοδο μαλακών συστημάτων [3] γίνεται η ανάλυση των απαιτήσεων ενώ για τον λογικό σχεδιασμό ακολουθήθηκε ένα υποσύνολο από τα διαγράμματα της UML [4].

Ακολουθεί η επεξήγηση της γραφικής διεπαφής στο κεφάλαιο 6 καθώς και οι λεπτομέρειες που αφορούν την υλοποίηση της εφαρμογής. Στο επόμενο κεφάλαιο παρατίθενται τα αποτελέσματα των πειραματικών μετρήσεων με αναλυτικούς αλλά και συγκεντρωτικούς πίνακες και διαγράμματα. Τα συμπεράσματα τα οποία προέκυψαν από τις προαναφερθείσες μετρήσεις παρουσιάζονται στο κεφάλαιο 8 μαζί τις προτεινόμενες μελλοντικές επεκτάσεις.

Επιπλέον προσθήκη αποτελούν τα τρία παραρτήματα. Στο πρώτο παράρτημα παρατίθενται όλες οι λέξεις κλειδιά που χρησιμοποιήθηκαν για τις πειραματικές μετρήσεις. Στο παράρτημα Β περιγράφεται η διαδικασία προγραμματισμού ενός νέου αλγορίθμου για την εφαρμογή, ενώ στο παράστημα Γ αναφέρονται τα εργαλεία λογισμικού που χρησιμοποιήθηκαν.

2 Ανασκόπηση Βιβλιογραφίας

2.1 Εισαγωγή

Στο παρόν κεφάλαιο, αναλύεται η βιβλιογραφία που έχει συγκεντρωθεί σχετικά με το πρόβλημα της εύρεσης ειδικών. Αρχικά, γίνεται μια πολύ σύντομη εισαγωγική αναφορά σε γενικούς όρους και στη συνέχεια παρουσιάζονται άλλες δημοσιεύσεις που προέκυψαν από τη βιβλιογραφική έρευνα.

2.2 Ορολογία

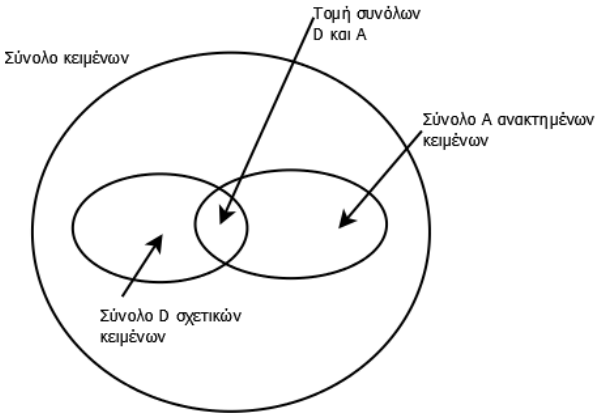
2.2.1 Ανάκτηση δεδομένων

Η έννοια της ανάκτησης δεδομένων (για συντομία IR) είναι αρκετά ευρεία. Ακόμα και το να βγάλει κάποιος μία κάρτα από την τσέπη του για να πληκτρολογήσει τον αριθμό της είναι μια μορφή ανάκτησης δεδομένων. Ωστόσο από την καθαρά ακαδημαϊκή σκοπιά θα μπορούσε να οριστεί ως η αναζήτηση υλικού (συνήθως εγγράφων) αδόμητης φύσης (συνήθως κειμένου) η οποία ικανοποιεί μια ανάγκη πληροφόρησης από μία μεγάλη συλλογή δεδομένων (συνήθως εγκατεστημένων σε Η/Υ). Ο όρος «αδόμητα δεδομένα» που χρησιμοποιήθηκε στον ορισμό αναφέρεται στα δεδομένα που δεν έχουν ξεκάθαρη, σημασιολογικά εμφανή δομή για ένα υπολογιστή. Είναι το ακριβώς αντίθετο των δομημένων δεδομένων όπως μία σχεσιακή βάση. Κάποτε πολύ λίγοι άνθρωποι ασχολούνταν με την διαδικασία της ανάκτησης πληροφορίας όπως βιβλιοθηκάριοι ή επαγγελματίες ερευνητές. Ο κόσμος πλέον όμως έχει αλλάξει και εκατομμύρια άνθρωποι εμπλέκονται στην ανάκτηση δεδομένων χρησιμοποιώντας μηχανές αναζήτησης ή ελέγχοντας το ηλεκτρονικό τους ταχυδρομείο [5].

2.2.2 Μέτρα σύγκρισης

Η αξιολόγηση των συστημάτων ανάκτησης πληροφορίας γίνεται μετρώντας πόσο καλά το σύστημα ικανοποιεί τις ανάγκες πληροφόρησης των χρηστών του. Χωρίς επαρκή αξιολόγηση δεν μπορεί να διαπιστωθεί η καταλληλόλητα του συστήματος για μία εφαρμογή και δεν μπορεί να γίνει αντικειμενική σύγκριση με άλλα συστήματα. Η συνηθέστερη πρακτική είναι η αντιπαραβολή των αποτελεσμάτων του συστήματος με ένα σύνολο αποτελεσμάτων το οποίο επιλέχθηκε χειροκίνητα και θεωρείται σωστό [6]. Δύο βασικά μέτρα σύγκρισης αποτελούν τα «Precision», «Recall».

Έστω *D* ένα σύνολο εγγράφων το οποίο απαντάται από το ερώτημα *E* το οποίο θέτει ο χρήστης. Το σύνολο *D* θεωρείται ως το σωστό σύνολο και έχει βρεθεί με τρόπο εξωτερικό προς το σύστημα. Έστω *A* το σύνολο των εγγράφων που επιστρέφει το σύστημα για το ίδιο ερώτημα Ε.



Εικόνα 1 Precision - Recall

Ως Recall ορίζεται η τομή των συνόλων *D* και *A* διαιρεμένη με το πλήθος των σχετικών εγγράφων του *D*.

$$\text{Recall} = \frac{|D \cap A|}{|D|}$$

Ως Precision ορίζεται η ένωση των συνόλων D και A διαιρεμένη με το πλήθος των ανακτημένων εγγράφων του *A*

$$\text{Precision} = \frac{|D \cup A|}{|A|}$$

Επιπλέον μπορεί να γίνει ένας συνδυασμός των παραπάνω μετρικών παίρνοντας τον αρμονικό μέσο τους, ο οποίος ονομάζεται F – measure

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Στα πλαίσια της παρούσας εργασίας δημιουργήθηκε ένα επιπλέον μέτρο σύγκρισης το οποίο ονομάζεται Goodness. Το Goodness ορίζεται για κάθε ερευνητή ως το πλήθος των δημοσιεύσεών του μέσα από ένα σύνολο επιλεγμένων συνεδρίων. Για παράδειγμα έστω το σύνολο συνεδρίων $C = \{c_1, c_2 c_3\}$ και το πλήθος των δημοσιεύσεων $W_1 = \{w_1, w_2 w_3\}$ για τον ερευνητή r_1. Το Goodness υπολογίζεται ως το άθροισμα των $w_1, w_2 w_3$

$$\text{Goodness} = w_1 + w_2 + w_3$$

2.2.3 Αλγόριθμος Cosine Similarity

Τα κείμενα μπορούν να αναπαρασταθούν ως n - διάστατα διανύσματα όπου κάθε όρος ανήκει σε μία διάσταση. Η συσχέτιση μεταξύ δύο κειμένων μπορεί με αυτόν τον τρόπο να μετατραπεί σε σύγκριση δύο διανυσμάτων. Αυτό ποσοτικοποιείται ως το συνημίτονο της γωνίας ανάμεσα στα δύο διανύσματα η οποία είναι η λεγόμενη ομοιότητα συνημίτονου.

Έστω δύο κείμενα $\overrightarrow{t_a}$ και $\overrightarrow{t_b}$. Η ομοιότητα συνημητόνου ορίζεται ως

$$SIM_C(\overrightarrow{t_a}, \overrightarrow{t_b}) = \cos(\theta) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|}$$

Όπου τα $\overrightarrow{t_a}$,$\overrightarrow{t_b}$ είναι n - διάστατα διανύσματα πάνω στο σύνολο όρων T = $\{t_1, \dots, t_m\}$ και ϑ η γωνία μεταξύ τους. Κάθε διάσταση αναπαριστά έναν όρο μαζί με ένα μη αρνητικό βάρος σχετικά με το κείμενο. Ως αποτέλεσμα η ομοιότητα συνημίτονου είναι πάντα μη αρνητική και βρίσκεται στο κλειστό διάστημα [0,1]. Η τιμή 0 σημαίνει ότι τα κείμενα δεν μοιράζονται κανένα στοιχείο, ενώ το 1 σημαίνει ότι είναι ταυτόσημα [7].

Η ομοιότητα συνημίτονου αποτελεί μία από τις πιο δημοφιλείς μεθόδους σύγκρισης που εφαρμόζεται σε συστήματα ανάκτησης δεδομένων.

2.3 TREC & γενικές προσεγγίσεις

Στο συνέδριο TREC [8] το οποίο πραγματεύεται θέματα ανάκτησης δεδομένων παρουσιάζονται συνεχώς καινούργιες προτάσεις και προσεγγίσεις. Οι μέθοδοι μπορούν να διακριθούν σε δύο μεγάλες κατηγορίες οι οποίες είναι οι βασισμένες σε προφίλ [9,10,11,12] και οι βασισμένες σε έγγραφα [13]. Στις μεθόδους με προφίλ πρώτα χτίζεται το προφίλ του κάθε συγγραφέα και στην συνέχεια με βάση την βαθμολογία του κατατάσσεται και ανακτάται χρησιμοποιώντας κλασσικά μοντέλα ανάκτησης δεδομένων. Στις μεθόδους βασισμένες σε έγγραφα αντί για την δημιουργία προφίλ από τους όρους αναζήτησης χρησιμοποιούνται συνοδευτικά έγγραφα ως γέφυρα και οι συγγραφείς βαθμολογούνται με βάση την συχνότητα εμφάνισης των όρων και των αναφορών των συγγραφέων στα συνοδευτικά έγγραφα.

Μια προσπάθεια για γενικευμένο πιθανοτικό μοντέλο/πλαίσιο για την μελέτη της εύρεσης ειδικών αποτελεί η προσπάθεια των Hui Fang και Cheng Xiang Zhai [14]. Από τη μελέτη τους απορρέουν δύο μεγάλες οικογένειες μοντέλων, αυτά που έχουν ως βάση τον υποψήφιο (π.χ. συγγραφέας, υποψήφιος) και αυτά που έχουν ως βάση, τους όρους αναζήτησης (θέμα, περιοχή ενδιαφέροντος), τα οποία είναι ανάλογα των μοντέλων J. Lafferty and C. Zhai [15]. Πιο συγκεκριμένα βαθμολογούν τους υποψήφιους ειδικούς με την πιθανότητα οι υποψήφιοι να είναι σχετικοί για ένα θέμα. Η πρόκληση που αντιμετωπίζουν είναι να υπολογίσουν την πιθανότητα αυτή. Χρησιμοποιούν ένα σύνολο κειμένων d και οι όροι αναζήτησης που μπαίνουν ως είσοδος αποτελούν την περιγραφή ενός θέματος. Δηλαδή η είσοδος

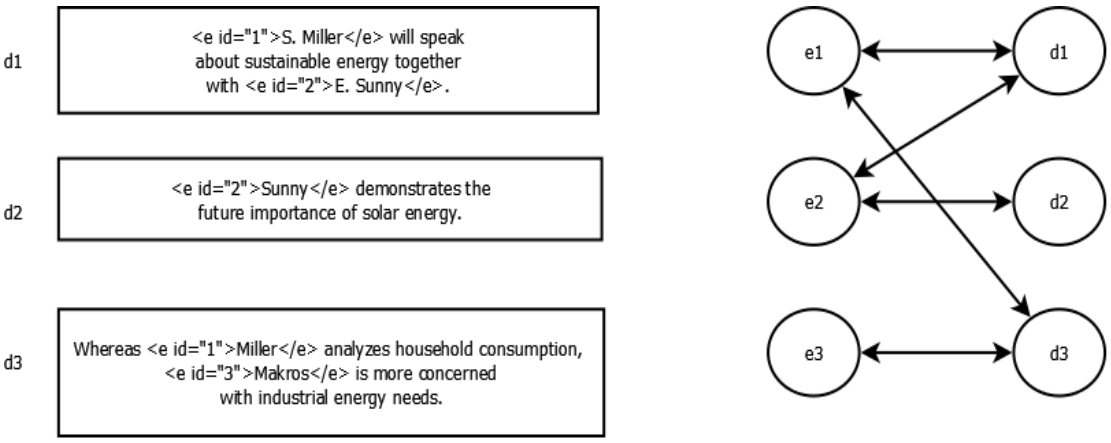
$$t = t_1, t_2 \dots, t_n$$

είναι η περιγραφή ενός θέματος όπου t_i ένας όρος στην περιγραφή αυτή σε αντίθεση με την προσέγγιση που ακολουθείται στην παρούσα εργασία όπου η είσοδος αποτελείται από πολλά θέματα αναζήτησης χωρίς την συνοδεία περιγραφής.

2.4 Καθορισμός εξειδίκευσης

Από του βασικότερους τομείς έρευνας για την ανάκτηση ειδικών είναι, το πώς μετράται η εξειδίκευση. Είναι συνήθης πρακτική να θεωρείται ότι όσο πιο συχνά ένα άτομο σχετίζεται με ένα κείμενο που περιέχει πολλές λέξεις που περιγράφουν ένα θέμα, τόσο πιο πιθανό είναι να θεωρήσουμε ότι αυτός είναι ειδικός. Η απόδειξη για την συσχέτιση μεταξύ ατόμου και εγγράφου θα μπορούσε να γίνει μέσω της αναφοράς προσωπικών χαρακτηριστικών όπως όνομα , email [16] μέσα στο κείμενο του εγγράφου. Με αυτό τον τρόπο οι περισσότερες προσεγγίσεις υπολογίζουν την εξειδίκευση κάποιου αθροίζοντας την βαθμολογία συσχέτισης από όλα τα κείμενα που σχετίζονται με αυτόν. Άλλες μέθοδοι χρησιμοποιούν πιθανοτικά μοντέλα (Probabilistic Models) ενώ οι πιο εξεζητημένες εκμεταλλεύονται και την σημασιολογία των λέξεων (Semantics) [17,18,19].

Πέρα όμως από τα «άμεσα δεδομένα» υπάρχουν και άλλες σημαντικές πληροφορίες που μπορούν να εξαχθούν με έμμεσο τρόπο. Δεν λαμβάνονται πιθανές συσχετίσεις μεταξύ των ειδικών όπως επίσης και κείμενα το οποία σχετίζονται εμμέσως με τα άτομα αυτά. Μία πρόταση για την λύση του παραπάνω προβλήματος είναι η αναπαράσταση των συσχετίσεων μεταξύ ειδικών και κειμένων χρησιμοποιώντας γράφους. Έστω ένα σύνολο κειμένων αντιστοιχισμένο με βαθμολογία ως αποτέλεσμα της κλασσικής ανάκτησης κειμένων για ένα δεδομένο θέμα. Από τα καταταγμένα κείμενα εξάγεται ένα δεύτερο σύνολο περιορισμένων υποψήφιων ειδικών. Εδώ οι σχέσεις μπορούν να αναπαρασταθούν σε έναν γράφο όπου ειδικοί και κείμενα γίνονται οι κορυφές και κατευθυνόμενες ακμές συμβολίζουν τις μεταξύ τους σχέσεις.



Εικόνα 2 Γράφος που αναπαριστά τις σχέσεις μεταξύ ειδικών και κειμένων

Η απλούστερη μορφή των γράφων είναι πάντα διμερής καθώς όλες οι ακμές δείχνουν μόνο από κείμενα προς ειδικούς και πίσω. Έχοντας πλέον έτοιμη όλη την πληροφορία σε μορφή γράφων θα πρέπει να εφαρμοστεί μια επαναληπτική διαδικασία. Η διαδικασία αυτή ονομάζεται περίπατος γράφου και στόχος είναι να επικεντρωθεί ο περίπατος αυτός γύρω από τα πιο σχετικά έγραφα [1].

Ως βοήθημα για τις παραπάνω μεθόδους μπορούν να χρησιμοποιηθούν τα λεγόμενα μη τοπικά στοιχεία για την εύρεση εξειδίκευσης. Για παράδειγμα ένας υποψήφιος ειδικός *r* σε σχέση με ένα κείμενο *d* δεν έχει την ίδια σημαντικότητα εάν απλά αναφέρεται το όνομά του μέσα στο κείμενο ή είναι ο συγγραφέας του κειμένου αυτού. Επίσης εάν κάποιος σχετίζεται

με πάρα πολλά κείμενα ίσως ο ρόλος του να μην ήταν ιδιαίτερα σημαντικός στην δημιουργία των εγγράφων αυτών και άρα δε θα πρέπει να ληφθούν υπόψιν. Αντίστοιχα εάν πολλά κείμενα που αναφέρονται σε έναν ειδικό και δεν σχετίζονται θεματολογικά με ένα άλλο κείμενο d το κείμενο αυτό ίσως να μην έπρεπε να ληφθεί υπόψιν [20].

2.5 Εύρεση αξιολογητών συνεδρίων

Η προσπάθεια των Karimzadehgan, Xiang Zhai, Belford [2]. Επικεντρώνεται στην ανάκτηση ειδικών για την αξιολόγηση δημοσιεύσεων. Η κεντρική ιδέα είναι ότι μία δημοσίευση δεν αφορά αποκλειστικά και μόνο έναν επιστημονικό τομέα. Για παράδειγμα μια δημοσίευση με τίτλο «Εφαρμογή μηχανικής μάθησης για την ανάπτυξη νέων μοντέλων ανάκτησης για αναζήτηση στον ιστό» αφορά τρεις τομείς ενδιαφέροντος: Μηχανική μάθηση, Μοντέλα ανάκτησης δεδομένων, Αναζήτηση στον ιστό. Άρα αυτός που θα αξιολογήσει την δημοσίευση θα πρέπει να καλύπτει και τους τρεις τομείς. Ακολουθούνται δύο βασικές στρατηγικές για την επίλυση του προβλήματος. Η πρώτη είναι να μοντελοποιηθούν οι αξιολογητές για τους τομείς στου οποίους είναι καλοί με βάση τις δικιές τους δημοσιεύσεις και στην συνέχεια να γίνει η αντιστοίχιση με την δημοσίευση προς αξιολόγηση. Η δεύτερη στρατηγική επικεντρώνεται στο να βρεθούν όλοι οι τομείς ενδιαφέροντος που αφορά μια δημοσίευση. Ανάλογα με το πλήθος των αξιολογητών που απαιτούνται το κείμενο διαχωρίζεται σε n τομείς και για κάθε τομέα βρίσκεται ο σχετικότερος ειδικός. Όσοι περισσότεροι είναι οι ειδικοί τόσοι και οι τομείς που χωρίζεται το κείμενο. Στο τελευταίο βήμα συγκεντρώνονται όλοι οι ειδικοί και εάν κάποιος ειδικός επιλέχθηκε σε δύο τομείς τότε κρατείται μόνο ο τομέας στον οποίο έχει την μεγαλύτερη βαθμολογία. Η μέθοδος τους επικεντρώνεται στην ανάθεση αξιολογητών για μια δημοσίευση τη φόρα και όχι σε γενικότερη αντιστοίχιση ειδικών με λέξεις κλειδιά.

Χτίζοντας πάνω σε αυτήν την ιδέα οι Karimzadehgan, Xiang Zhai [21]. Παρουσιάζουν το επόμενο βήμα όπου μπαίνουν ως παράμετροι οι διαθέσιμοι αξιολογητές αλλά και ο μέγιστος αριθμός των δημοσιεύσεων που μπορεί να αξιολογήσει ένας ειδικός. Αρχικά τα κείμενα ταξινομούνται με το πλήθος των επιστημονικών τομέων που καλύπτουν από το μεγαλύτερο στο μικρότερο. Στην συνέχεια σε κάθε στάδιο ανάθεσης ο ειδικός που καλύπτει τους περισσότερους τομείς για ένα κείμενο επιλέγεται να το αξιολογήσει ελέγχοντας ταυτόχρονα το όριο για το πλήθος των αξιολογητών ανά δημοσίευση και το πλήθος των δημοσιεύσεων που μπορούν να αξιολογηθούν από έναν ειδικό. Η πρόταση αυτή στοχεύει σε ένα πλήρως αυτοματοποιημένο σύστημα για την ανάθεση δημοσιεύσεων προς αξιολόγηση και όχι γενικότερα στην εύρεση ειδικών σύμφωνα με συγκεκριμένους όρους αναζήτησης.

3 Μοντελοποίηση και παρουσίαση του προβλήματος

Το πρόβλημα της εύρεσης ειδικών αντικατοπτρίζει μια πολύ συχνά εμφανιζόμενη ανάγκη για εξαγωγή γνώσης από ένα σύνολο πληροφοριών όπως για παράδειγμα το σενάριο της εύρεσης αξιολογητών δημοσιεύσεων. Στο σενάριο αυτό, στόχος είναι να βρεθεί ένα σύνολο αξιολογητών βάσει λέξεων κλειδιών οι οποίες αντιστοιχούν σε γνωστικές περιοχές. Το πρόβλημα μπορεί να γενικευτεί, για οποιαδήποτε περίπτωση χρειάζεται να αντληθεί μια ομάδα οντοτήτων Α οι οποίες καλύπτουν ένα σύνολο ιδιοτήτων Β.

ΟΡΙΣΜΟΣ 1. (Αναζήτηση/λέξεις κλειδιά/τομείς εξειδίκευσης)

Η είσοδος του προβλήματος αποτελείται από λέξεις κλειδιά. Η λέξεις αυτές συγκροτούν το σύνολο για την αναζήτηση των ειδικών, έστω Q.

$$Q = \{q_1, \ldots, q_m\}$$

ΟΡΙΣΜΟΣ 2. (Ειδικοί /Ερευνητές)

Η αναζήτηση θα πρέπει να πραγματοποιηθεί πάνω σε δεδομένα που βρίσκονται σε μία συγκεκριμένη πηγή. Η πηγή αυτή διαθέτει ένα σύνολο ειδικών έστω R'.

$$R' = \{r_1, \ldots, r_n\}$$

ΟΡΙΣΜΟΣ 3. (Ψευδοκείμενο)

Ως ψευδοκείμενο d_i όριζεται το σύνολο της πληροφορίας που υπάρχει για κάθε ειδικό r_i. Καθώς η πληροφορία είναι σε μορφή κειμένου, το d_i μπορεί να αποτελείται από συνένωση κειμένων και να αναπαρίσταται ως σύνολο με στοιχεία τις λέξεις τους.

$$d_i=\{t_1,\ldots,t_p\}$$

Συνδυάζοντας τα ψευδοκείμενα και τους ερευνητές δημιουργείται το σύνολο R

$$R = \{\{r_1, d_1\},\ldots,\{r_n, d_n\}\}$$

ΟΡΙΣΜΟΣ 2. (Προφίλ)

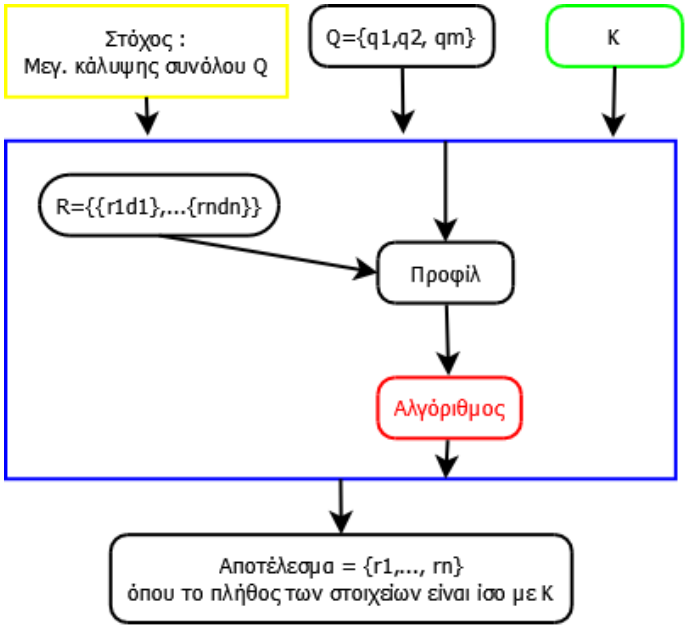
Προφίλ ονομάζεται η ιδεατή οντότητα η οποία παρέχει την πληροφορία για το γνωστικό αντικείμενο του ειδικού r_i καθώς και μία ένδειξη *w* για το πόσο καλός είναι στο αντικείμενο αυτό. Το προφίλ δημιουργείται κάθε φορά που πραγματοποιείται αναζήτηση.

$$r_1: \{q_1, w_1\}, \ldots, \{q_j, w_j\}$$

$$r_2: \{q_1, w_{11}\}, \ldots, \{q_{jj}, w_{jj}\}$$

$$r_3: \{q_1, w_{15}\}, \ldots, \{q_{jjj}, w_{jjj}\}$$

…



Εικόνα 3 Μοντελοποίηση προβλήματος

Ζητούμενο του προβλήματος είναι να βρεθεί ένα σύνολο ειδικών. Εκ των προτέρων είναι γνωστό το πλήθος των στοιχείων του συνόλου αυτού. Δηλαδή μαζί με το σύνολο *Q* εισάγεται και ένας αριθμός *K* ο οποίος συμβολίζει τον προσδοκώμενο αριθμό αποτελεσμάτων. Ο αριθμός αυτός χρησιμοποιείται ως παράμετρος καθώς σε ένα πραγματικό περιβάλλον μπορεί να χρειάζονται περισσότερα του ενός άτομα για να καλύψουν το ίδιο γνωστικό αντικείμενο. Για παράδειγμα σε ένα συνέδριο εάν υπάρχουν πολλές δημοσιεύσεις με θέμα τις βάσεις δεδομένων δεν είναι δυνατό να τις αξιολογήσει όλες ένας και μόνο άνθρωπος. Υπό περίπτωση των παραπάνω αποτελεί η εύρεση του ελάχιστου δυνατού συνόλου που καλύπτει το *Q*. Με την προϋπόθεση ότι το ελάχιστο πλήθος των αποτελεσμάτων είναι γνωστό, το ερώτημα είναι ακριβώς το ίδιο με το να ανατεθεί στο *K* το πλήθος αυτό.

ΠΡΟΒΛΗΜΑ 1. Κάλυψη συνόλου ειδικών

Να βρεθεί το ελάχιστο σύνολο L ερευνητών $L \subseteq R$, για τους οποίους ισχύει

$$(r_i \cup \dots \cup r_j) \cap Q = Q$$

Στην περίπτωση που λαμβάνεται υπόψιν ο παράγοντας βάρους *w* θα πρέπει ταυτόχρονα να ικανοποιείται και η συνθήκη

$$\sum_{a=i}^{j} w_a = MAX$$

Πρόβλημα 2. Μέγιστη κάλυψη συνόλου ειδικών

Το πρόβλημα της μέγιστης κάλυψης προσθέτει την παράμετρο *K*. Δοθέντος ενός αριθμού K, να βρεθεί σύνολο *L* όπου $|L| = K$ και $L \subseteq R$ που καλύπτει όσο το δυνατόν περισσότερα στοιχεία του Q δηλαδή μεγιστοποιείται η παρακάτω παράσταση

$$|(r_1 \cup \dots \cup r_L) \cap Q|$$

Στην περίπτωση που λαμβάνεται υπόψιν ο παράγοντας βάρους *w* θα πρέπει ταυτόχρονα να ικανοποιείται και η συνθήκη

$$\sum_{a=i}^{j} w_a = MAX$$

Τα προβλήματα με τον τρόπο που ορίστηκαν αντιστοιχούν σε ήδη γνωστά προβλήματα στον χώρο των μαθηματικών και της πληροφορικής. Πιο συγκεκριμένα το πρόβλημα 1 αντιστοιχεί στην κάλυψη συνόλου (Set Cover problem) ενώ το πρόβλημα 2 στην μέγιστη κάλυψη συνόλου (Maximum Coverage problem ή max k-cover) [22,23].

Δοθείσης μιας συλλογής Φ από υποσύνολα του *S*= {1,…,n} κάλυψη συνόλου (Set Cover) είναι το πρόβλημα της επιλογής όσο το δυνατόν λιγότερων υποσυνόλων από το Φ έτσι ώστε η ένωση τους να κάνει το *S.* Αντίστοιχα μέγιστη κάλυψη (max k-cover) είναι το πρόβλημα της επιλογής k υποσυνόλων από την συλλογή Φ έτσι ώστε η ένωσή τους να έχει μέγιστη πληθικότητα. Καθώς και τα δύο ανήκουν στην οικογένεια NP-hard δεν μπορούν να λυθούν σε πολυωνυμικό χρόνο και για αυτό χρησιμοποιείται ένας άπληστος αλγόριθμος [24]. Ο αλγόριθμος επιλέγει σύνολα με βάση έναν κανόνα. Σε κάθε βήμα διαλέγει το σύνολο με τον μεγαλύτερο αριθμό μη επιλεγμένων στοιχείων. Η προσέγγιση αυτή δίνει την καλύτερη δυνατή επίλυση σε πολυωνυμικό χρόνο [25]. Αρκεί λοιπόν να χρησιμοποιηθούν οι μέθοδοι που λύνουν τα γνωστά αυτά προβλήματα για να επιτευχθεί το ζητούμενο αποτέλεσμα.

4 Μέθοδος επίλυσης

4.1 Εισαγωγή

Για την ευκολότερη κατανόηση του τρόπου επίλυσης θα χρησιμοποιηθεί παράλληλα και ένα ιδεατό παράδειγμα. Η μέθοδος αναλύεται περεταίρω σε πέντε διακριτά στάδια τα οποία θα παρουσιαστούν εκτενέστερα στην συνέχεια του κεφαλαίου.

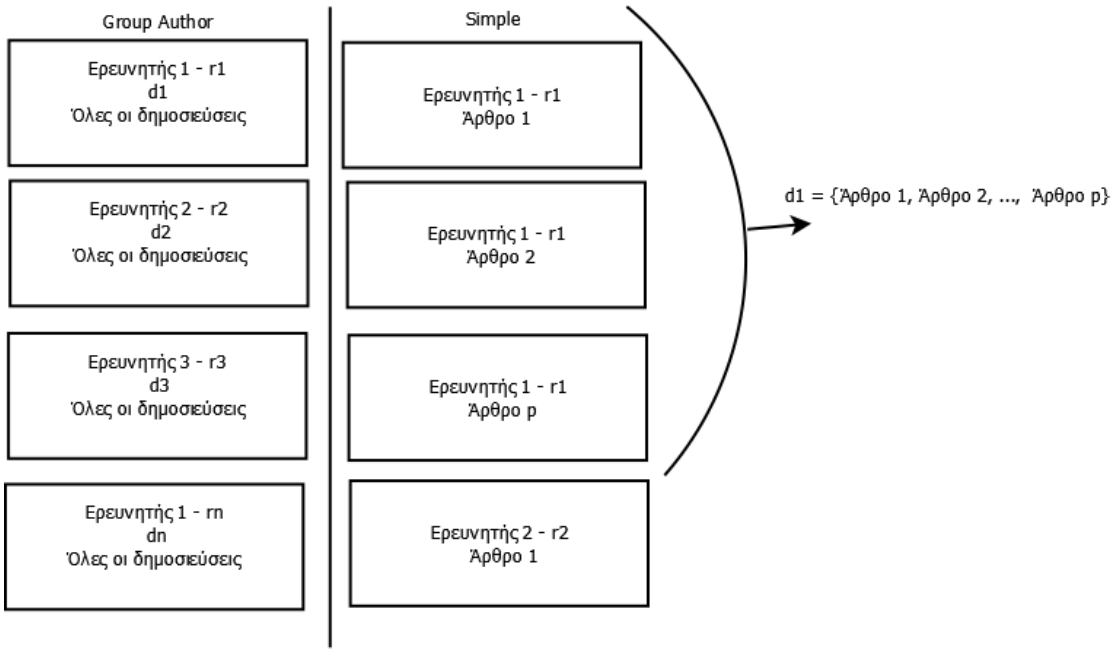
Εικόνα 4 Στάδια μεθόδου επίλυσης

4.2 Ευρετήριο

Έστω για κάθε ερευνητή r_i , υπάρχει ένα ψευδοκείμενο d_i, το οποίο περιγράφει τους τομείς στους οποίους είναι εξειδικευμένος. Για τον τρόπο δημιουργίας των ψευδοκειμένων δεν τίθεται κάποιος περιορισμός καθώς υπάρχουν πολλοί τρόποι όπως π.χ. βιογραφικά. Στην παρούσα προσέγγιση το ψευδοκείμενο d_i αποτελείται από το σύνολο των τίτλων δημοσιεύσεων του r_i που έχει συγγράψει. Διαισθητικά, το d_i περιέχει όρους που αντικατοπτρίζουν τους τομείς ενδιαφέροντος του r_i. Υπό προϊόν αυτής της αναπαράστασης, είναι το γεγονός ότι η συχνότητα εμφάνισης ενός όρου στο d_i συνδέεται με τον αριθμό των δημοσιεύσεων που έχουν γραφτεί πάνω στο θέμα και άρα είναι μια ένδειξη για την εξειδίκευση του ερευνητή. Καθώς οι γνώσεις και η εμπειρία των ερευνητών κυμαίνονται ανάλογα με το γνωστικό αντικείμενο, όσο μεγαλύτερη η συχνότητα ενός όρου στο d_i τόσο “καλύτερος” είναι στον τομέα αυτό.

Τα ψευδοκείμενα αποθηκεύονται σε ευρετήριο με σκοπό την αποτελεσματικότερη δυνατότητα για αναζήτηση και πρόσβαση. Για λόγους πληρότητας στην παρούσα εργασία δημιουργήθηκε και ένας δεύτερος τύπος ευρετηρίου ο οποίος αντί για ψευδοκείμενα χρησιμοποιεί ξεχωριστά κάθε τίτλο. Σκοπός είναι η δοκιμή της μεθόδου σε ένα πραγματικό σενάριο όπου θα πρέπει να γίνει αναζήτηση σε μια βάση δεδομένων που περιέχει ερευνητές και τους τίτλους από τις δημοσιεύσεις τους, για να διαπιστωθεί εάν η προσέγγιση που ακολουθείται θα απαιτήσει και το βήμα της συνένωσης των τίτλων σε ένα ψευδοκείμενο ή όχι. Επιπλέον οι δύο τύποι ευρετηρίου διαφοροποιούνται περαιτέρω, με βάση το εάν έχουν αφαιρεθεί οι καταλήξεις των λέξεων ή όχι (word stemming).

Για την ευρετηριοποίηση των δεδομένων αλλά και για την αναζήτηση χρησιμοποιήθηκε η βιβλιοθήκη Lucene η οποία παρέχεται από τον οργανισμό Apache.



Εικόνα 5 Δομή ευρετηρίων

GroupAuthor: Το ευρετήριο περιέχει τις δημοσιεύσεις των συγγραφέων συγκεντρωτικά.

Κλειδί: Όνομα Συγγραφέα	Περιεχόμενα
Ερευνητής 1	Όλες οι δημοσιεύσεις του ερευνητή 1 –(d1)
Ερευνητής 2	Όλες οι δημοσιεύσεις του ερευνητή 2 –(d2)
Ερευνητής 3	Όλες οι δημοσιεύσεις του ερευνητή 3 –(d3)
Ερευνητής 4	Όλες οι δημοσιεύσεις του ερευνητή 4 –(d4)
…	…

Simple: Το ευρετήριο περιέχει τις δημοσιεύσεις των συγγραφέων αναλυτικά.

Κλειδί: Όνομα Συγγραφέα	Περιεχόμενα	
Ερευνητής 1	Άρθρο Α	d1
Ερευνητής 1	Άρθρο Β	
Ερευνητής 1	Άρθρο Γ	
Ερευνητής 2	Άρθρο Δ	d2
Ερευνητής 2	Άρθρο Ε	
Ερευνητής 4	Άρθρο ΣΤ	…
…	…	

4.3 Δημιουργία Προφίλ

Αρχικά για κάθε όρο $q_i \in Q$ γίνεται αναζήτηση στο ευρετήριο Lucene και επιστρέφεται μία ταξινομημένη λίστα, ψευδοκειμένων *d,* με βάση την σχετικότητα τους ως προς το q_i. Μετά από την επεξεργασία όλων των όρων δημιουργείται για κάθε ερευνητή r_j ένα υποσύνολο ζευγαριών $\{q_j, w_j\}$ όπου w η βαθμολογία που έδωσε το Lucene για το ψευδοκείμενο *d*. Προφίλ ονομάζεται η αντιστοίχιση που γίνεται μετά από την παραπάνω διαδικασία, μεταξύ των λέξεων κλειδιών και των σχετικών με αυτά ερευνητών.

$$r_1: \{q_1, w_1\}, \dots, \{q_j, w_j\}$$

$$r_2: \{q_1, w_{11}\}, \dots, \{q_{jj}, w_{jj}\}$$

$$r_3: \{q_1, w_{15}\}, \dots, \{q_{jjj}, w_{jjj}\}$$

…

Philip S. Yu:{{Uncertainty, 3.0},{ Systems Performance, 4.0},{ Transaction Processing, 32.0},{Knowledge Discovery, 8.0},{ Cloud Computing, 2.0}}

Bhavani M. Thuraisingham:{{Security Privacy, 4.0},{ Map Reduce, 4.0 }}

Christos Faloutsos:{{Text Databases, 4.0}, { Graph Management, 2.0}}

David Taniar:{Parallel Distributed, 2.0}

Z. M. Ma:{Database Models, 10.0}

Debrup Chakraborty:{Authenticated Query Processing, 3.0}

Το Lucene βαθμολογεί τα στοιχεία του ευρετηρίου του με βάση μια προ υπάρχουσα συνάρτηση η οποία ονομάζεται TFIDFSimilarity. Εκτός από αυτή, αναπτύχθηκε μια δεύτερη μέθοδος βαθμολόγησης η οποία λαμβάνει υπόψιν της μόνο την συχνότητα εμφάνισης του όρου q_i μέσα στο d_i.

4.3.1 TFIDF Similarity

Η προεπιλεγμένη συνάρτηση βαθμολόγησης του Lucene είναι η [26]

$$score(q,d) = coord(q,d) * queryNorm(q) * \sum_{t\ in\ q} (tf(t\ in\ d) * idf(t)^2 * t.getBoost() * norm(t,d))$$

Όπου οι επιμέρους παράγοντες είναι

tf(t in d) Η συχνότητα εμφάνισης του όρου t μέσα στο κείμενο d. Όσο περισσότερες φορές εμφανίζεται ο όρος t τόσο μεγαλύτερη είναι η βαθμολογία.

$$\text{tf(t in d)} = frequency^{1/2}$$

idf(t) Η αντίστροφη συχνότητα εμφάνισης του t. Στον υπολογισμό αυτό οι σπανιότεροι όροι έχουν μεγαλύτερη επίδραση.

$$\text{idf(t)} = 1 + \log\left(\frac{numDocs}{docFreq+1}\right)$$

coord(q,d) Ο συντελεστής μετράει πόσες από τις λέξεις αναζήτησης q βρίσκονται μέσα στο κείμενο d.

queryNorm(q) Ο συντελεστής κανονικοποιεί τις βαθμολογίες έτσι ώστε να είναι συγκρίσιμες (Ευκλείδεια Νόρμα) χωρίς να επηρεάζει την κατάταξη των εγγράφων.

$$\text{queryNorm(q)} = \text{queryNorm(sumOfSquaredWeights)} = \frac{1}{sumOfSquaredWeights^{1/2}}$$

t.getBoost() Χρησιμοποιείται κατά την αναζήτηση όταν υπάρχει κάποιος όρος στον οποίο πρέπει να δοθεί περισσότερη σημασία.

norm(t,d) Συμπεριλαμβάνονται συντελεστές οι οποίοι υπολογίστηκαν όταν δημιουργήθηκε το ευρετήριο ανάλογα με την σημαντικότητα που έχει το κάθε πεδίο.

4.3.2 Term Count (TC)

Εκτός της προεπιλεγμένης συνάρτησης του Lucene υλοποιήθηκε και μία επιπλέον η οποία μετρά μόνο το πόσες φορές εμφανίζεται ένας όρος αναζήτησης q μέσα σε ένα κείμενο d του ευρετηρίου.

$$score(q,d) = \sum_{q\ in\ d} 1$$

4.4 Περιορισμοί

Για την αποτελεσματικότερη αντιμετώπιση του προβλήματος, εφαρμόζονται περιορισμοί έτσι ώστε η τελική λίστα να έχει όσο το δυνατόν πιο σχετικά αποτελέσματα. Έστω η παρακάτω λίστα βαθμολόγησης για τον όρο Index.

Συγγραφέας	Θέση	Βαθμολογία
Leo Egghe	1	34
Ivan Gutman	2	28
Ronald Rousseau	3	19
Gonzalo Navarro	4	18
Hans-Peter Kriegel	5	16
Michael Schreiber	6	16
Lutz Bornmann	7	15
Sandi Klavzar	8	15
Bolian Liu	9	14
Chin-Chen Chang	10	14

4.4.1 Ημερομηνία δημοσίευσης

Για κάθε συγγραφέα λαμβάνονται υπόψιν μόνο οι δημοσιεύσεις από μια συγκεκριμένη ημερομηνία και μετά. Λόγω της φύσης του περιορισμού υπάρχει δυνατότητα εφαρμογής του μόνο στο ευρετήριο Simple καθώς είναι το μοναδικό που κρατάει αναλυτικά όλες τις δημοσιεύσεις για κάθε συγγραφέα.

4.4.2 Μέγιστη θέση ανά όρο

Για κάθε όρο κρατούνται μόνο οι πρώτοι S συγγραφείς σύμφωνα με την βαθμολογία τους.

Έστω S = 5

Συγγραφέας	Θέση	Βαθμολογία
Leo Egghe	1	34
Ivan Gutman	2	28
Ronald Rousseau	3	19
Gonzalo Navarro	4	18
Hans-Peter Kriegel	5	16
Michael Schreiber	6	16
Lutz Bornmann	7	15
Sandi Klavzar	8	15
Bolian Liu	9	14
Chin-Chen Chang	10	14

4.4.3 Ελάχιστη βαθμολογία ανά όρο

Για κάθε όρο κρατούνται μόνο οι πρώτοι S συγγραφείς των οποίων η βαθμολογία είναι μεγαλύτερη ή ίση με βαθμολογία W.

Έστω W = 16

Συγγραφέας	Θέση	Βαθμολογία
Leo Egghe	1	34
Ivan Gutman	2	28
Ronald Rousseau	3	19
Gonzalo Navarro	4	18
Hans-Peter Kriegel	5	16
Michael Schreiber	6	16
Lutz Bornmann	7	15
Sandi Klavzar	8	15
Bolian Liu	9	14
Chin-Chen Chang	10	14

4.4.4 Απόσταση όρων

Για κάθε συγγραφέα κρατούνται μόνο οι πρώτοι S όροι με απόσταση μικρότερη ή ίση με την απόσταση που ορίζει ο χρήστης.

Έστω οι όροι αναζήτησης: q_1, q_2, q_3, q_4, q_5 και απόσταση όρων ίση με 30.

Για τον συγγραφέα Σ1

q_1: 1η θέση

q_2: 10η θέση

q_3: 500η θέση

q_4: 501η θέση

q_5: 1000η θέση

Άρα μόνοι οι όροι Ο1 και Ο2 θα ληφθούν υπόψιν σε περαιτέρω υπολογισμούς.

Παρακάτω ακολουθούν πραγματικά παραδείγματα:

1. Έστω οι όροι αναζήτησης: index,storage,network και απόσταση όρων ίση με 500.
   - Για τον συγγραφέα Alexandros G. Dimakis
     Index: 420η θέση
     Storage: 6η θέση
     Network: 1047η θέση
     Άρα μόνοι οι όροι Storage και Index θα ληφθούν υπόψιν σε περαιτέρω υπολογισμούς.
2. Έστω οι όροι αναζήτησης: storage, index, network και απόσταση όρων ίση με 20.
   - Για τον συγγραφέα Yue Li
     Index: 5229η θέση

Network: 42636η θέση

Άρα θα μόνο ο όρος index θα ληφθεί υπόψιν στην τελική βαθμολόγηση του συγγραφέα.

- Για τον συγγραφέα Yu Wu

Index: 4710η θέση

Storage: 4997η θέση

Network: 41190η θέση

Άρα θα μόνο ο όρος index θα ληφθεί υπόψιν στην τελική βαθμολόγηση του συγγραφέα.

3. Έστω οι όροι αναζήτησης: storage, index, network και απόσταση όρων ίση με 300.
   - Για τον συγγραφέα Yue Li

   Index: 5229η θέση

   Network: 42636η θέση

   Άρα θα μόνο ο όρος index θα ληφθεί υπόψιν στην τελική βαθμολόγηση του συγγραφέα.

   - Για τον συγγραφέα Yu Wu

   Index: 4710η θέση

   Storage: 4997η θέση

   Network: 41190η θέση

   Άρα θα μόνο οι όροι index και storage θα ληφθούν υπόψιν στην τελική βαθμολόγηση του συγγραφέα.

4.5 Συνολική Βαθμολόγηση

Η διαδικασία της βαθμολόγησης ξεκινά από την δημιουργία των προφίλ. Στο σημείο αυτό κάθε όρος αναζήτησης εισάγεται στο Lucene και επιστρέφεται μία βαθμολογία ανά όρο και ειδικό. Στην συνέχεια τα δεδομένα περνούν στο στάδιο των περιορισμών και καταλήγουν στον υπολογισμό της συνολικής βαθμολόγησης. Αυτό σημαίνει ότι πλέον θα υπάρχει ένας και μόνο αριθμός που αντιπροσωπεύει την εξειδίκευση ενός ατόμου.

Έστω ο συγγραφέας Alexandros G. Dimakis και οι όροι αναζήτησης index, storage, network.

Όρος αναζήτησης	TFIDF	TC	Πλήθος σχετικών άρθρων
index	0.62	3	3
storage	1.17	29	29
network	0.54	12	12

Η συνολική βαθμολογία προκύπτει κάθε φορά από το άθροισμα των επιμέρους βαθμολογιών ανά όρο. Θα πρέπει να σημειωθεί ότι αλγόριθμος Set Cover που έπεται της παρούσας διαδικασίας λειτουργεί επαναληπτικά. Αρχικά λοιπόν ο Alexandros G. Dimakis έχει σύνολο 2.33 με TFIDF και 44 με TC. Στον επόμενο κύκλο όμως εάν για παράδειγμα επιλεχθεί ο ειδικός X για τον όρο index τότε ο Alexandros G. Dimakis θα έχει 1.71 με TFIDF και 41 με TC. Δηλαδή η συνολική βαθμολογία μεταβάλλεται σε κάθε επανάληψη του αλγορίθμου καθώς θα πρέπει να λαμβάνει υπόψιν μόνο τους όρους που δεν επιλέχθηκαν ακόμα.

4.6 Ιδεατό παράδειγμα εκτέλεσης Greedy Set Cover

Έστω το σύνολο από ειδικούς

$$R = \{\{r_1,d_1\},\ldots,\{r_n,d_n\}\}$$

R = {{auth1,'Relational Databases Network Database Algorithms Faster Indexing systems text mining systems'},{auth2,' database semantics network semantics'} …}

Όπου r ο συγγραφέας και d ένα κείμενο που περιγράφει το γνωστικό του αντικείμενο. Για την δική μας προσέγγιση το d αποτελεί ένα ψευδοκείμενο από την συνένωση των τίτλων όλων των δημοσιεύσεων του συγγραφέα (r). Ως d θα μπορούσε να χρησιμοποιηθεί οποιαδήποτε πληροφορία προβάλει αποδοτικά το πεδίο ενδιαφέροντος και έρευνας του.

Η είσοδος στο αλγόριθμο αποτελείται από το σύνολο R και επιπλέον από το σύνολο

$$Q = \{ q_1, q_2,\ldots q_m\}$$

το οποίο είναι οι λέξεις κλειδιά τα οποία επιλέγει ο χρήστης.

Επιπλέον υπάρχει και ένας αριθμός K ο οποίος συμβολίζει το πλήθος των συγγραφέων που θα πρέπει να επιστραφούν από τον αλγόριθμο. Σκοπός είναι να επιστραφούν K ερευνητές οι οποίοι καλύπτουν όσο το δυνατόν περισσότερες λέξεις.

$$Q = \{ q_1, q_2,\ldots q_m\}$$

Q = { database, indexing, network, semantics, vlsi, text mining, systems}

Το πρώτο στάδιο είναι η δημιουργία προφίλ με βάση τις λέξεις κλειδιά. Για κάθε λέξη κλειδί μετράται ο αριθμός εμφάνισης(Term count) αυτής μέσα σε κάθε ψευδοκείμενο. Ο αριθμός αυτός αποτελεί την βαθμολογία του εκάστοτε συγγραφέα για την συγκεκριμένη λέξη κλειδί. Η βαθμολογία εκφράζει το πόσο σχετικός είναι ένας συγγραφέας με τον όρο αναζήτησης και όσο μεγαλύτερη, τόσο πιο εξειδικευμένος είναι ο συγγραφέας. Από την παραπάνω διαδικασία προκύπτει ένα νέο σύνολο όπου για κάθε συγγραφέα αντιστοιχεί και μια βαθμολογία για κάθε λέξη κλειδί.

$r_1 : q_1 w_1, q_2 w_2, q_3 w_3$

$r_2 : q_1 w'_1, q_2 w'_2, q_3 w'_3$

…

$r_n : q_1 w''_1, \ \ldots q_m \ w_m$

auth1 : database 2, indexing 1, network 1, text mining 1, systems 2

auth2 : database 1, semantics 2, network

auth3 : semantics 1, vlsi 1

auth4 : database 1, indexing 1, network 1, text mining

Με την παραπάνω διάταξη το πρόβλημα μπορεί να μοντελοποιηθεί ως ένα πρόβλημα μέγιστης κάλυψης συνόλου (Max Coverage). Επειδή όμως το Max Coverage ανήκει στην οικογένεια αλγορίθμων NP-hard θα χρησιμοποιηθεί ένας άπληστος αλγόριθμος ο οποίος λύνει το πρόβλημα αποδοτικά χωρίς να εγγυάται όμως βέλτιστη λύση.

Αρχικά ταξινομούνται οι συγγραφείς με βάση το πλήθος των όρων στους οποίους είναι ειδικοί από το μεγαλύτερο στο μικρότερο.

1. auth1 : database 1, indexing 1, network 1, text mining, systems
2. auth4 : database 1, indexing 1, network 1,text mining
3. auth2 : database 1, semantics 1, network 1
4. auth3 : semantics 1, vlsi 1

Επιλέγεται ο πρώτος συγγραφέας και τον αποθηκεύεται ως αποτέλεσμα έστω το σύνολο

L = {auth1}

Οι όροι στους οποίους είναι ειδικός ο auth1 αφαιρούνται από το σύνολο Q και άρα

Q = {semantics, vlsi}

Στην συνέχεια ταξινομούνται οι υπόλοιποι συγγραφείς σύμφωνα με το πλήθος των όρων στους οποίους είναι ειδικοί με βάση τους εναπομείναντες όρους στο Q.

1. auth4 :
2. auth2 : semantics 1
3. auth3 : semantics 1, vlsi 1

(Για το παράδειγμα ο συγγραφέας auth4 θα απορριφθεί καθώς δεν έχει άλλους διαθέσιμους όρους)

Άρα η νέα διάταξη θα είναι

1. auth3 : semantics 1, vlsi 1
2. auth2 : semantics 1

Επιλέγεται και πάλι ο πρώτος συγγραφέας auth3 και πλέον το σύνολο του αποτελέσματος είναι

$$L = \{auth1, auth3\}$$

Με τους δύο αυτούς συγγραφείς έχει καλυφθεί το σύνολο των λέξεων στο Q. Εάν το K = 2 τότε ο αλγόριθμος τερματίζει. Εάν ο χρήστης επέλεξε μεγαλύτερο K τότε η διαδικασία συνεχίζει μέχρι το πλήθος των συγγραφέων προς εμφάνιση να γίνει ίσο με αυτό.

Εάν το σύνολο Q εξαντληθεί πριν φτάσουμε στον K τότε η διαδικασία ξεκινάει από την αρχή. Δηλαδή αρχικοποιείται και πάλι το σύνολο Q

Q = { database, indexing, network, semantics, vlsi, text mining, systems}

και γίνεται ταξινόμηση των συγγραφέων από το σύνολο

$$R' = R - L$$

1. auth4 : database 1, indexing 1, network 1,text mining
2. auth2 : database 1, semantics 1, network 1

Εάν K = 3 τότε το τελικό σύνολο θα είναι

$$L = \{auth1, auth3, auth4\}$$

4.7 Εφαρμογή αλγορίθμου

Για το τελικό αποτέλεσμα της μεθόδου χρησιμοποιείται ένας άπληστος αλγόριθμος (Greedy Algorithm) ο οποίος λύνει αποδοτικά το πρόβλημα και μάλιστα αποτελεί την καλύτερη δυνατή, πολυωνυμικού χρόνου, λύση [27].

Πριν από την εφαρμογή του αλγορίθμου θα πρέπει να γίνει η τελική κατάταξη η οποία συνδυάζει τα αποτελέσματα από όλες τις προηγούμενες λίστες που περιέχουν τους καλύτερους συγγραφείς ανά όρο.

Για τον όρο Index

Συγγραφέας	Θέση	Βαθμολογία
Leo Egghe	1	34.0
Ivan Gutman	2	28.0
Ronald Rousseau	3	19.0
Gonzalo Navarro	4	18.0
Hans-Peter Kriegel	5	16.0
Michael Schreiber	6	16.0
…		
A. A. Krizhanovsky	2212	1.0
A. Al-Badarneh	2213	1.0
A. B. Mutiara	2214	1.0
Yan Chen	2215	1.0

Για τον όρο Storage

Συγγραφέας	Θέση	Βαθμολογία
Dan Feng	1	68.0
Changsheng Xie	2	36.0
Kannan Ramchandran	3	35.0
…		
Yan Chen	1475	3.0

Για τον όρο Network

Συγγραφέας	Θέση	Βαθμολογία
Muriel Médard	1	135.0
Kotaro Hirasawa	2	124.0
Yoshitaka Shibata	3	102.0
…		
Yan Chen	12	52

4.7.1 Greedy Set Cover

Ο αλγόριθμος για να καλύψει πλήρως όλο το σύνολο των στοιχείων του Q χρησιμοποιεί έναν και μόνο κανόνα. Σε κάθε επανάληψη, διαλέγει τον ερευνητή που έχει τους περισσότερους όρους εξειδίκευσης από τους εναπομείναντες όρους αναζήτησης. Σε περίπτωση ισοπαλίας προτεραιότητα έχει ο συγγραφέας με την μεγαλύτερη βαθμολογία.

Έστω το σύνολο των συγγραφέων R = $\{r_1 , r_2 , r_3\}$

Και το σύνολο των όρων αναζήτησης Q = $\{q_1 , q_2 , q_3 , q_4, q_5, q_6\}$

Οι αντίστοιχοι τομείς εξειδίκευσης

r_1 = $\{q_1 , q_2 , q_3, q_4\}$

r_2= { q_3 ,q_4 , q_5}

r_3= $\{q_5, q_6\}$

Εάν η είσοδος K = 1 τότε το αποτέλεσμα είναι ο συγγραφέας r_1

Ενώ για K = 2 στο πρώτο βήμα ο αλγόριθμος επιλέγει τον συγγραφέα r_1και στην συνέχεια τον r_3με τον οποίο καλύπτει πλήρως το σύνολο των όρων αναζήτησης και τερματίζει.

Παράδειγμα

	Θέση	Βαθμολογία	Επιλεγμένοι όροι
Yan Chen	1	56	Index, network, storage
Wei Wang	2	48	Index, network, storage
Alexandros G. Dimakis	3	44	Index, network, storage
…			

Yan Chen
Final score : 56.0
Term: index, Score : 1.0, Lucene Documents: 1
Term: network, Score : 52.0, Lucene Documents: 1
Term: storage, Score : 3.0, Lucene Documents: 1
Selected Set Cover Terms: (Universe Full) #1 index,network,storage

Wei Wang
Final score : 48.0
Term: index, Score : 3.0, Lucene Documents: 1
Term: network, Score : 42.0, Lucene Documents: 1
Term: storage, Score : 3.0, Lucene Documents: 1
Selected Set Cover Terms: (Universe Full) #2 index,network,storage

Alexandros G. Dimakis
Final score : 44.0
Term: index, Score : 3.0, Lucene Documents: 1
Term: network, Score : 12.0, Lucene Documents: 1
Term: storage, Score : 29.0, Lucene Documents: 1
Selected Set Cover Terms: (Universe Full) #3 index,network,storage

4.7.2 Weighted Greedy Set Cover

Στην έκδοση του αλγορίθμου με βάρη, σε κάθε επανάληψη επιλέγεται ο ερευνητής με την μεγαλύτερη βαθμολογία από τους εναπομείναντες όρους αναζήτησης.

Έστω το σύνολο των συγγραφέων R = $\{r_1 , r_2 , r_3\}$

Και το σύνολο των όρων αναζήτησης Q = $\{q_1 , q_2 , q_3 , q_4, q_5, q_6\}$

Οι αντίστοιχοι τομείς εξειδίκευσης

r_1 = $\{\{q_1, w_1\} , \{q_2, w_2\} , \{q_3, w_3\}, \{q_4, w_4\}\}$

r_2= $\{\{q_3, w'_3\} ,\{q_4, w'_4\} ,\{ q_5, w'_5\}\}$

r_3= $\{\{q_5, w''_5\}, \{q_6, w''_6\}\}$

Εάν η είσοδος K = 1 τότε το αποτέλεσμα είναι ο συγγραφέας r_1

Ενώ για K = 2 στο πρώτο βήμα ο αλγόριθμος επιλέγει τον συγγραφέα r_1και στην συνέχεια τον r_3με τον οποίο καλύπτει πλήρως το σύνολο των όρων αναζήτησης και τερματίζει.

Παράδειγμα

Συγγραφέας	Θέση	Βαθμολογία	Επιλεγμένοι όροι
Muriel Médard	1	140.0	Network, storage
Kotaro Hirasawa	2	126.0	Index
Yoshitaka Shibata	3	102.0	
…			

Muriel Médard
Final score : 140.0
Term: network, Score : 135.0, Lucene Documents: 1
Term: storage, Score : 5.0, Lucene Documents: 1
Selected Set Cover Terms: #1 network,storage

Leo Egghe
Final score : 34.0
Term: index, Score : 34.0, Lucene Documents: 1
Selected Set Cover Terms: (Universe Full) #2 index

Kotaro Hirasawa
Final score : 125.0
Term: index, Score : 1.0, Lucene Documents: 1
Term: network, Score : 124.0, Lucene Documents: 1
Selected Set Cover Terms: #3 index,network

4.7.3 Custom Greedy Set Cover

Εκτός από την κλασσική υλοποίηση του Greedy Set Cover, αναπτύχθηκε και ένας επιπλέον αλγόριθμος παρόμοιος με αυτόν αλλά με το μειονέκτημα ότι δεν λύνει αποδοτικά το Πρόβλημα 1 (Κεφ. Μοντελοποίηση και Παρουσίαση προβλήματος). Τα βήματα του αλγορίθμου ακολουθούν παρακάτω.

Έστω το σύνολο των συγγραφέων R = $\{r_1 , r_2 , r_3\}$

Και το σύνολο των όρων αναζήτησης Q = $\{q_1 , q_2 , q_3 , q_4, q_5, q_6\}$

Οι αντίστοιχοι τομείς εξειδίκευσης

r_1 = $\{q_1 , q_2 , q_3, q_4\}$

r_2= { q_3 ,q_4 , q_5}

r_3= $\{q_5, q_6\}$

Πριν την εκκίνηση του αλγορίθμου οι ειδικοί κατατάσσονται ανάλογα με το πλήθος των όρων στους οποίους είναι ειδικοί ή ανάλογα με την τελική τους βαθμολογία στην περίπτωση της έκδοσης με βάρη. Η κατάταξη αυτή είναι ουσιαστικά μία ουρά προτεραιότητας. Σε κάθε βήμα ελέγχεται ο πρώτος ειδικός και εάν αυτός έχει όρους οι οποίοι ακόμα δεν έχουν καλυφθεί τότε επιλέγεται, αλλιώς το βήμα επαναλαμβάνεται με τον επόμενο στην σειρά.

Εάν η είσοδος K = 1 τότε το αποτέλεσμα είναι ο συγγραφέας r_1ενώ για K = 2 στο πρώτο βήμα ο αλγόριθμος επιλέγει τον συγγραφέα r_1και στην συνέχεια τον r_2 γιατί είναι ο επόμενος και καλύπτει τον επιπλέον όρο q_5. Στο σημείο αυτό φαίνεται το μειονέκτημα της μεθόδου, διότι ο κλάσσικος Greedy Set Cover θα επέστρεφε τον ειδικό r_3 με τον οποίο καλύπτει πλήρως το σύνολο των όρων αναζήτησης. Παρόλα αυτά ο αλγόριθμος Custom Greedy Set Cover δίνει πολύ καλά αποτελέσματα τα οποία παρουσιάζονται σε επόμενο κεφάλαιο.

Παράδειγμα

	Θέση	Βαθμολογία	Επιλεγμένοι όροι
Yan Chen	1	56	Index, network, storage
Wei Wang	2	48	Index, network, storage
Alexandros G. Dimakis	3	44	Index, network, storage
…			

Yan Chen
Final score : 56.0
Term: index, Score : 1.0, Lucene Documents: 1
Term: network, Score : 52.0, Lucene Documents: 1
Term: storage, Score : 3.0, Lucene Documents: 1
Selected Set Cover Terms: index,network,storage

Wei Wang
Final score : 48.0
Term: index, Score : 3.0, Lucene Documents: 1
Term: network, Score : 42.0, Lucene Documents: 1
Term: storage, Score : 3.0, Lucene Documents: 1
Selected Set Cover Terms: Additional

Alexandros G. Dimakis
Final score : 44.0
Term: index, Score : 3.0, Lucene Documents: 1
Term: network, Score : 12.0, Lucene Documents: 1
Term: storage, Score : 29.0, Lucene Documents: 1
Selected Set Cover Terms: Additional

4.7.4 Custom Greedy Set Cover Weighted

$r_1 = \{\{q_1, w_1\}, \{q_2, w_2\}, \{q_3, w_3\}, \{q_4, w_4\}\}$

$r_2 = \{\{q_3, w'_3\}, \{q_4, w'_4\}, \{q_5, w'_5\}\}$

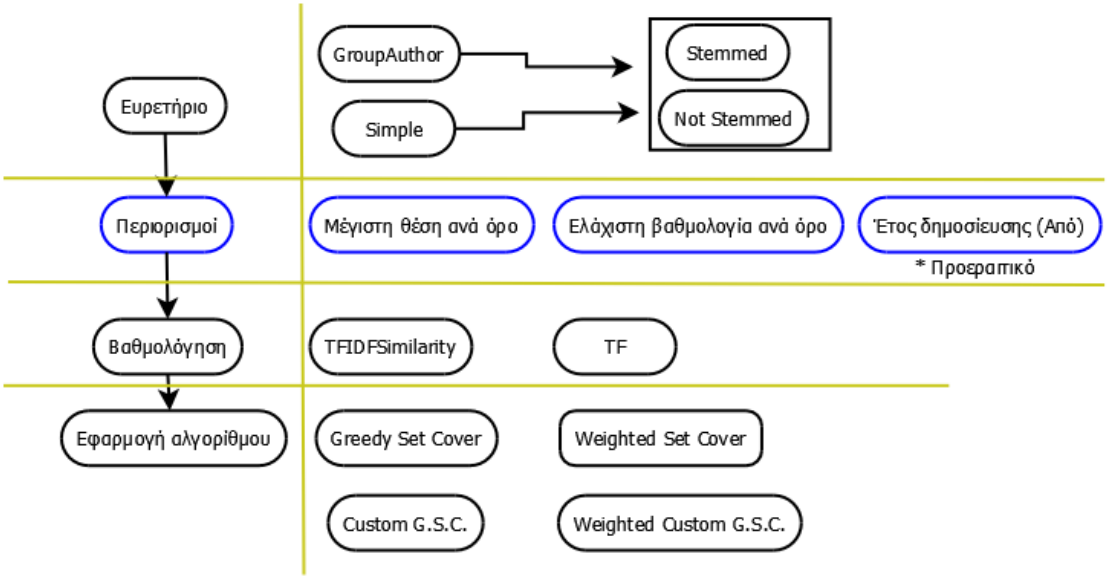
$r_3 = \{\{q_5, w''_5\}, \{q_6, w''_6\}\}$

Συγγραφέας	Θέση	Βαθμολογία	Επιλεγμένοι όροι
Muriel Médard	1	140.0	Network, storage
Kotaro Hirasawa	2	126.0	Index
Yoshitaka Shibata	3	102.0	
…			

Muriel Médard
Final score : 140.0
Term: network, Score : 135.0, Lucene Documents: 1
Term: storage, Score : 5.0, Lucene Documents: 1
Selected Set Cover Terms: network,storage

Kotaro Hirasawa
Final score : 125.0
Term: index, Score : 1.0, Lucene Documents: 1
Term: network, Score : 124.0, Lucene Documents: 1
Selected Set Cover Terms: index

Yoshitaka Shibata
Final score : 102.0
Term: network, Score : 102.0, Lucene Documents: 1
Selected Set Cover Terms: Additional



Εικόνα 6 Δυνατοί συνδυασμοί μεθόδων

5 Αρχιτεκτονική

5.1 Ανάλυση απαιτήσεων

Στα πλαίσια της διπλωματικής αναπτύχθηκε μια εφαρμογή η οποία υλοποιεί την προτεινόμενη μέθοδο επίλυσης του προβλήματος της εύρεσης ειδικών. Για την καλύτερη και αποτελεσματικότερη σχεδίαση της εφαρμογής ακολουθήθηκε η διαδικασία της ανάλυσης μαλακών συστημάτων (Soft Systems Methodology, SSM). Η μεθοδολογία στοχεύει στην αντιμετώπιση προβλημάτων του πραγματικού κόσμου που αφορούν στην αλληλεπίδραση μεταξύ ανθρώπου και μηχανής. Αν και μέσα στην διάρκεια πολλών ετών έχουν περιέλθει πολλές αλλαγές στην μορφή της SSM, η αρχική έκδοση της περιλαμβάνει επτά (7) διακριτά στάδια [3].

1. Εισαγωγή στην προβληματική κατάσταση
2. Εκφράζοντας την προβληματική κατάσταση
3. Θεμελιακός ορισμός συστημάτων
4. Ιδεατά μοντέλα συστημάτων
5. Σύγκριση μοντέλων με τον πραγματικό κόσμο
6. Καθορισμός αλλαγών που είναι θεμιτές και εφικτές
7. Ενέργειες για να βελτιωθεί η υπάρχουσα κατάσταση

Όλα τα παραπάνω βήματα δεν είναι υποχρεωτικά σε κάθε εφαρμογή της μεθόδου. Αντιθέτως αλλάζουν ανάλογα με την έκταση και τις προϋποθέσεις του κάθε έργου. Καθώς το παρόν έργο έχει μικρή έκταση και είναι πολύ εξειδικευμένο δεν θα χρησιμοποιηθεί το βήμα 6 το οποίο προσανατολίζεται για εφαρμογές μεγαλύτερης κλίμακας.

5.1.1 Εισαγωγή στην προβληματική κατάσταση

Ξεκινώντας την ανάλυση θα γίνει αναφορά στην ανάγκη για την δημιουργία ενός προγράμματος αναζήτησης ειδικών. Στην σημερινή εποχή υπάρχει ένας όλο και αυξανόμενος όγκος πληροφοριών. Αντίστοιχα προκαλείται μια όλο και αυξανόμενη ανάγκη για την αξιολόγηση των πληροφοριών αυτών. Πιο συγκεκριμένα υπάρχει η τάση για αναζήτηση ατόμων ή ομάδες ατόμων οι οποίες εξειδικεύονται σε συγκεκριμένους τομείς ενδιαφέροντος. Ακόμα και οι μηχανές αναζήτησης αδυνατούν να επιστρέψουν την εν λόγω πληροφορία καθώς στοχεύουν στο να βρουν κείμενα ή αναφορές σε κείμενα, τα οποία φιλοξενούνται σε ιστοσελίδες και όχι πρόσωπα.

5.1.2 Εκφράζοντας την προβληματική κατάσταση

Μέχρι στιγμής δεν υπάρχει ολοκληρωμένη και εύκολη (για τον χρήστη) μέθοδος για την εύρεση ειδικών. Θα πρέπει να υπάρχει δυνατότητα μέσα από ένα σύνολο ατόμων , με βάση λέξεις κλειδιά να εξάγεται ένα σύνολο Κ ειδικών το οποίο να καλύπτει πλήρως τις λέξεις αυτές.

5.1.3 Θεμελιακός ορισμός

Ένα σύνηθες πρόβλημα είναι ότι προσπαθώντας κάποιος να ορίσει ένα σύστημα συγχέει το ποιος εκτελεί τις εργασίες του συστήματος και ποιες είναι αυτές. Ο θεμελιακός ορισμός εκφράζει το σκοπό ενός συστήματος με δομημένο τρόπο. Για την δημιουργία κατάλληλου θεμελιακού ορισμού χρησιμοποιήθηκε η μέθοδος CATWOE η οποία αποτελεί έναν μνημονικό κανόνα για την σωστή δημιουργία του [28]. Εστιάζοντας στην περιγραφή των απαραίτητων στοιχείων που συνθέτουν ένα σύστημα αλληλεπίδρασης ανθρώπου – μηχανής η τεχνική δίνει την απαραίτητη καθοδήγηση για έναν πλήρη θεμελιακό ορισμό.

Η λέξη CATWOE αποτελείται από τα αρχικά των αγγλικών λέξεων Clients, Actors, Transformation, Worldview, Owner, Environmental constraints. Παρακάτω αναλύονται όλα τα βήματα της διαδικασίας.

Πελάτες (Clients)

Στα επιστημονικά συνέδρια η σύσταση επιτροπών αξιολόγησης είναι μια διαδικασία χρονοβόρα και πολύπλοκη. Θα πρέπει σε κάθε αξιολογητή να ανατεθούν κείμενα βάση της εξειδίκευσής του. Επειδή ένας ερευνητής έχει πάρα πολλούς τομείς ενδιαφέροντος είναι πολύ δύσκολο να καθοριστούν οι καταλληλότεροι άνθρωποι χειροκίνητα. Λόγω της απουσίας αυτοματοποιημένου συστήματος ενδεχομένως να προκύπτουν και προβλήματα μεροληψίας. Επίσης, για τα τμήματα ανθρωπίνων πόρων των εταιρών η διαδικασία εύρεσης κατάλληλων υπαλλήλων είναι εξίσου χρονοβόρα και δεν εγγυάται το καλύτερο δυνατό αποτέλεσμα.

Το σύστημα λοιπόν είναι ωφέλιμο για τους παρακάτω

- Διοργανωτές συνεδρίων για την εύρεση αξιολογητών.
- Τμήματα ανθρώπινων πόρων για την εύρεση νέων υπαλλήλων ή για την κινητικότητα εντός του οργανισμού

Χρήστες (Actors)

Οι χρήστες θα έχουν την δυνατότητα να εξάγουν αξιόπιστα αποτελέσματα σε σχέση με την χειροκίνητη διαδικασία η οποία είναι και χρονοβόρα, αφήνοντας επιπλέον μεγάλα περιθώρια λάθους.

Οι χρήστες του συστήματος στην προκειμένη περίπτωση είναι οι

- Διοργανωτές συνεδρίων
- Τμήματα ανθρώπινων πόρων εταιριών

Μετασχηματισμός (Transformation)

Κάθε σύστημα πραγματοποιεί μια μετατροπή δεδομένων. Αρχικά τα δεδομένα εισάγονται στο σύστημα και στην συνέχεια θα υποστούν κάποιου είδους επεξεργασία. Το τελικό αποτέλεσμα του συστήματος είναι τα μετασχηματισμένα δεδομένα.

Η είσοδος αποτελείται από λέξεις κλειδιά οι οποίες προέρχονται από τον χρήστη. Μαζί με ένα ευρετήριο που παρέχει όλη την απαιτούμενη πληροφορία το σύστημα θα

μετασχηματίσει τα δεδομένα και θα εξάγει λίστα εξειδικευμένων ατόμων που καλύπτουν πλήρως την είσοδο του χρήστη.

Εικόνα 7 Ιδεατή Είσοδος/Εξοδος

Κοσμοθεώρηση (Worldview)

Η κοσμοθεώρηση αποτελεί τον ευρύτερο αντίκτυπο που θα έχει το σύστημα ή αλλιώς η διαδικασία μετασχηματισμού που πραγματοποιεί. Το σύστημα αναλύεται για να βρεθούν οι αρνητικές και θετικές επιδράσεις στο σύνολο τους.

Η χρήση του συστήματος σκοπεύει στην εξάλειψη της σπατάλης χρόνου και κατ' επέκταση χρήματος για την διαδικασία εύρεσης ειδικών. Άμεση συνέπεια είναι η αύξηση της παραγωγικότητας για τους εμπλεκόμενους «Clients» αλλά και η βελτίωση της αξιοπιστίας των αποτελεσμάτων καθώς θα μπορούν να θεωρηθούν με ασφάλεια αμερόληπτα.

Ιδιοκτήτης (Owner)

Οι άνθρωποι που βρίσκονται στις κατάλληλες θέσεις ισχύος και έχουν την εξουσιοδότησης να αλλάξουν ή ακόμα να σταματήσουν την υλοποίηση του συστήματος χαρακτηρίζονται ως ιδιοκτήτες.

Κάθε οργανισμός ή εταιρία θα μπορούσε να χρησιμοποιεί το σύστημα και να το διαθέτει στους υπαλλήλους/συνεργάτες τους. Στην περίπτωση αυτή ιδιοκτήτης θεωρείται ο υπεύθυνος για την χρηματοδότηση του εκάστοτε τμήματος. Μια άλλη δυνατότητα είναι το σύστημα να δοθεί σε επιστημονικές και εκπαιδευτικές κοινότητες όπου ιδιοκτήτες θεωρούνται οι πρυτανικές αρχές.

Περιορισμοί περιβάλλοντος (Environmental constraints)

Οι περιορισμοί περιβάλλοντος, αποτελούν τους εξωτερικούς παράγοντες που μπορούν να επηρεάσουν την επιτυχή ολοκλήρωση του συστήματος. Αυτοί μπορεί να είναι κανονισμοί λειτουργίας, κοινωνικοί περιορισμοί ή μειωμένη χρηματοδότηση.

Για το εν λόγω σύστημα θα θεωρηθεί μόνο ένας τεχνικός περιορισμός. Οι πληροφορίες ή ειδικότερα το ευρετήριο που περιέχει όλους τους ειδικούς για τους οποίους γίνεται αναζήτηση, θεωρείται ότι είναι σταθερό και δεν αλλάζει.

Ορισμός

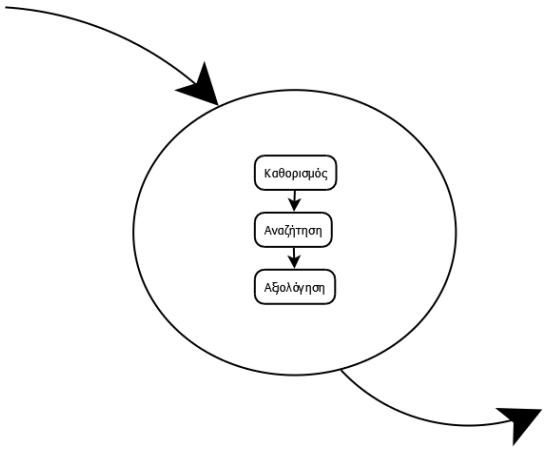
Σκοπός της παραπάνω διαδικασίας ήταν η σωστή κατασκευή του θεμελιακού ορισμού. Ο ορισμός αποτελεί τον ακρογωνιαίο λίθο της ανάλυσης ενός συστήματος και ο λογικός σχεδιασμός που αίπεται στηρίζεται σε αυτόν. Για το εν λόγω σύστημα ο θεμελιακός ορισμός έχει ως εξής :

Ένα σύστημα για εταιρίες/οργανισμούς με σκοπό την αυτοματοποιημένη εύρεση ειδικών η χρήση του οποίου θα προσφέρει μείωση χρόνου και αύξηση αξιοπιστίας, στην εργασία των διοργανωτών συνεδρίων ή στα τμήματα ανθρωπίνων πόρων, λαμβάνοντας ως είσοδο λέξεις κλειδιά και επιστρέφοντας λίστες ειδικών, δεδομένου ότι το ευρετήριο δεν αλλάζει, χρησιμοποιώντας την καταλληλότερη και αποτελεσματικότερη μέθοδο για την εύρεση ειδικών.

5.1.4 Ιδεατό διάγραμμα (Conceptual diagram)

Για την δημιουργία του ιδεατού διαγράμματος θα καθοριστούν οι ενέργειες που πρέπει να κάνουν οι χρήστες προκειμένου να επιτευχθεί ο μετασχηματισμός.

- Να καθοριστούν οι λέξεις κλειδιά ή αλλιώς οι τομείς ενδιαφέροντος.
- Να γίνει αναζήτηση στο ευρετήριο
- Να γίνει αξιολόγηση των αποτελεσμάτων.

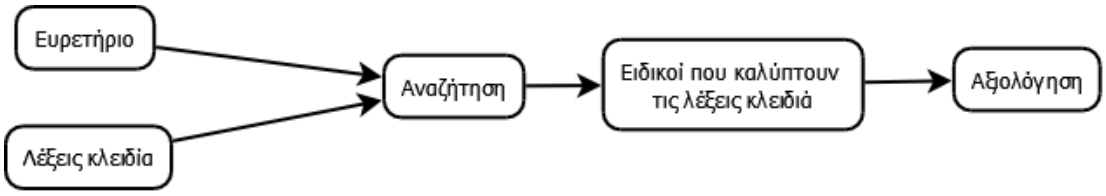


Εικόνα 8 Ιδεατό διάγραμμα

Σύγκριση μοντέλων με τον πραγματικό κόσμο

Τα μοντέλα που έχουν αναπτυχθεί μέχρι στιγμής δεν αναπαριστούν τον πραγματικό κόσμο , αλλά μία ιδεατή κατάσταση. Για να μετατραπεί το ιδεατό σε πραγματικό θα πρέπει να υπάρχουν προϋποθέσεις. Η κάλυψη των προϋποθέσεων αυτών θα δώσει την δυνατότητα στο σύστημα να λειτουργήσει.

Το βασικότερο προαπαιτούμενο του συστήματος είναι η ύπαρξη μίας τράπεζας πληροφορίων/ευρετήριο η οποία θα περιέχει όλα τα απαραίτητα δεδομένα με την χρήση των οποίων θα εξαχθούν τα αποτελέσματα.



Εικόνα 9 Πραγματική ροή

Καθορισμός των αλλαγών που είναι θεμιτές και εφικτές

Λόγω της μικρής έκτασης του συστήματος όλες οι αλλαγές που πρέπει να γίνουν σε είδη υπάρχοντα συστήματα είναι τεχνικά εφικτές και απουσία πραγματικής εταιρίας/ οργανισμού, στα πλαίσια της διπλωματικής, θεωρείται ότι είναι και θεμιτές.

Η κύρια αλλαγή επικεντρώνεται στην δημιουργία συστήματος από τρίτα μέρη το οποίο θα παρέχει συνεχώς μια πλήρως ενημερωμένη τράπεζα πληροφοριών .

Ενέργειες για να βελτιωθεί η υπάρχουσα κατάσταση

Τη στιγμή που η παρούσα ανάλυση δεν αντιπροσωπεύει τις απαιτήσεις μιας πραγματικής εταιρίας/οργανισμού θα θεωρηθεί, για τους σκοπούς της διπλωματικής, ότι οι ενέργειες που πρέπει να γίνουν για την βελτίωση της υπάρχουσας κατάστασης, είναι η δημιουργία τράπεζας πληροφοριών και στην συνέχεια η υλοποίηση του ίδιου του συστήματος.

5.2 Λογικός σχεδιασμός

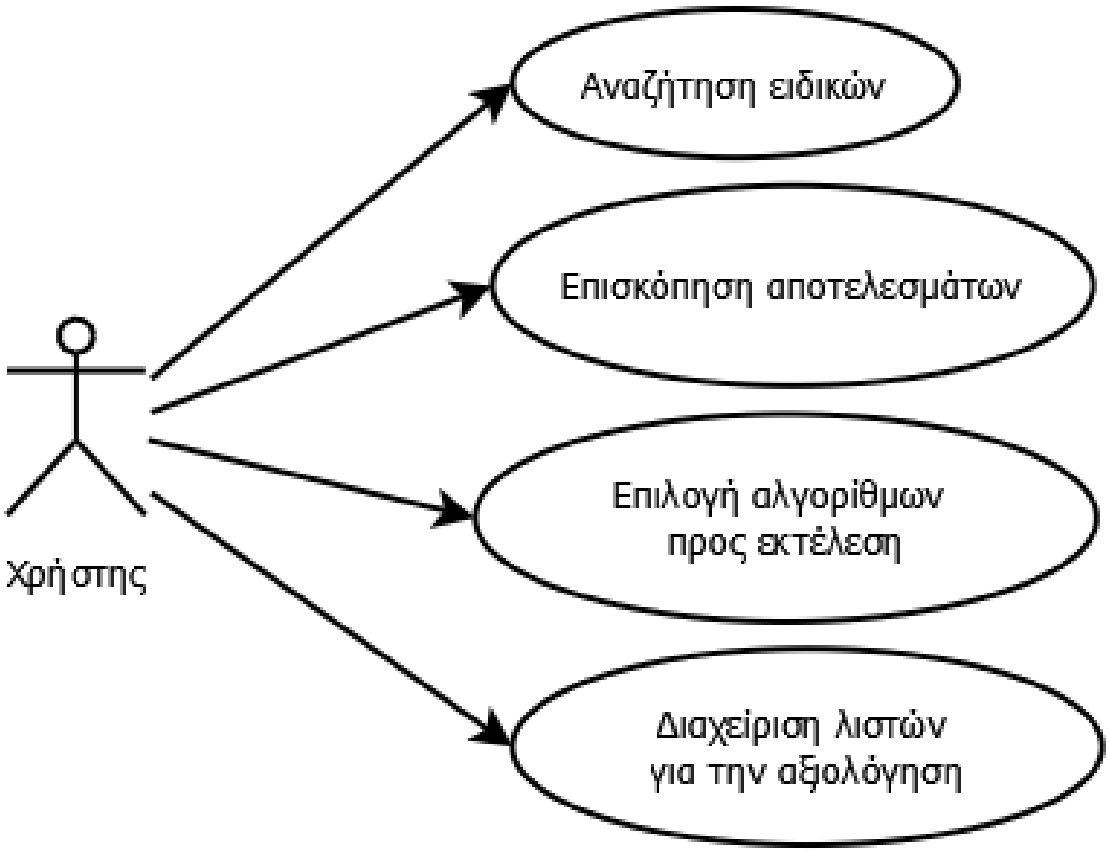
Μια από τις πιο ευπρόσδεκτες εξελίξεις στο πεδίο του αντικειμενοστραφούς σχεδιασμού και ανάλυσης ήλθε όταν τρεις αξιόλογοι ερευνητές, οι Booch, Jacobson και Rumbaugh, ένωσαν τις δυνάμεις τους και δημιούργησαν μια μοναδική κοινή γραφική γλώσσα μοντελοποίησης [29], την «ενοποιημένη γλώσσα μοντελοποίησης» (Unified Modeling Language,UML) [30]. Η γλώσσα αυτή παρέχει μια πληθώρα διάφορων διαγραμμάτων τα οποία περιγράφουν τα βασικά σχεδιαστικά κομμάτια οποιουδήποτε συστήματος.

Για τον λογικό σχεδιασμό χρησιμοποιήθηκε ένα υποσύνολο από τα διαγράμματα UML τα οποία παρουσιάζονται αναλυτικά στις υποενότητες που ακολουθούν.

5.2.1 Περιπτώσεις χρήσης (Use Case Diagram)

Το πρώτο και βασικότερο διάγραμμα της UML είναι το διάγραμμα περιπτώσεων χρήσης (Use Case diagram). Εδώ σημειώνονται οι τύποι των χρηστών του συστήματος και οι σημαντικότερες λειτουργείες που μπορούν να πραγματοποιήσουν.

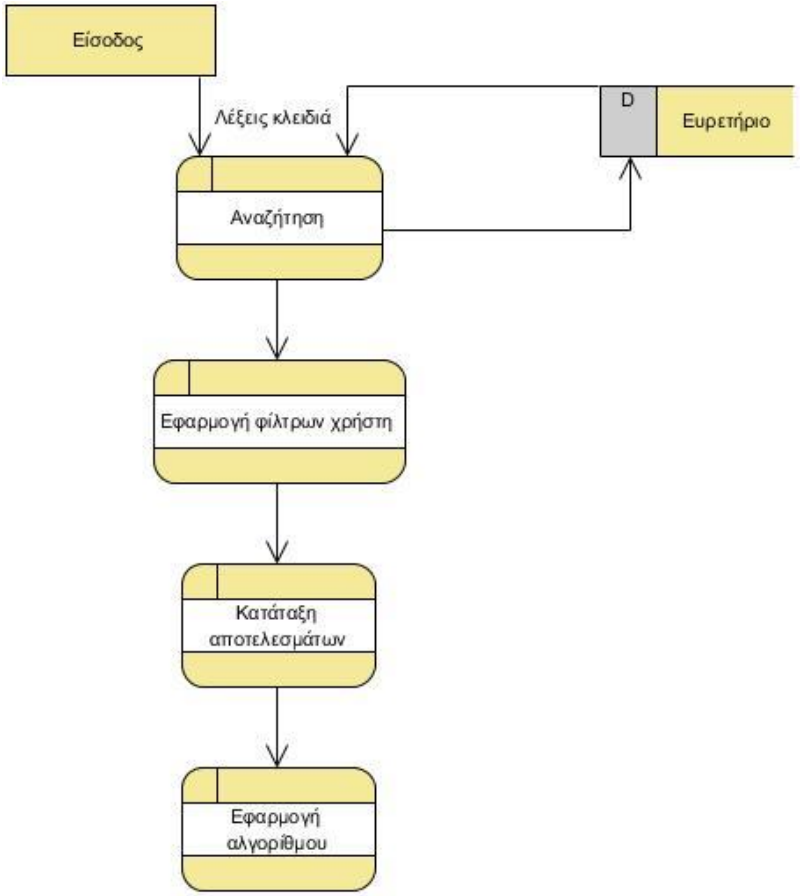
Οι περιπτώσεις χρήσης του συστήματος εύρεσης ειδικών περιορίζονται στην αναζήτηση ειδικών και μετέπειτα στην δυνατότητα αξιολόγησης των αποτελεσμάτων.



Εικόνα 10 Βασικές περιπτώσεις χρήσης

5.2.2 Διάγραμμα Διαδικασίας (Activity Diagram)

Τα διαγράμματα διαδικασιών προβάλουν την βασική ροή της επιχειρησιακής λογικής ενός συστήματος.



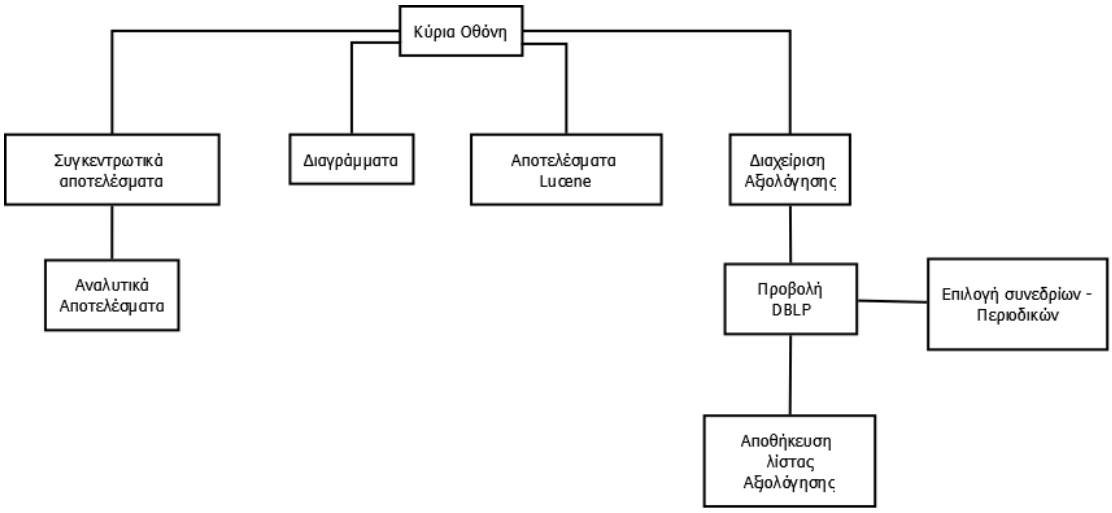
Εικόνα 11 Ροή διαδικασιών

Το πρώτο στάδιο είναι η είσοδος λέξεων κλειδιών από τον χρηστή. Οι λέξεις θα αντιπροσωπεύουν επιστημονικούς τομείς. Αμέσως μετά ξεκινά η αναζήτηση στο ευρετήριο για να βρεθούν όλοι οι σχετικοί ερευνητές. Το επιστρεφόμενο αποτέλεσμα θα αποτελείται από πολλαπλές λίστες ερευνητών κάθε μια από τις οποία αντιστοιχεί σε έναν επιστημονικό τομέα. Πάνω στο σύνολο αυτό, θα εφαρμοστούν ,προαιρετικά, φίλτρα καθορισμένα από τον χρήστη. Τα φίλτρα θα αφορούν περιορισμούς στην ποσότητα και την ποιότητα των ερευνητών. Για παράδειγμα να μην επιστρέφονται ερευνητές που δεν ξεπερνούν ένα προκαθορισμένο κατώφλι. Το αποτέλεσμα του φιλτραρίσματος θα ταξινομηθεί με βάση κάποιο βαθμολογικό κριτήριο. Προτεραιότητα δίνεται πάντα στους ερευνητές με το μεγαλύτερο εύρος εξειδίκευσης. Σε περίπτωση ισοβαθμίας θα ελέγχεται το επίπεδο εξειδίκευσης ανά τομέα. Το τελευταίο στάδιο της ροής είναι ο αλγόριθμος ο οποίος θα πρέπει να επιστρέψει το κατάλληλο σύνολο ειδικών που καλύπτει πλήρως όλους τους τομείς ενδιαφέροντος που έδωσε ο χρήστης ως λέξεις κλειδιά στην είσοδο.

5.2.3 Διάγραμμα οθονών

Η εφαρμογή διαθέτει πληθώρα λειτουργιών και κρίνεται απαραίτητος ο σωστός διαμοιρασμός αυτών στις αντίστοιχες οθόνες. Σημείο αναφοράς αποτελεί η κύρια οθόνη όπου πραγματοποιείται η είσοδος λέξεων κλειδιών από τον χρήστη και η αναζήτηση. Στην συνέχεια τα συγκεντρωτικά και αναλυτικά αποτελέσματα προβάλλονται στις δικές τους ξεχωριστές οθόνες. Μετά από την εκτέλεση των αλγορίθμων δημιουργούνται διαγράμματα για την καλυτερη παρουσίαση των μετρήσεων, τα οποία είναι προσβάσιμα από την αντίστοιχη οθόνη. Επόμενη διακλάδωση αποτελούν τα «Αποτελέσματα Lucene» όπου υπάρχει η δυνατότητα λεπτομερέστερης επισκόπησης των δεδομένων που εξάγονται από το Lucene κατά την διάρκεια της αναζήτησης.

Η «διαχείριση αξιολόγησης» δίνει την δυνατότητα για αποθήκευση, επεξεργασία και διαγραφή λιστών αξιολόγησης. Η οθόνη αυτή οδηγεί στην «Προβολή DBLP» όπου υπάρχει η δυνατότητα αναζήτησης πληροφορίας μέσα στην βιβλιοθήκη του DBLP με την χρήση φίλτρων. Δύο από τα φίλτρα αυτά είναι τα ονόματα συνεδρίων και περιοδικών. Τα ονόματα για τους σκοπούς της εφαρμογής είναι προκαθορισμένα και έτσι υπάρχει μια επιπλέον οθόνη για ευκολότερη καταχώρηση. Ο χρήστης δεν χρειάζεται να γράψει το όνομα του συνεδρίου/περιοδικού αλλά να το επιλέξει από μία έτοιμη προφορτωμένη λίστα. Τέλος η αποθήκευση της λίστας αξιολόγησης γίνεται σε ξεχωριστή οθόνη μαζί με ένα διακριτικό όνομα για να ξεχωρίζουν εύκολα μεταξύ τους.

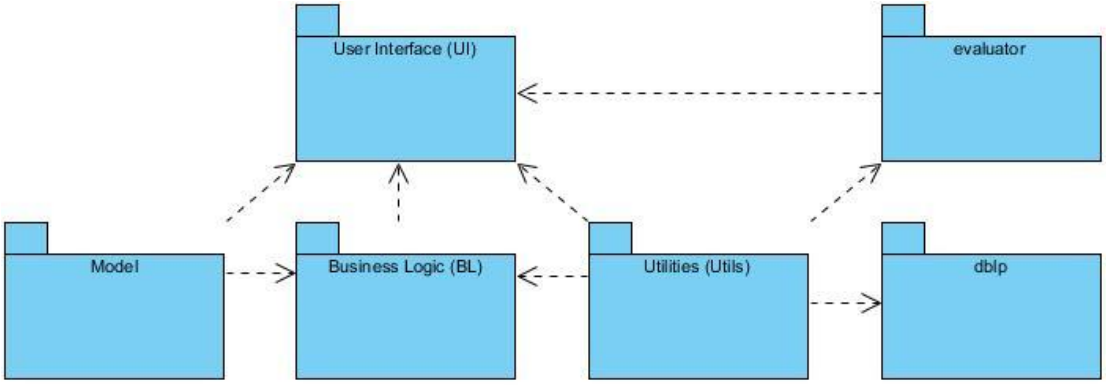


Εικόνα 12 Διάγραμμα Οθονών

5.2.4 Διαγράμματα Πακέτων-Κλάσεων (Package – Object Diagrams)

Τα διαγράμματα κλάσεων προβάλουν την πραγματική δομή ενός συστήματος. Αναπαριστούν τα αντικείμενα και τις σχέσεις μεταξύ τους καθώς και τις εκάστοτε ιδιότητές τους. Υπερσύνολο των προαναφερθέντων διαγραμμάτων αποτελούν τα διαγράμματα πακέτων. Τα πακέτα περιέχουν κλάσεις με κοινά χαρακτηριστικά και επιχειρησιακή λογική.

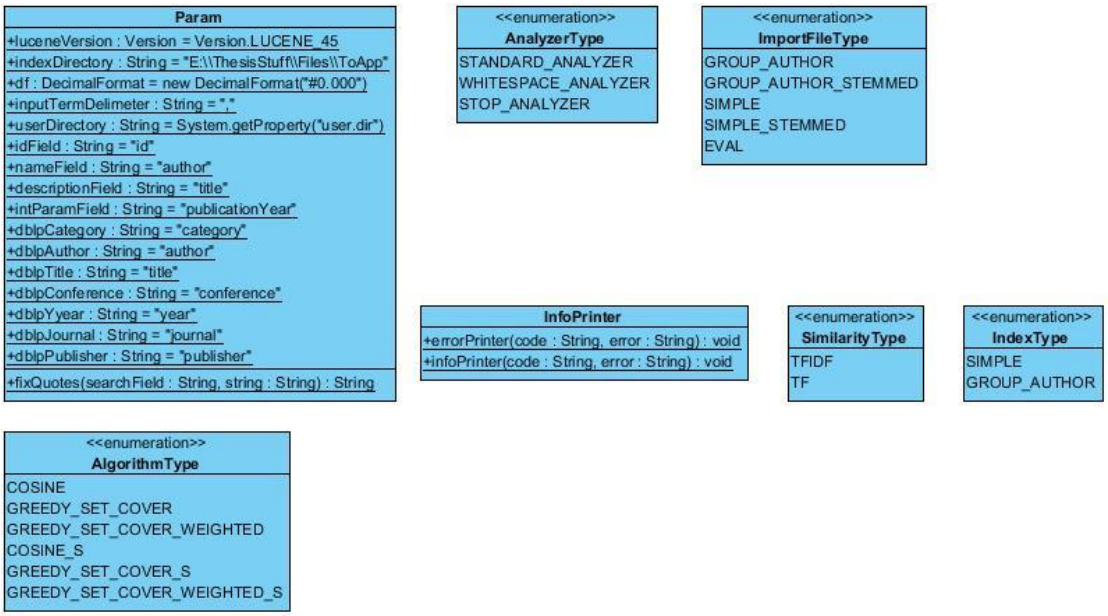
Τα πακέτα της εφαρμογής είναι χωρισμένα με βάση την λειτουργικότητα τους. Εξαίρεση αποτελεί το πακέτο που περιέχει την υλοποίηση της αξιολόγησης καθώς κρίθηκε σκόπιμο η υλοποίηση αυτή να είναι ανεξάρτητη από την υπόλοιπη εφαρμογή.



Εικόνα 13 Διάγραμμα πακέτων

Utils (Utilities)

Το πακέτο περιλαμβάνει συναρτήσεις και σταθερές οι οποίες χρησιμοποιούνται σε όλη την εφαρμογή.



Εικόνα 14 Κλάσεις στο πακέτο Utils

Param

Η κλάση περιέχει τις βασικές παραμέτρους του προγράμματος. Δεν αλλάζουν κατά την διάρκεια εκτέλεσής του. Περιλαμβάνονται πληροφορίες για τα δεδομένα που περιέχει το ευρετήριο καθώς και την τοποθεσία του στον σκληρό δίσκο του Η/Υ που λειτουργεί.

Types

Οι οντότητες με κατάληξη Type αποτελούν απαριθμήσεις (enumerations)

AlgorithmType

Οι ονομασίες των διαθέσιμων αλγόριθμων.

AnalyzerType

Οι ονομασίες των διαθέσιμων Analyzers. Οι analyzers αποτελούν ειδικές κλάσεις που αναλαμβάνουν την γλωσσική επεξεργασία κειμένων. Αποτελούν κομμάτι της βιβλιοθήκης Lucene η οποία θα αναλυθεί παρακάτω.

ImportFileType

Χρησιμοποιείται για τον καθορισμό του τύπου ευρετηρίου που θα δημιουργηθεί. Η κλάση αφορά το κομμάτι του κώδικα που δημιουργεί το ευρετήριο και για την διπλωματική θεωρείται εξωτερικό κομμάτι.

SimilarityType

Οι ονομασίες των τύπων βαθμολόγησης κειμένων κατά την διάρκεια αναζήτησης.

IndexType

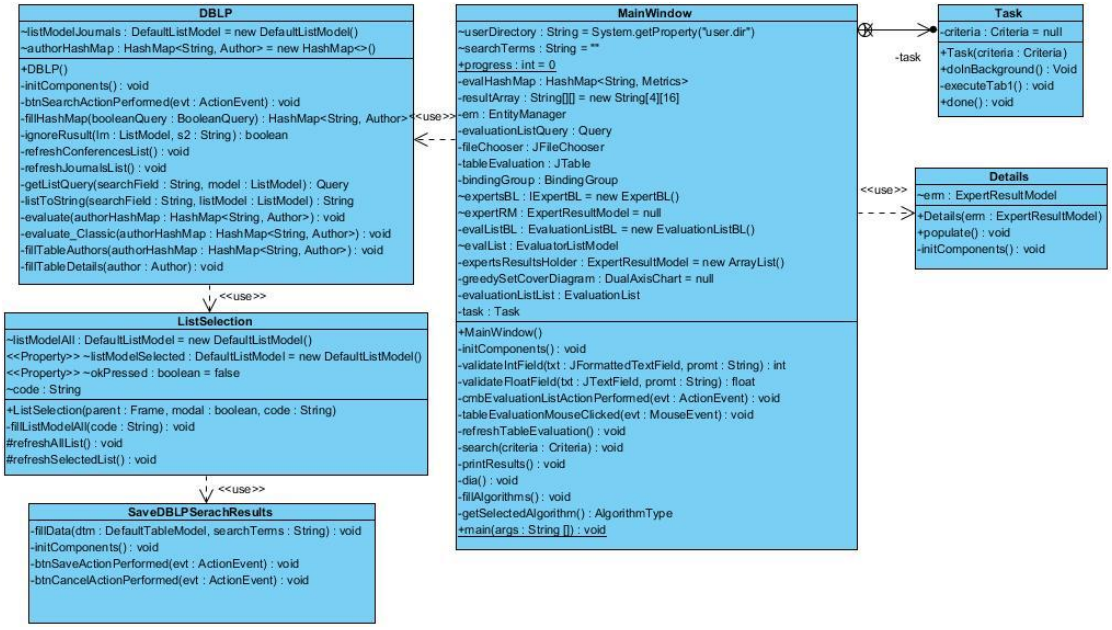
Οι ονομασίες των τύπων ευρετηρίων που χρησιμοποιούνται. Οι δύο διακριτοί τύποι είναι το GroupAuthor και Simple η δομή των οποίων παρουσιάστηκε σε προηγούμενο κεφάλαιο.

InfoPrinter

Βοηθητική κλάση για την εκτύπωση μνημάτων και σφαλμάτων στο στάδιο υλοποίησης της εφαρμογής.

Ui (User Interface)

Όλες οι οθόνες και γενικότερα οτιδήποτε αφορά την γραφική διεπαφή βρίσκεται σε αυτό το πακέτο. Το UI επικοινωνεί με την υπόλοιπη λειτουργικότητα χρησιμοποιώντας μόνο το πακέτο bl που θα αναλυθεί παρακάτω.



Εικόνα 15 Κλάσεις στο πακέτο Ui

MainWindow

Το κύριο παράθυρο της εφαρμογής δημιουργείται στην κλάση MainWindow. Καθώς η οθόνη περιλαμβάνει πολλαπλές καρτέλες αλλά και μία ασύγχρονη μπάρα προόδου ο κώδικάς της είναι αρκετά εκτενής.

Task

Εσωτερική κλάση του MainWindow με σκοπό την λειτουργία της ασύγχρονης μπάρας προόδου. Η μπάρα προβάλει την πρόοδο της κάθε αναζήτησης του χρήστη.

Details

Οθόνη για την εμφάνιση αποτελεσμάτων του κάθε σταδίου του αλγόριθμου που εκτελέστηκε. Η οθόνη είναι καθαρά πληροφοριακή και απλά φορτώνει λίστες που προέρχονται από το βασικό παράθυρο (MainWindow) και τις προβάλει στην οθόνη αναλυτικότερα.

DBLP

Οθόνη παρέχει δυνατότητα αναζήτησης σε ευρετήριο. Είναι συνώνυμη με την αντίστοιχη βιβλιοθήκη επιστημονικών δημοσιεύσεων η οποία θα αναλυθεί σε παρακάτω κεφάλαιο.

ListSelection

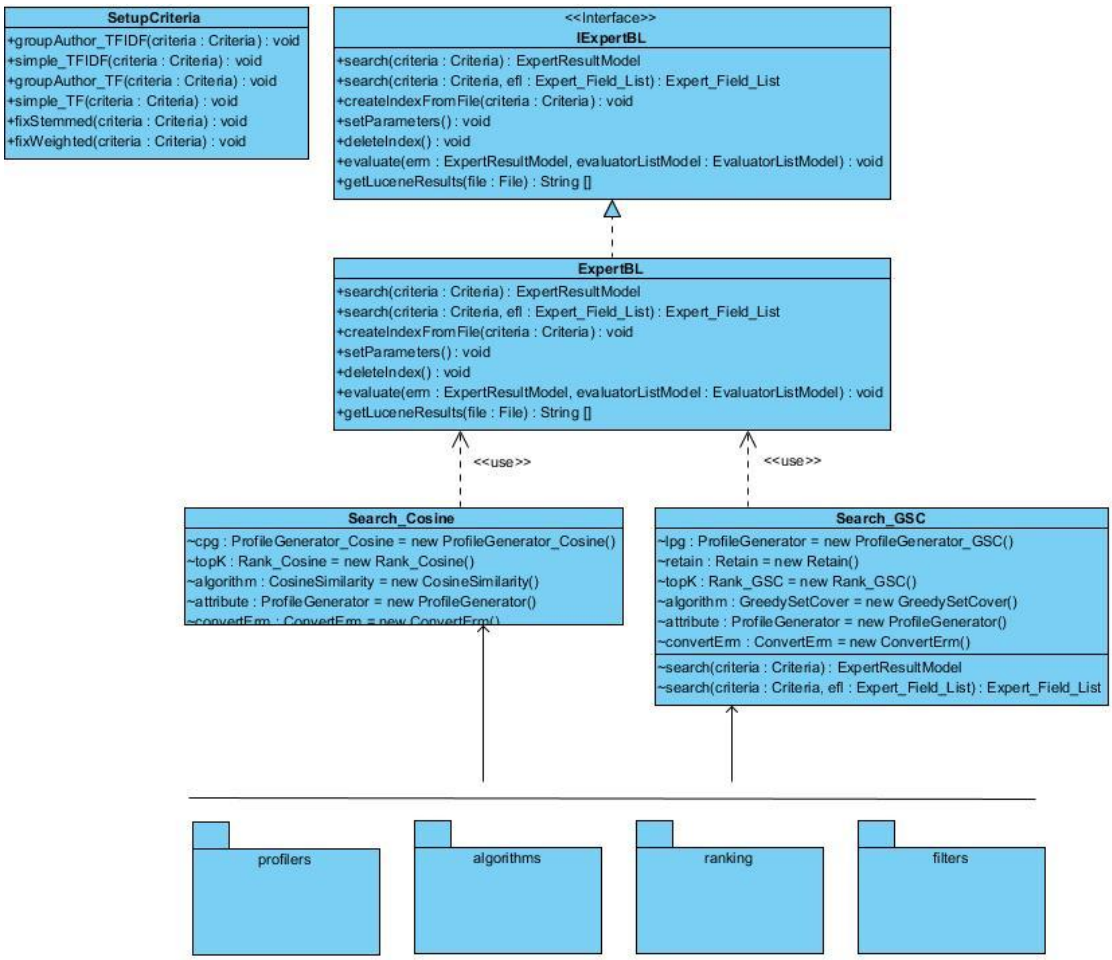
Η οθόνη παρέχει κώδικα για την εύκολη αναζήτηση μέσα σε μία λίστα. Κάθε φορά που πατιέται ένα πλήκτρο αυτόματα πραγματοποιείται αναζήτηση σε μία προ φορτωμένη λίστα και τα αποτελέσματα προβάλλονται και αποθηκεύονται σε μία δεύτερη λίστα.

SaveDBLPSearchResults

Η οθόνη χρησιμοποιείται για την αποθήκευση των αποτελεσμάτων αναζήτησης που πραγματοποιήθηκαν στην προηγούμενη οθόνη DBLP.

Bl (Business Logic)

Ο πυρήνας της λογικής του προγράμματος βρίσκεται εδώ. Για καλύτερη οργάνωση το πακέτο χωρίστηκε σε μερικά υπό-πακέτα έτσι ώστε να αντικατοπτρίζεται πιστότερα η αρχική ιδέα για την επίλυση του προβλήματος.



Εικόνα 16 Κλάσεις στο πακέτο BL

SetupCriteria

Κάθε αλγόριθμος αποτελείται από συγκεκριμένες παραμέτρους όπως ο τύπος του ευρετηρίου ή του analyzer. Όλες αυτές οι ρυθμίσεις βρίσκονται στην κλάση setupCriteria.

IExpertBL

Η επικοινωνία του πακέτου BL με το UI γίνεται μέσω αυτής της προγραμματιστικής διεπαφής.

ExpertBL

Η κλάση αναλαμβάνει την διαχείριση της εκτέλεσης των αλγορίθμων.

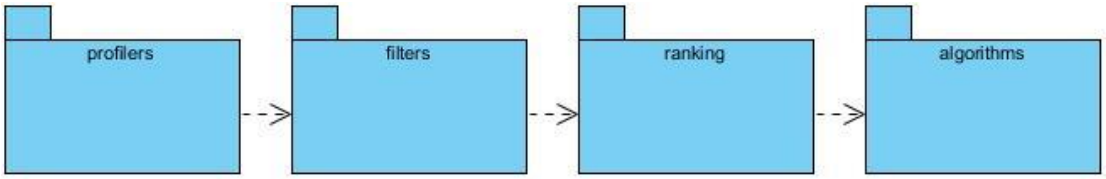
Search_Cosine

Διαχείριση του αλγόριθμου Cosine Similarity.

Search_GSC

Διαχείριση του αλγόριθμου Greedy Set Cover.

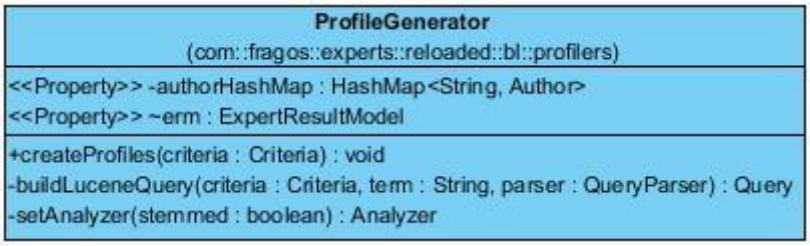
Η μέθοδος επίλυσης όπως αναφέρθηκε στο κεφάλαιο 4 αναλύεται σε πέντε διακριτά στάδια. Καθώς το πρώτο εξ' αυτών ,το ευρετήριο, αποτελεί δομή οργάνωσης δεδομένων και άρα βρίσκεται σε χαμηλότερο επίπεδο προγραμματισμού δε μπορεί να αναπαρασταθεί ως οντότητα. Όλα τα υπόλοιπα όμως στάδια έχουν υλοποιηθεί ως υποπακέτα.



Εικόνα 17 Υπο πακέτα BL

Profilers

Περιλαμβάνονται οντότητες υπεύθυνες για αναζήτηση σε ευρετήριο. Σε αυτές τις κλάσεις πραγματοποιείται η λεγόμενη δημιουργία προφίλ.



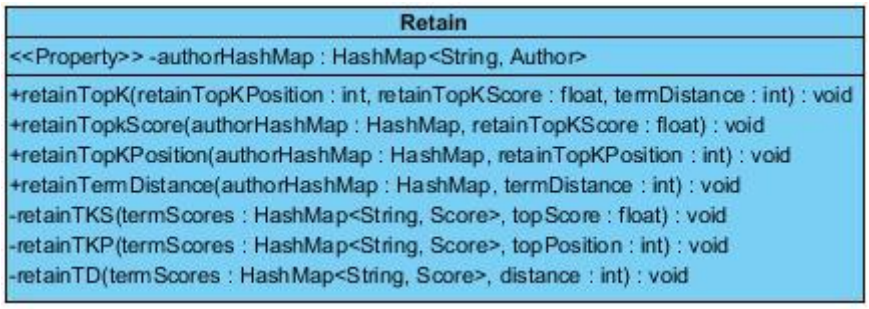
Εικόνα 18 Η κλάση για την δημιουργία προφίλ

ProfileGenerator

Δημιουργεί τα προφίλ με βάση τις λέξεις αναζήτησης που έδωσε ο χρήστης. Τα προφίλ αποτελούν έναν πίνακα κατακερματισμού (HashTable) [31]. Το κλειδί είναι κάθε φορά το όνομα του ειδικού. Αντίστοιχα το πεδίο της τιμής περιέχει μια οντότητα που συγκρατεί την βαθμολογία του ειδικού για κάθε όρο αναζήτησης.

Filters

Το πακέτο αυτό είναι υπεύθυνο για το φιλτράρισμα των προφίλ σύμφωνα με τις επιλογές του χρήστη.



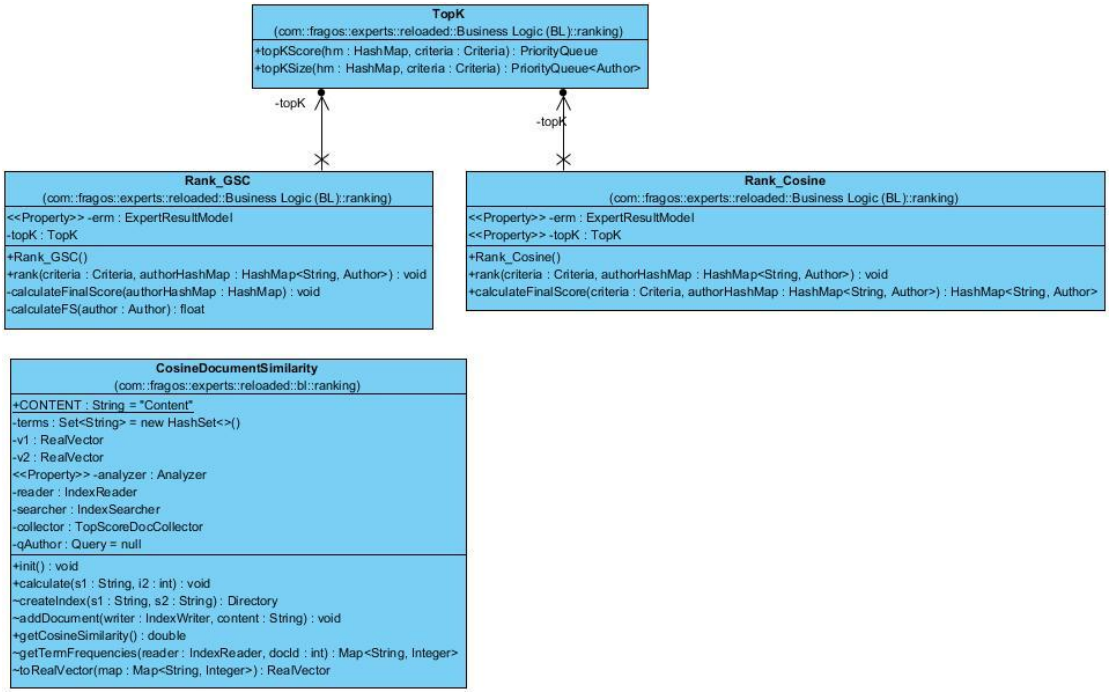
Εικόνα 19 Η κλάση για την εφαρμογή περιορισμών

Retain

Η κλάση υλοποιεί όλες τις συναρτήσεις που αφορούν το φιλτράρισμα των προφίλ που δημιουργήθηκαν βάσει των επιλογών του χρήστη. Οι συναρτήσεις δέχονται ως παράμετρο το hashtable των προφίλ μαζί με μία αριθμητική τιμή που χρησιμοποιείται ως κατώφλι. Ανάλογα με την περίπτωση θα διαγράψουν τις εγγραφές με μεγαλύτερη ή μικρότερη τιμή από το κατώφλι αυτό.

Ranking

Εδώ περιέχονται όλες οι μέθοδοι που κατατάσσουν τα προφίλ με βάση κάποιο κριτήριο. Τα δύο βασικότερα κριτήρια είναι το πλήθος των όρων στους οποίους κάποιος είναι ειδικός και το άθροισμα των βαθμολογιών τους.



Εικόνα 20 Κλάσεις για την κατάταξη αποτελεσμάτων

TopK

Ο διαχειριστής για την ταξινόμηση με βάση ένα βαθμολογικό κριτήριο.

Rank_GSC

Η κλάση υπολογίζει την τελική βαθμολογία κάθε ειδικού και στην συνέχεια κατατάσσει τα αποτελέσματα έτσι ώστε να είναι κατάλληλα για τον αλγόριθμο Greedy Set Cover. Εάν ο αλγόριθμος που εκτελείται δεν χρησιμοποιεί βάρη τότε η ταξινόμηση γίνεται πρώτα με το πλήθος των όρων που συγκεντρώνει ένα ειδικός και μετά την τελική του βαθμολογία. Αντίθετα εάν ο αλγόριθμος που εκτελείται χρησιμοποιεί βάρη τότε η ταξινόμηση γίνεται κατευθείαν με την τελική βαθμολογία.

Rank_Cosine

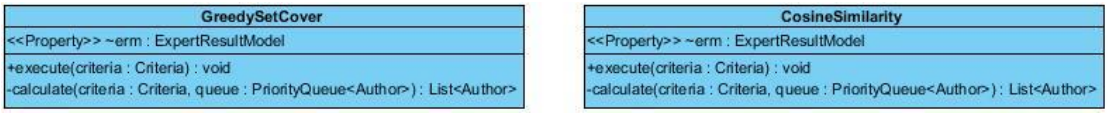
Η κλάση είναι υπεύθυνη για τον υπολογισμό των τελικών βαθμολογίων μετά τον οποίο θα ακολουθήσει ταξινόμηση των αποτελεσμάτων για τον αλγόριθμο Cosine Similarity.

CosineDocumentSimilarity

Η βαθμολόγηση κατά cosine similarity πραγματοποιείται σε αυτήν την κλάση. Η κλήση της συνάρτησης γίνεται από την προαναφερθείσα κλάση Rank_Cosine.

Algorithms

Οι αλγόριθμοι οι οποίοι εξάγουν το τελικό αποτέλεσμα. Στα πλαίσια της διπλωματικής υλοποιήθηκαν οι αλγόριθμοι Greedy Set Cover και Cosine Similarity.



Εικόνα 21 Οι Κλάσεις των αλγορίθμων

GreedySetCover

Η υλοποίηση του αλγορίθμου Greedy Set Cover μαζί με την έκδοση με βάρη πραγματοποιείται εδώ. Το επιστρεφόμενο αποτέλεσμα είναι ένα συγκεκριμένο μοντέλο λίστας το οποίο θα χρησιμοποιείται από όλους τους αλγορίθμους έτσι ώστε να καταλήγουν όλοι σε μία ενιαία μορφή αποτελέσματος.

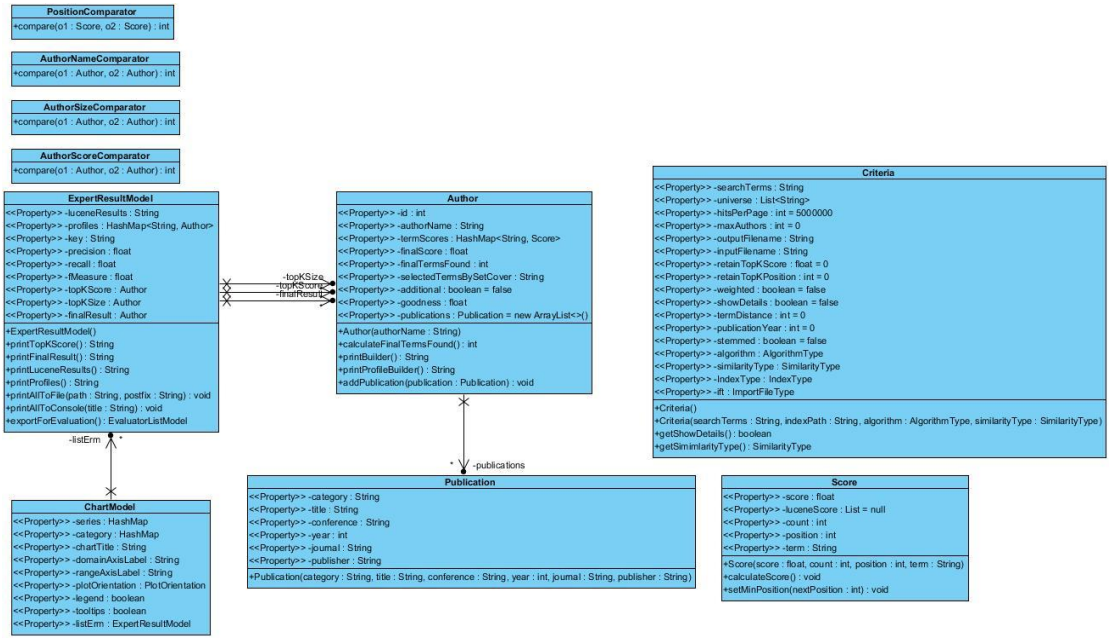
CosineSimilarity

Λόγω της φύσης του αλγόριθμου Cosine Similarity, το ουσιαστικότερο κομμάτι του είναι ο υπολογισμός του συνημίτονου. Καθώς πρέπει να υπάρχει ενιαία σχεδίαση έτσι ώστε να φιλοξενηθούν εύκολα και άλλοι αλγόριθμοι, αυτό το τελευταίο βήμα για τον cosine similarity δεν κάνει κανέναν υπολογισμό. Ο υπολογισμός του συνημίτονου έχει πραγματοποιηθεί στο πακέτο ranking σε προηγούμενο βήμα. Το μόνο που μένει να κάνει η κλάση CosineSimilarity είναι να επιστρέψει τα αποτελέσματα με την ενιαία μορφή αποτελεσμάτων που απαιτείται.

Η ενιαία μορφή αποτελεσμάτων αποτελεί την κλάση ExpertsResultModel του πακέτου Model. Η κλάση είναι ορατή και από το πακέτο της γραφικής διεπαφής UI και είναι η μοναδικά αποδεκτή μορφή αποτελεσμάτων. Κάθε μελλοντικός αλγόριθμος που υλοποιείται στο πακέτο Algorithms θα πρέπει να επιστρέφει ένα αντικείμενο της κλάσης αυτής.

Model

Στο πακέτο αυτό περιλαμβάνονται κλάσεις οι οποίες είναι ορατές σε όλη την εφαρμογή και παρέχουν τα βασικά μοντέλα όπως συγγραφέας (Author) και δημοσίευση (Publication).



Εικόνα 22 Τα χρησιμοποιούμενα μοντέλα

Comparators

Η ομάδα κλάσεων που είναι υπεύθυνη για τον καθορισμό των κριτηρίων σύγκρισης κατά την δημιουργία ουράς προτεραιότητας.

ExpertResultModel

Κλάση για την ενιαία διαχείριση των αποτελεσμάτων των αλγορίθμων. Συγκρατεί τα αποτελέσματα όλων των σταδίων επίλυσης από τα προφίλ (luceneResults) μέχρι το τελικό αποτέλεσμα (finalResult).

ChartModel

Κλάση για την απεικόνιση των αποτελεσμάτων σε διαγράμματα. Οι τιμές που απεικονίζονται είναι τα μετρικά precision, recall , goodness. Για την υλοποίηση του γραφικού κομματιού χρησιμοποιείται η έτοιμη βιβλιοθήκη διαγραμμάτων JFreeChart [32].

Author

Τον ακρογωνιαίο λίθο της εφαρμογής αποτελεί η οντότητα Author η οποία μοντελοποιεί τους υποψήφιους ειδικούς. Οι πληροφορίες που παρέχονται είναι το ονοματεπώνυμο του ειδικού, ένας μοναδικός αριθμός που χρησιμοποιείται για να ξεχωρίζουν τυχόν συνωνυμίες και τέλος οι σχετικές βαθμολογίες του για όλους τους όρους αναζήτησης.

Score

Η κλάση αποθηκεύει αναλυτικά και συγκεντρωτικά την βαθμολογία του κάθε συγγραφέα για έναν όρο αναζήτησης. Πιο συγκεκριμένα αποθηκεύεται η ονομασία του όρου, η βαθμολογία του και η θέση που έχει ο συγγραφέας στην κατάταξη για τον όρο αυτό.

Publication

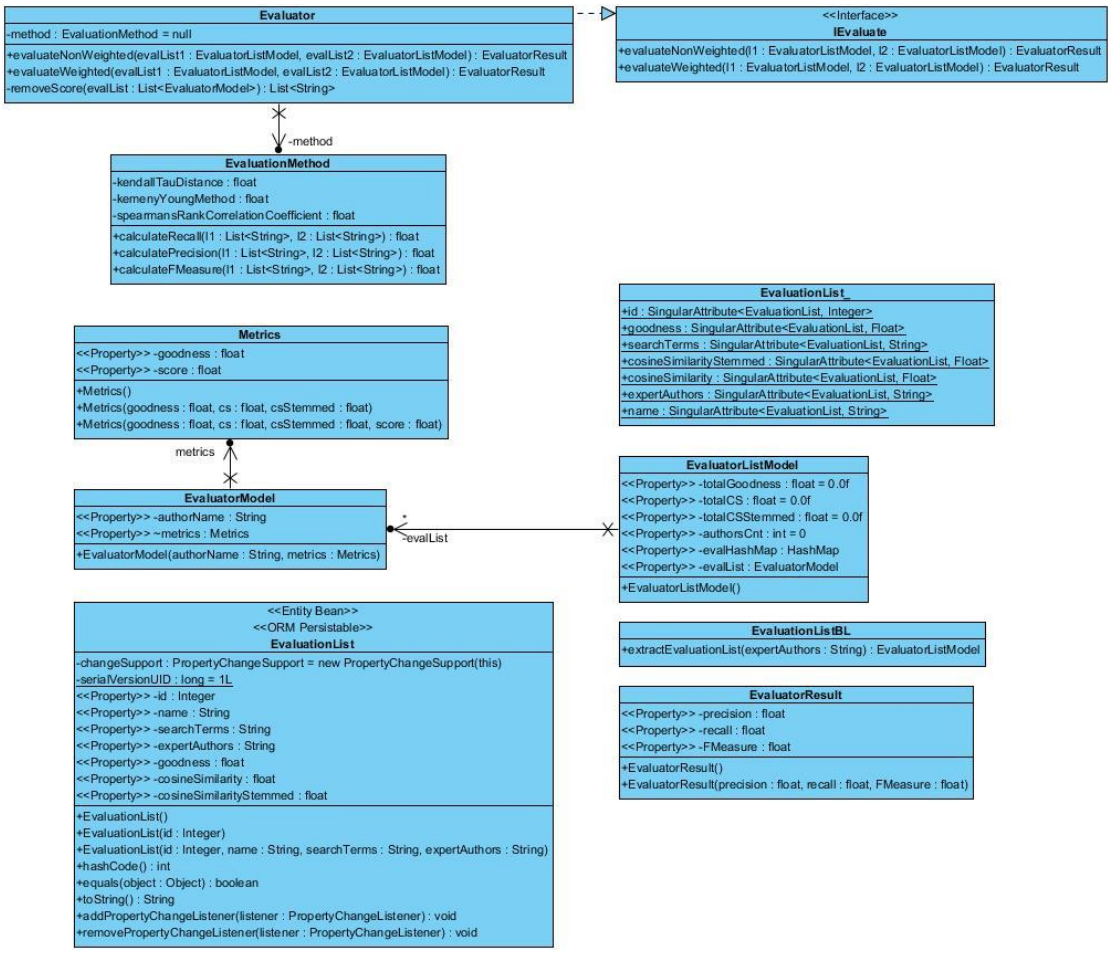
Η κλάση αυτή χρησιμοποιείται κατά την δημιουργία των ευρετηρίων. Μοντελοποιεί την οντότητα δημοσίευση σύμφωνα με τις προδιαγραφές του DBLP. Τα δεδομένα που συγκρατεί είναι ο τίτλος, το έτος και το συνέδριο ή το περιοδικό αντίστοιχα που έγινε η δημοσίευση.

Criteria

Η κλάση Criteria περιέχει όπως δηλώνει το όνομά της όλα τα κριτήρια και όλες οι επιλογές του χρήστη. Κάθε αντικείμενο της κλάσης αυτής διαπερνά όλα τα πακέτα της εφαρμογής καθώς μεταφέρει τις παραμέτρους που εισάγει ο χρήστης από γραφική διεπαφή στον τελικό υπολογισμό που κάνει ο αλγόριθμος.

Evaluator

Για την αξιολόγηση των αποτελεσμάτων των αλγορίθμων δημιουργήθηκε ένα ξεχωριστό πακέτο αποκλειστικά για τον σκοπό αυτό. Ο λόγος που το πακέτο θεωρείται εξωτερικό ως προς την εφαρμογή είναι για να υπάρχει η δυνατότητα επέκτασης και με διαφορετικό σύστημα αξιολόγησης εάν χρειαστεί.



Εικόνα 23 Οι Κλάσεις για την αξιολόγηση

IEvaluate

Διεπαφή για την επικοινωνία της αξιολόγησης με την υπόλοιπη εφαρμογή.

Evaluator

Η κεντρική διαχείριση της αξιολόγησης.

EvaluationMethod

Η κλάση περιλαμβάνει τις μεθόδους αξιολόγησης που έχουν υλοποιηθεί. Οι μέθοδοι αφορούν την αξιολόγηση δύο λιστών και το πόσο αυτές απέχουν μεταξύ τους. Για τις ανάγκες τις διπλωματικής έχουν υλοποιηθεί οι μετρικές precision, recall, F-Measure και υπάρχει η δυνατότητα επέκτασης και σε άλλες μεθόδους όπως kendall Tau Distance, Spearmans Distance [33].

Metrics

Η κλάση αφορά μετρικά που αξιολογούν μια λίστα στο σύνολο της χωρίς σύγκριση με μία δεύτερη. Στην προκειμένη περίπτωση χρησιμοποιείται μόνο το μετρικό goodness.

EvaluatorModel

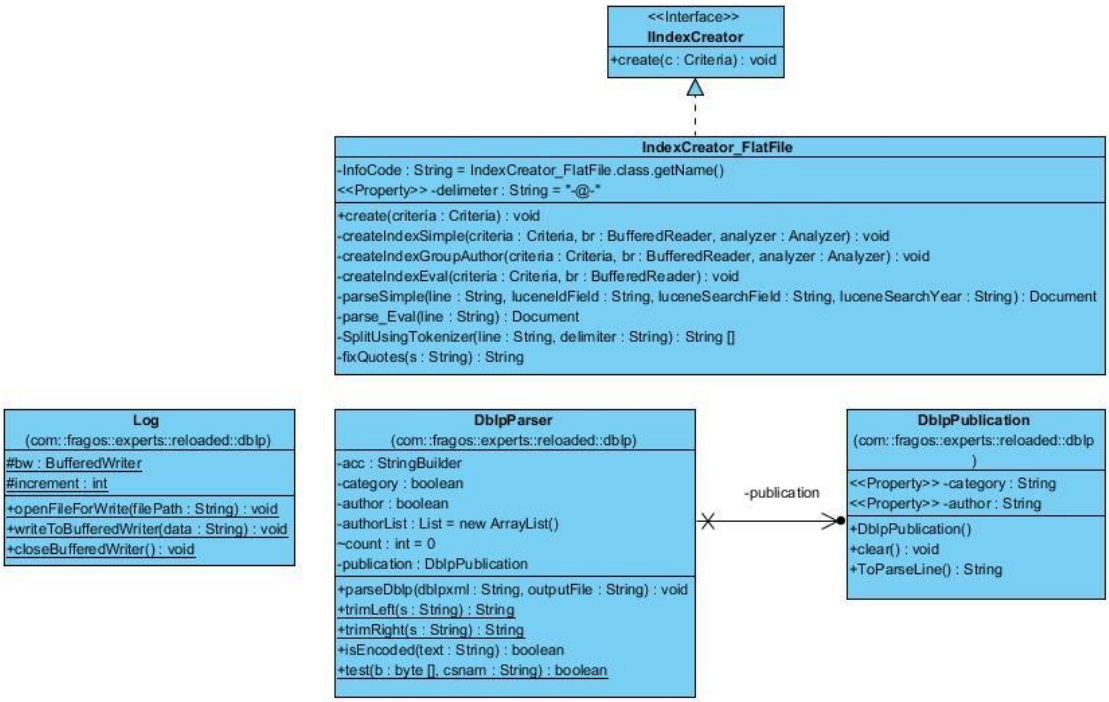
Μοντέλο για την αποθήκευση των δεδομένων αξιολόγησης ενός συγγραφέα.

EvaluatorListModel

Η εφαρμογή υποστηρίζει την αποθήκευση λιστών αξιολόγησης στον σκληρό δίσκο. Η κλάση EvaluatorListModel μοντελοποιεί τις λίστες αυτές έτσι ώστε να μπορούν να χρησιμοποιηθούν από το υπόλοιπο της εφαρμογής.

DBPL

Το κομμάτι του πακέτου DBLP αφορά σε κλάσεις οι οποίες μετατρέπουν το αρχικό DBLP.xml σε μορφή αντιληπτή από την εφαρμογή. Η μορφή στην οποία καταλήγει το DBLP.xml είναι ένα ευρετήριο Lucene. Λόγω της πολυπλοκότητας της μετατροπής αυτής πραγματοποιήθηκαν και εξωτερικές διεργασίες εκτός του κώδικα στο πακέτο DBLP.



Εικόνα 24 Κλάσεις για την μετατροπή του DBLP xml

IIndexCreator

Διεπαφή για την δημιουργία ευρετηρίου.

IndexCreator_FlatFile

Δημιουργία ευρετηρίου Lucene με βάση αρχείο κειμένου. Τα αρχείο DBLP.xml που περιέχει όλα τα δεδομένα έχει μετατραπεί σε απλό αρχείο κειμένου με διαδικασία εξωτερική της εφαρμογής.

DblpParser

Ο μετατροπέας του DBLP.xml σε αρχείο απλού κειμένου.

DblpPublication

Κλάση η οποία αναπαριστά την οντότητα της δημοσίευσης σύμφωνα με τις προδιαγραφές του DBLP.

Log

Βοηθητική κλάση υπεύθυνη για την αποθήκευση αλφαριθμητικών σε αρχείο κειμένου.

5.2.5 Διαλειτουργικότητα

Η διαλειτουργικότητα της εφαρμογής εντοπίζεται σε δύο σημεία. Το πρώτο είναι η μορφή του ευρετηρίου σε περίπτωση που υπάρχει η απαίτηση να χρησιμοποιηθούν διαφορετικά δεδομένα από το DBLP. Δηλαδή ποια θα πρέπει να είναι η μορφή του ευρετηρίου που θα φτιάξει κάποιος έτσι ώστε να συνδέετε με την εφαρμογή. Το δεύτερο είναι η γραμμογράφηση του αρχείου για την αναλυτική προβολή των αποτελεσμάτων του Lucene. Η εφαρμογή έχει την δυνατότητα μετά από κάθε εκτέλεση του αλγορίθμου να εξάγει ένα αρχείο με τα αναλυτικά αποτελέσματα που προέκυψαν κατά το βήμα της δημιουργίας των προφίλ. Καθώς τα αποτελέσματα είναι πολλά και η εποπτεία τους δύσκολη, καθορίζονται κανόνες γραμμογράφησης έτσι ώστε το αρχείο να επανεισαχθεί στην ειδική καρτέλα μέσα στην κεντρική οθόνη.

Ευρετήριο

Η εφαρμογή μπορεί να λειτουργήσει με δύο τύπους ευρετηρίου.

Simple

Σε αυτόν τον τύπο ευρετηρίου, υπάρχουν πολλά ξεχωριστά documents ανά συγγραφέα.

Πεδίο	Όνομα	Επεξήγηση
nameField	Author	Το όνομα του συγγραφέα
descriptionField	Title	Το πεδίο πάνω στο οποίο γίνεται η αναζήτηση.(Για την εφαρμογή χρησιμοποιούνται οι τίτλοι των άρθρων των συγγραφέων)
intParamField	Publication Year	Έτος δημοσίευσης
idField	Id	Καθορίζει μονοσήμαντα τον συγγραφέα

GroupAuthor

Σε αυτόν τον τύπο ευρετηρίου, χρησιμοποιείται ένα μοναδικό κείμενο που περιγράφει τον συγγραφέα.

Πεδίο	Όνομα	Επεξήγηση
nameField	Author	Το όνομα του συγγραφέα
descriptionField	Title	Το πεδίο πάνω στο οποίο γίνεται η αναζήτηση.(Για την εφαρμογή χρησιμοποιούνται οι τίτλοι των άρθρων των συγγραφέων)
idField	Id	Καθορίζει μονοσήμαντα τον συγγραφέα

Αρχεία εξαγωγής αποτελεσμάτων

Τα αποτελέσματα των αναζητήσεων μπορούν να αποθηκευτούν σε αρχεία στον δίσκο. Για την ευκολότερη εποπτεία των αποτελεσμάτων το αρχείο χωρίζεται σε τομείς οι οποίοι μπορούν να συμπτυχθούν ή να αναπτυχθούν χρησιμοποιώντας το πρόγραμμα Notepad++ και θέτοντας την γλώσσα σε Java.

Παρατίθεται η μορφή του αρχείου:

```
Lucene Results

//{

        //{

        #

        @Όρος αναζήτησης

        Θέση Βαθμ. Συγγραφέας

        ------------------------------------------------------------

        //}

//}

========================================

Profiles( Transposed Lucene Results)

//{

Όνομα συγγραφέα Term: Όρος αναζήτησης, Score: x

//}

========================================

Top-K Results (Score First)

//{

Final score: x Όνομα συγγραφέα Term: Όρος αναζήτησης, Score: x

//}

========================================

Set Cover Results

//{

Όνομα Συγγραφέα

Final  x

Term: x,  x, Lucene Documents: x

//}
```

Παράδειγμα αρχείου

Lucene Results

//{

//{

#

@Systems for Data Management

Θέση Βαθμ. Συγγραφέας

1 3.0 Aviral Shrivastava

2 3.0 Sang Hyuk Son

3 3.0 Hans-Peter Kriegel

…

==========

Profiles (Transposed Lucene Results)

//{

Sung-Wook Park Term: multi-media, Score: 2.0

José Antonio Portilla-Figueras Term: search, Score: 8.0 Term: multi-media, Score: 3.0

Anjun Wang Term: multi-media, Score: 1.0

…

==========

Top-K Results (Score First)

//{

Final score: 140.0 Jean Vanderdonckt Term: virtualization, Score: 1.0 Term: XML, Score: 1.0 Term: multi-media, Score: 14.0 Term: User Interfaces, Score: 124.0

Final score: 134.0 Hans-Peter Kriegel Term: spatial data, Score: 14.0 Term: query processing, Score: 46.0 Term: information extraction, Score: 2.0 Term: search, Score: 42.0 Term: multi-media, Score: 20.0 Term: Systems for Data Management, Score: 3.0 Term: temporal data, Score: 6.0 Term: storage, Score: 1.0

…

//}

Set Cover Results

//{

Ling Liu

Final 66.0

Term: information extraction, 2.0, Lucene Documents: 1

Term: consistency, 1.0, Lucene Documents: 1

…

//}

5.3 Εξωτερικές Βιβλιοθήκες

Για την δημιουργία του προγράμματος χρειάστηκε η βοήθεια κάποιων επιπλέον βιβλιοθηκών λογισμικού.

5.3.1 Lucene

Η βιβλιοθήκη Lucene παρέχει την δυνατότητα για δημιουργία ευρετηρίου κειμένων και αναζήτηση μέσα σε αυτό [26]. Είναι υλοποιημένη εξολοκλήρου στην γλώσσα προγραμματισμού java και θεωρείται το κορυφαίο εργαλείο στον χώρο του. Στην εφαρμογή χρησιμοποιείται για την κατασκευή του ευρετηρίου πάνω στο οποίο θα γίνει η αναζήτηση για την εύρεση των ειδικών.

5.3.2 SQLite

Η sqlite αποτελεί μία αυτοδύναμη SQL βάση δεδομένων [34]. Δεν χρειάζεται καμία ρύθμιση για να λειτουργήσει και τρέχει τοπικά στο μηχάνημα που εγκαθίσταται χωρίς την ανάγκη για server. Επίσης παρέχει και την δυνατότητα εκτέλεσης transactions. Στην εφαρμογή χρησιμοποιείται για να αποθηκευτούν οι λίστες αξιολόγησης που δημιουργεί ο χρήστης.

5.3.3 jFreeCharts

Η βιβλιοθήκη jFreeCharts παρέχει μία ευρύτατη γκάμα συναρτήσεων για την δημιουργία γραφημάτων και είναι υλοποιημένη εξολοκλήρου στην γλώσσα προγραμματισμού java [32]. Στην εφαρμογή χρησιμοποιείται για την προβολή των μετρικών precision, recall, f measure στην αντίστοιχη καρτέλα της κεντρικής οθόνης.

5.3.4 DBLP

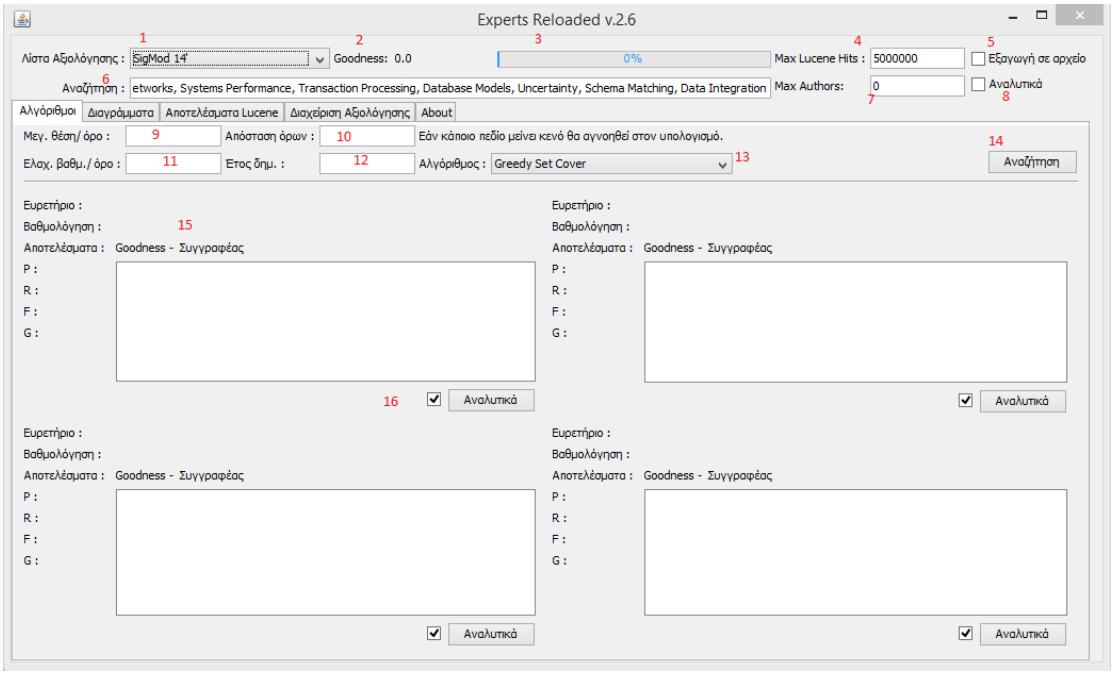
Η DBLP δεν αποτελεί προγραμματιστικό εργαλείο αλλά παρέχει όλη την απαραίτητη πληροφορία για να γίνουν οι δοκιμές των αλγορίθμων. Στην βιβλιοθήκη αυτή υπάρχουν όλες οι δημοσιεύσεις σχετικά με την επιστήμη των υπολογιστών [35] έως την ημερομηνία που αποκτήθηκε στις 6/10/2013. Η βιβλιοθήκη παρέχεται ως αρχείο xml και δε μπορεί να εισαχθεί κατευθείαν στο Lucene προς ευρετηριοποίηση. Για να μπορέσει να γίνει η διαδικασία αυτή γρήγορα το αρχείο xml θα μετατραπεί σε απλό αρχείο txt μέσω ειδικού κώδικα που φτιάχτηκε για το σκοπό αυτό και στην συνέχεια ακολουθεί η ευρετηριοποίηση του Lucene.

6 Υλοποίηση

Στο κεφάλαιο αυτό παρουσιάζονται όλες λειτουργίες και η γραφική διεπαφή της εφαρμογής. Για την ανάπτυξη της χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον (IDE) Netbeans και η γλώσσα προγραμματισμού Java. Επιπλέον αξιοποιήθηκαν οι εξωτερικές βιβλιοθήκες Lucene, SQLite, jFreeCharts η λειτουργεία των οποίων περιγράφεται στο κεφάλαιο 5.3 .

6.1 Βασική καρτέλα

Η βασική καρτέλα αποτελεί τον πυρήνα της εφαρμογής, εδώ πραγματοποιείται η αναζήτηση των ειδικών και εκτελούνται όλοι οι υλοποιημένοι αλγόριθμοι.



Εικόνα 25 Κεντρική οθόνη

6.1.1 Γραφική Διεπαφή

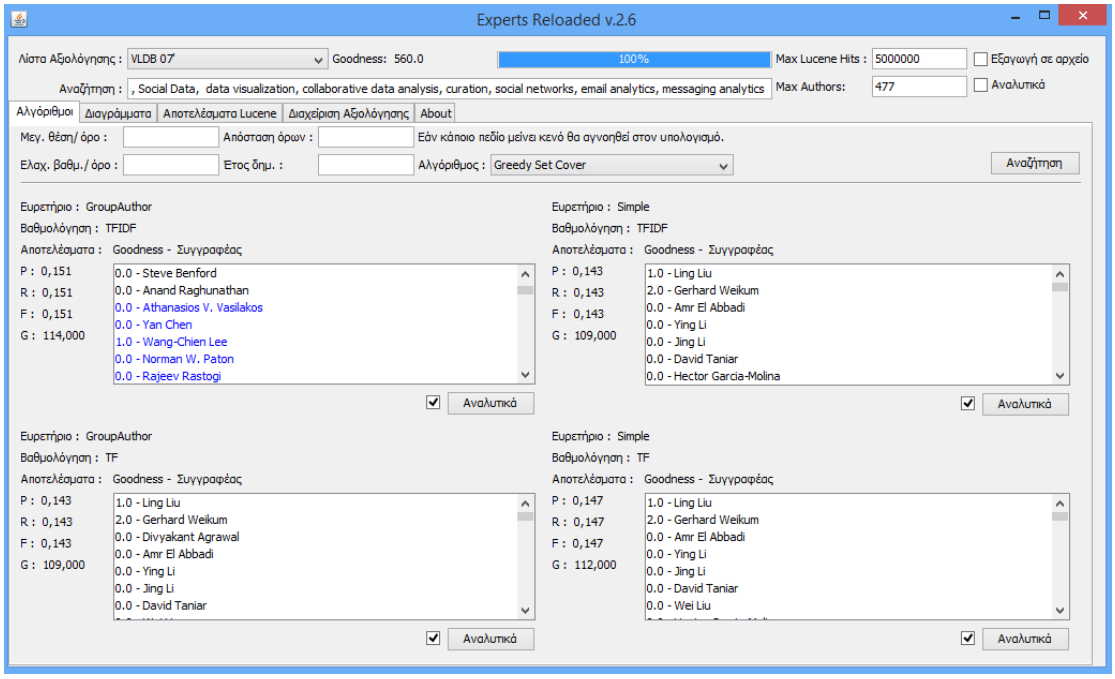
1. Λίστα αξιολόγησης: Επιλογή λίστας με βάση την οποία θα γίνει η σύγκριση των αποτελεσμάτων του αλγορίθμου.
2. Goodness: Συμπληρώνεται αυτόματα το συνολικό goodness της εκάστοτε επιλεγμένης λίστας αξιολόγησης.
3. Μπάρα προόδου: Προβάλει την πρόοδο της εκτέλεσης της αναζήτησης. Η ένδειξη 100% συμβολίζει την ολοκλήρωση όλων των επιλεγμένων μεθόδων.
4. Αποτελέσματα Lucene: Ο μέγιστος αριθμός των αποτελεσμάτων που θα επιστρέφει το κομμάτι της δημιουργίας προφίλ χρησιμοποιώντας το Lucene. Ο αριθμός αυτός θα πρέπει να είναι αρκετά μεγάλος έτσι ώστε να μπορέσει το Lucene να βαθμολογήσει πολλούς ειδικούς και στην συνέχεια να αναλάβει ο αλγόριθμος Set Cover για τα περεταίρω.
5. Εξαγωγή σε αρχείο: Επιλογή εάν ο χρήστης επιθυμεί εξαγωγή των αποτελεσμάτων σε αρχείο ή όχι.
6. Λέξεις κλειδιά: Εισαγωγή λέξεων κλειδιών προς αναζήτηση των ειδικών.

7. K authors: Μέγιστος αριθμός επιστρεφόμενων ειδικών (K). Στην περίπτωση που ο αριθμός αυτός είναι μικρότερος από το ελάχιστο πλήθος συγγραφέων(K') που καλύπτουν τις λέξεις κλειδιά τότε θα προβληθεί το σύνολο (K').
8. Αναλυτικά: Με την επιλογή αυτή αποθηκεύονται προσωρινά στην μνήμη τα αποτελέσματα από τα διάφορα στάδια του αλγορίθμου, όπως η δημιουργία προφίλ και η ταξινόμηση κατά μέγιστη βαθμολογία. Εάν η επιλογή συνδυαστεί με την εξαγωγή σε αρχείο τότε τα δεδομένα αυτά θα καταγραφούν και στον δίσκο.
9. Μέγιστη θέση ανά όρο: Η μέγιστη θέση στην κατάταξη των προφίλ την οποία πρέπει να έχει ένας συγγραφέας ανά όρο για να ληφθεί υπόψιν στον μετέπειτα υπολογισμό.
10. Απόσταση όρων: Η μέγιστη επιτρεπτή διαφορά βαθμολογίας μεταξύ δύο όρων του ίδιου συγγραφέα. Η τιμή μηδέν αγνοείται.
11. Ελάχιστη βαθμολογία ανά όρο: Η ελάχιστη βαθμολογία κατά την δημιουργία προφίλ την οποία πρέπει να έχει ένας συγγραφέας ανά όρο για να ληφθεί υπόψιν στον μετέπειτα υπολογισμό.
12. Έτος δημοσίευσης: Οποιαδήποτε δημοσίευση πριν από το έτος αυτό θα αγνοηθεί από την δημιουργία προφίλ.
13. Επιλογή αλγορίθμου
14. Αναζήτηση
15. Πλαίσιο αποτελεσμάτων: Καθώς στην παρούσα προσέγγιση δοκιμάστηκαν πολλές διαφορετικές μέθοδοι, υπάρχει η δυνατότητα απεικόνισης τεσσάρων μεθόδων ταυτόχρονα για έναν αλγόριθμο. Οι διαφοροποιήσεις μπορεί να είναι είτε στον τύπο του ευρετηρίου είτε στον τρόπο βαθμολόγησης.
    - Ευρετήριο: Ο τύπος του ευρετηρίου που χρησιμοποιήθηκε. Η επιλογή αυτή είναι προ εγκατεστημένη στην εφαρμογή και δεν αλλάζει από τον χρήστη.
    - Αποτελέσματα: Τα αποτελέσματα προβάλλονται σε ζευγάρια Goodness – Όνομα συγγραφέα. Το ελάχιστο σύνολο που καλύπτει τους όρους αναζήτησης είναι χρωματισμένο με μαύρο χρώμα , ενώ κάθε επιπλέον συγγραφέας με μπλε.
    - Precision
    - Recall
    - F Measure
    - Goodness
16. Επιλογέας ενεργοποίησης/απενεργοποίησης πλαισίου ελέγχου

6.1.2 Λειτουργικότητα

Αρχικά ο χρήστης μπορεί να διαλέξει μία λίστα αξιολόγησης με την οποία θα συγκριθούν τα αποτελέσματα της αναζήτησης. Η δημιουργία μιας τέτοιας λίστας θα περιγραφεί σε παρακάτω κεφάλαιο. Με την επιλογή της λίστας αυτομάτως συμπληρώνεται το πεδίο με τις λέξεις κλειδιά (6) αλλά και το πεδίο με τους Κ συγγραφείς (7). Το βήμα αυτό δεν είναι υποχρεωτικό για τον χρήστη. Στην περίπτωση που δεν έχει επιλεχθεί λίστα θα πρέπει να συμπληρωθούν χειροκίνητα οι όροι αναζήτησης (6) καθώς και το πεδίο των Κ συγγραφέων (7). Στην συνέχεια μπορεί να θέσει τιμές σε οποιοδήποτε από τα φίλτρα (9,11,11,12) και να διαλέξει τον αλγόριθμο (13) που θα εκτελεστεί. Το τελευταίο βήμα είναι να πατήσει το κουμπί «Αναζήτηση» (14) και τα αποτελέσματα θα εμφανιστούν στην οθόνη.

Στο παράδειγμα της παρακάτω εικόνας φαίνονται τα αποτελέσματα για την αναζήτηση με λίστα αξιολόγησης από το συνέδριο VLDB 07'.

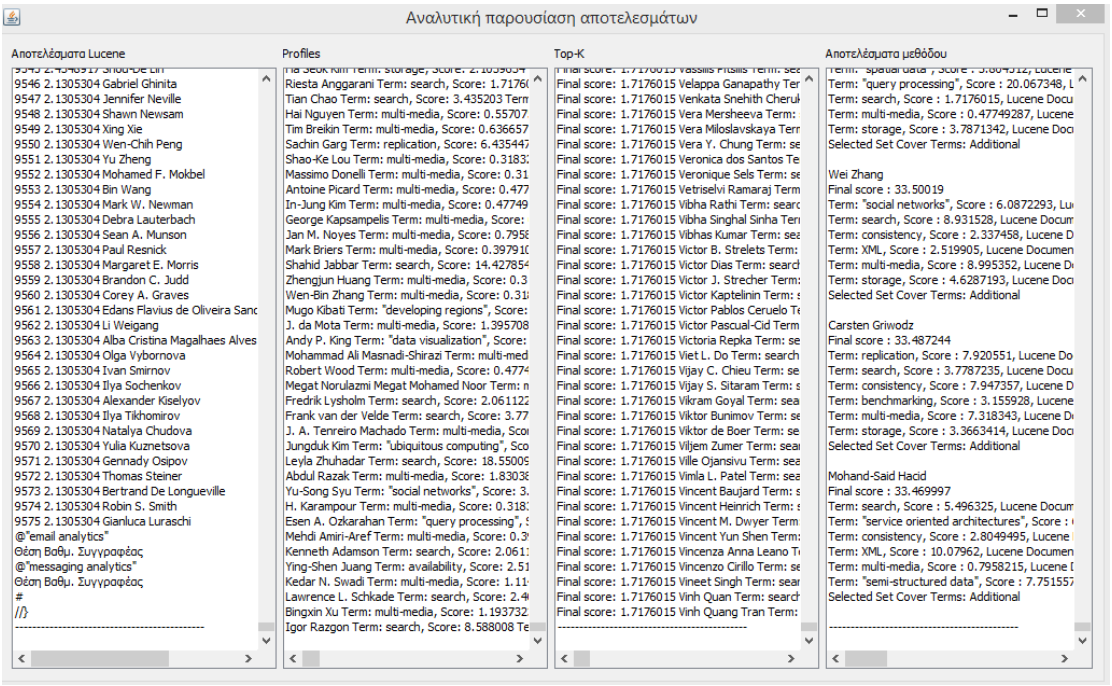


Εικόνα 26 Παράδειγμα αναζήτησης

Κάθε πλαίσιο αποτελέσματος (15) μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί με την χρήση του επιλογέα (16) δίπλα από το κουμπί «Αναλυτικά». Στην περίπτωση που ο επιλογέας δεν είναι μαρκαρισμένος δεν θα τρέξει ο αντίστοιχος αλγόριθμος.

6.1.3 Αναλυτικά αποτελέσματα

Στην περίπτωση που ο χρήστης επιθυμεί να δει αναλυτικά τα αποτελέσματα από κάθε στάδιο εκτέλεσης του αλγορίθμου θα πρέπει να μαρκάρει την επιλογή «Αναλυτικά» (8) και στην συνέχεια αφού ολοκληρωθεί η αναζήτηση, πατώντας το κουμπί «Αναλυτικά» που βρίσκεται σε κάθε πλαίσιο αποτελεσμάτων ανοίγει παρακάτω οθόνη.

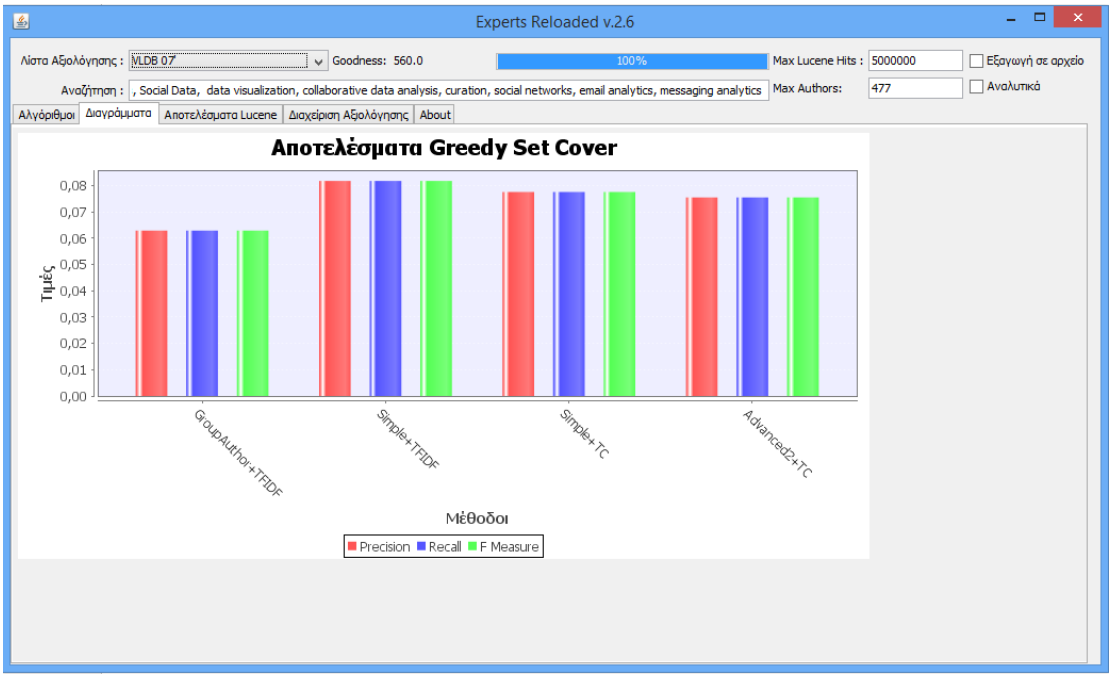


Εικόνα 27 Οθόνη αναλυτικής παρουσίασης αποτελεσμάτων

Η πρώτη στήλη προβάλει τα αποτελέσματα του Lucene όπως ακριβώς επιστρέφονται από αυτό. Για κάθε όρο αναζήτησης προβάλλεται η κατάταξη του συγγραφέα, η βαθμολογία του Lucene και το όνομά του. Στην δεύτερη στήλη τα αποτελέσματα του Lucene έχουν ομαδοποιηθεί ανά συγγραφέα και έτσι σε κάθε γραμμή υπάρχει το όνομα του συγγραφέα και η βαθμολογία του σχετικά με την κάθε λέξη κλειδί. Οι λέξεις κλειδιά αντιστοιχούν σε αυτές που εισήγαγε ο χρήστης κατά την αναζήτηση. Η στήλη αυτή ουσιαστικά προβάλει τα δημιουργημένα προφίλ. Στην επόμενη στήλη τα αποτελέσματα έχουν ταξινομηθεί με βάση την τελική βαθμολογία κάθε συγγραφέα η οποία προκύπτει από το άθροισμα των επιμέρους βαθμολογιών του. Η τελευταία στήλη προβάλει τα ζητούμενα αποτελέσματα σύμφωνα με τον επιλεγμένο αλγόριθμο. Κάτω από το όνομα του συγγραφέα υπάρχει η τελική βαθμολογία και οι όροι στους οποίους είναι ειδικός. Επίσης ανάλογα με το ευρετήριο που χρησιμοποιείται προβάλλεται και το πλήθος των σχετικών συγγραμμάτων. Τέλος, για την ομάδα των συγγραφέων που αποτελούν το ελάχιστο σύνολο εμφανίζονται για κάθε έναν οι όροι που επιλέχθηκαν από τον αλγόριθμο. Για τους υπόλοιπους συγγράφεις εμφανίζεται η σημείωση «Επιπλέον» για να σηματοδοτήσει το γεγονός ότι το ελάχιστο σύνολο ειδικών έχει ήδη βρεθεί και ο συγγραφέας δεν ανήκει σε αυτό.

6.2 Διαγράμματα

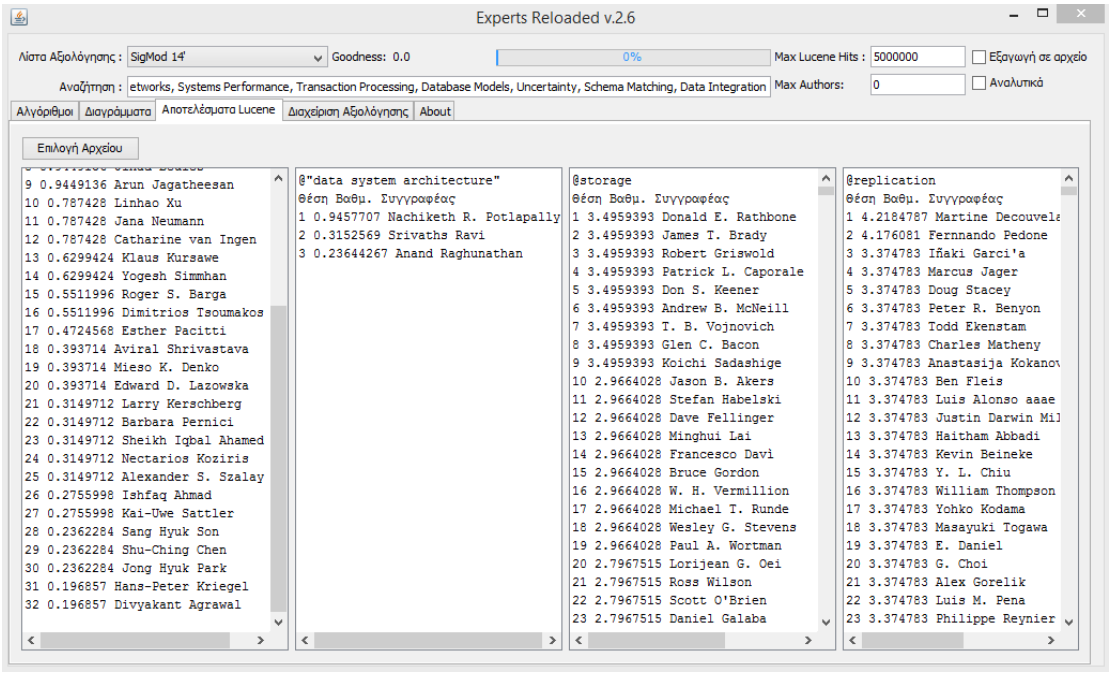
Η καρτέλα των διαγραμμάτων προβάλει τα στοιχεία από τα αποτελέσματα που δημιουργήθηκαν στην πιο πρόσφατη αναζήτηση του χρήστη. Οι τιμές στον κάθετο άξονα περιλαμβάνουν το Precision, Recall, FMeasure ενώ στον οριζόντιο βρίσκονται οι τέσσερις διαφοροποιήσεις του εκάστοτε επιλεγμένου αλγορίθμου. Δηλαδή οι συνδυασμοί για τα ευρετήρια (GroupAuthor-Simple) και οι τρόποι βαθμολόγησης του Lucene (TFIDF-TC).



Εικόνα 28 Οθόνη διαγραμμάτων

6.3 Αποτελέσματα Lucene

Για την καλύτερη εποπτεία και αποσφαλμάτωση της εφαρμογής δημιουργήθηκε η καρτέλα «Αποτελέσματα Lucene» στην οποία ο χρήστης μπορεί να εισάγει τα εξαγόμενα αρχεία της εφαρμογής. Στην οθόνη προβάλλονται τα αποτελέσματα που επιστρέφει η βιβλιοθήκη Lucene για τους τέσσερις πρώτους όρους αναζήτησης. Σκοπός της οθόνης είναι να βοηθήσει τον χρήστη να βγάλει συμπεράσματα παρέχοντας του την δυνατότητα να δει ταυτόχρονα στην οθόνη αναλυτικά πολλά αποτελέσματα.



Εικόνα 29 Οθόνη εισαγωγής αρχείων για επισκόπηση

6.4 Διαχείριση αξιολόγησης

Στην καρτέλα αυτή παρέχεται η δυνατότητα διαχείρισης των λιστών οι οποίες θα χρησιμοποιηθούν για την αξιολόγηση των αποτελεσμάτων του αλγορίθμου. Για τις ανάγκες της ανάκτησης των δεδομένων αυτών χρησιμοποιήθηκε βάση SQLite η οποία αποθηκεύεται ολόκληρη σε ένα αρχείο και διευκολύνει την φορητότητα της εφαρμογής.

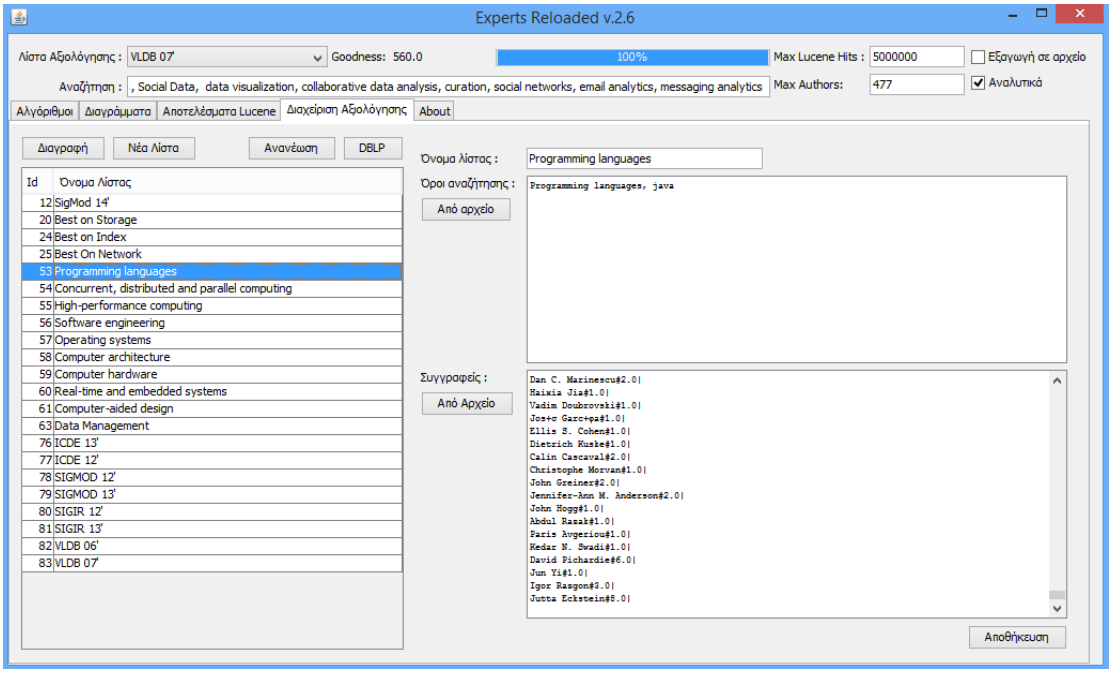
Ο χρήστης μπορεί να προσθέσει, διαγράψει και να επεξεργαστεί μία λίστα. Τα δεδομένα που συνοδεύουν μία λίστα αξιολόγησης είναι το όνομά της, οι όροι αναζήτησης και οι συγγραφείς οι οποίοι θεωρούνται ειδικοί για τους όρους αυτούς. Οι όροι θα πρέπει να χωρίζονται μεταξύ τους με κόμμα (,) για να θεωρούνται έγκυροι, ενώ οι συγγραφείς θα πρέπει να ακολουθούν την παρακάτω γραμμογράφηση.

Συγγραφέας Α # Βαθμολογία Α|

Συγγραφέας Β # Βαθμολογία Β|

…

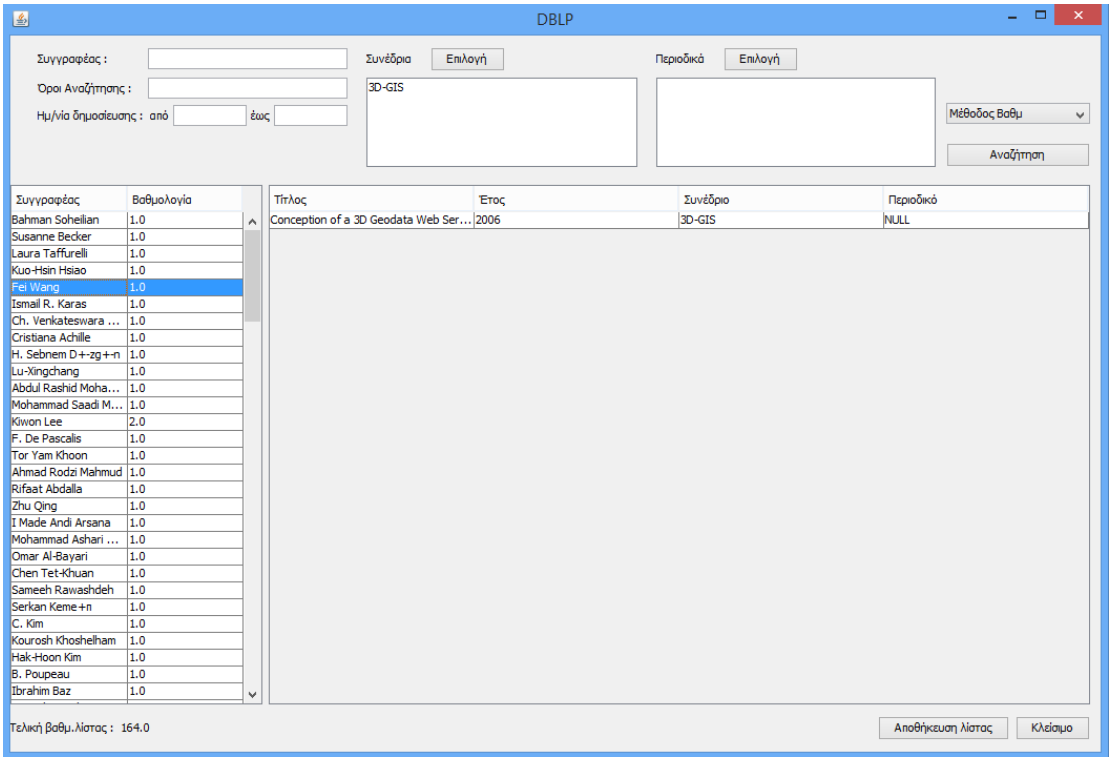
Πατώντας το κουμπί «DBLP» ανοίγει μια οθόνη από την οποία μπορεί να αναζητήσει δεδομένα για να φτιάξει τις λίστες του. Η λειτουργικότητα αυτής της οθόνης παρατίθεται στο επόμενο κεφάλαιο.



Εικόνα 30 Οθόνη διαχείρισης αξιολόγησης

6.5 Βιβλιοθήκη DBLP

Για την ταχύτερη αναζήτηση και επαλήθευση αποτελεσμάτων, αναπτύχθηκε μια μίνι εφαρμογή η οποία έχει ως στόχο την προβολή της βιβλιοθήκης DBLP το ευρετήριο της οποίας χρησιμοποιείται στην κύρια εφαρμογή.



Εικόνα 31 Οθόνη αναζήτησης στο DBLP

Ο χρήστης έχει την δυνατότητα να επιλέξει φίλτρα και να κάνει αναζήτηση στην βιβλιοθήκη. Με κάθε εμφάνιση αποτελεσμάτων αναγράφεται στην οθόνη η τελική βαθμολογία της λίστας η οποία δεν είναι άλλη από το άθροισμα του πλήθους των δημοσιεύσεων. Ως μελλοντική επέκταση έχει προστεθεί η δυνατότητα να αλλάζει η τελική βαθμολογία της λίστας, αλλά για της ανάγκες της διπλωματικής δε κρίθηκε απαραίτητη η δημιουργία επιπλέον μεθόδου.

6.5.1 Φίλτρα

Η αναζήτηση περιεχομένων στο DBLP γίνεται με βάση πέντε(5) κριτήρια αναζήτησης.

Συγγραφέας

Στο πεδίο των συγγραφέων ο χρήστης μπορεί να τοποθετήσει οποιοδήποτε αριθμό ονομάτων αρκεί να χωρίζονται με κόμμα. Επίσης θα πρέπει τα ονόματα να είναι πλήρη, για παράδειγμα Wang και όχι Wa%.

Όροι Αναζήτησης

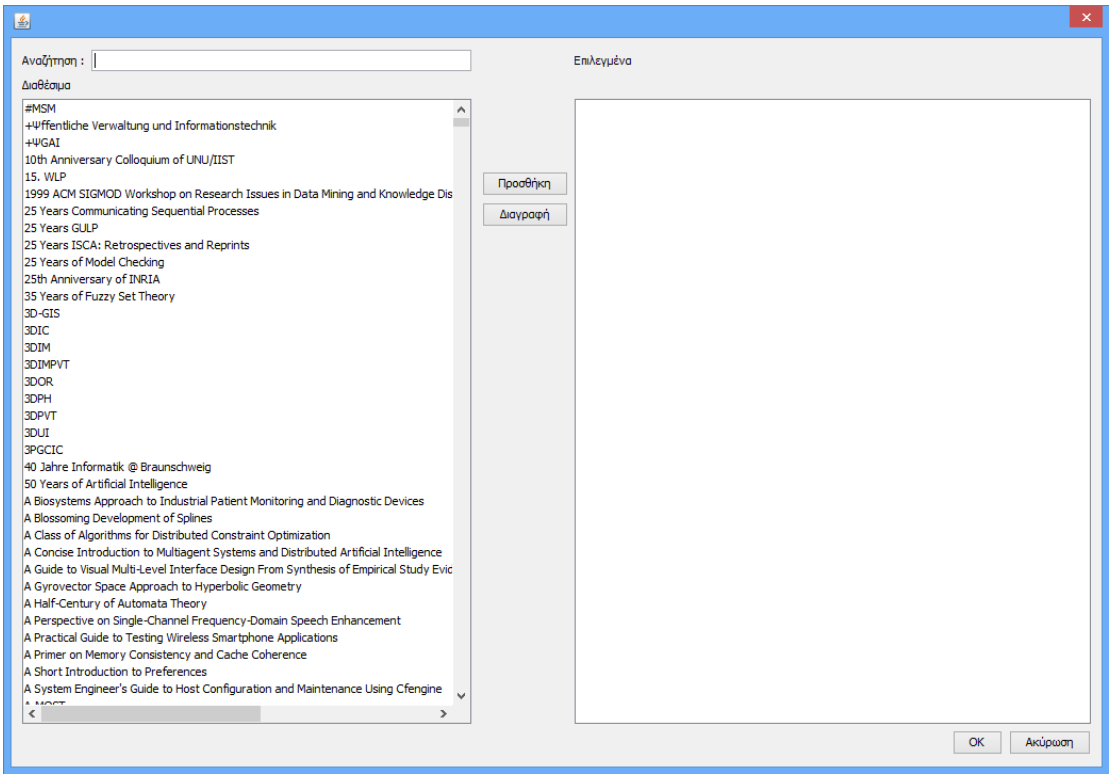
Οι όροι αναζήτησης αντιστοιχούν στους τίτλους των άρθρων των συγγραφέων όπως άλλωστε και στην κύρια εφαρμογή. Οι διαφορετικοί όροι θα πρέπει να χωρίζονται με κόμμα.

Ημ/νία δημοσίευσης

Η ημερομηνία δημοσίευσης, αναφέρεται στα άρθρα των συγγραφέων και μπορεί να λειτουργήσει με διάστημα από – έως. Για παράδειγμα η είσοδος 2012 έως 2014 θα συμπεριλάβει και τα άρθρα του 12’ μέχρι και αυτά του 14’.

Συνέδρια-Περιοδικά

Όλα τα συνέδρια και τα περιοδικά που περιέχει το DBLP έχουν φορτωθεί στην τοπική βάση δεδομένων έτσι ώστε ο χρήστης να μπορεί να διαλέξει ακριβώς αυτά που χρειάζεται. Εάν επιλεχθούν και συνέδρια αλλά και περιοδικά τότε η διαδικασία της αναζήτησης θα ψάξει να βρει συγγραφείς που καλύπτουν είτε το ένα είτε το άλλο κριτήριο.



Εικόνα 32 Βοηθητική οθόνη για την εύρεση συνεδρίων/περιοδικών

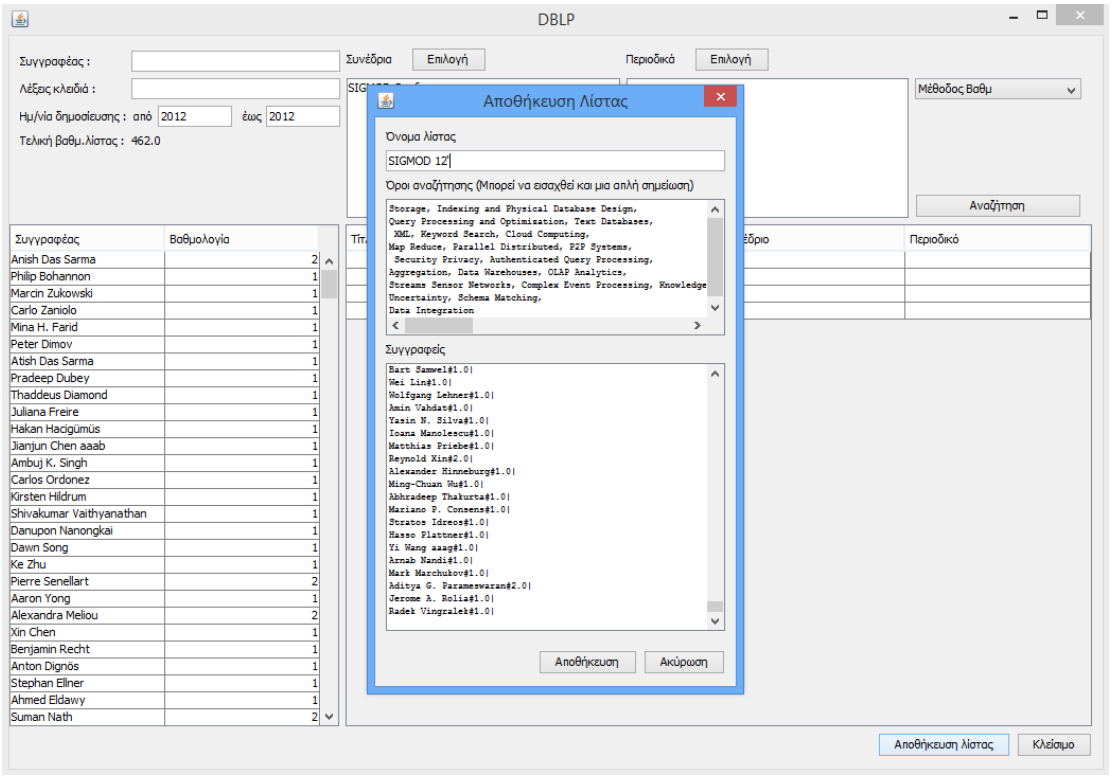
Για παράδειγμα έστω το συνέδριο 3DIC και το περιοδικό ACM SIGMOD Anthology. Η αναζήτηση θα επιστρέψει όλους τους συγγραφείς με άρθρα είτε στο 3DIC είτε στο ACM SIGMOD Anthology.

Αποτελέσματα – Βαθμολόγηση

Η περιοχή αποτελεσμάτων χωρίζεται σε δύο πίνακες, ο πρώτος προβάλει το όνομα του συγγραφέα μαζί την βαθμολογία του και ο δεύτερος προβάλει τα αντίστοιχα άρθρα του συγγραφέα από τα οποία προέκυψε η βαθμολογία αυτή.

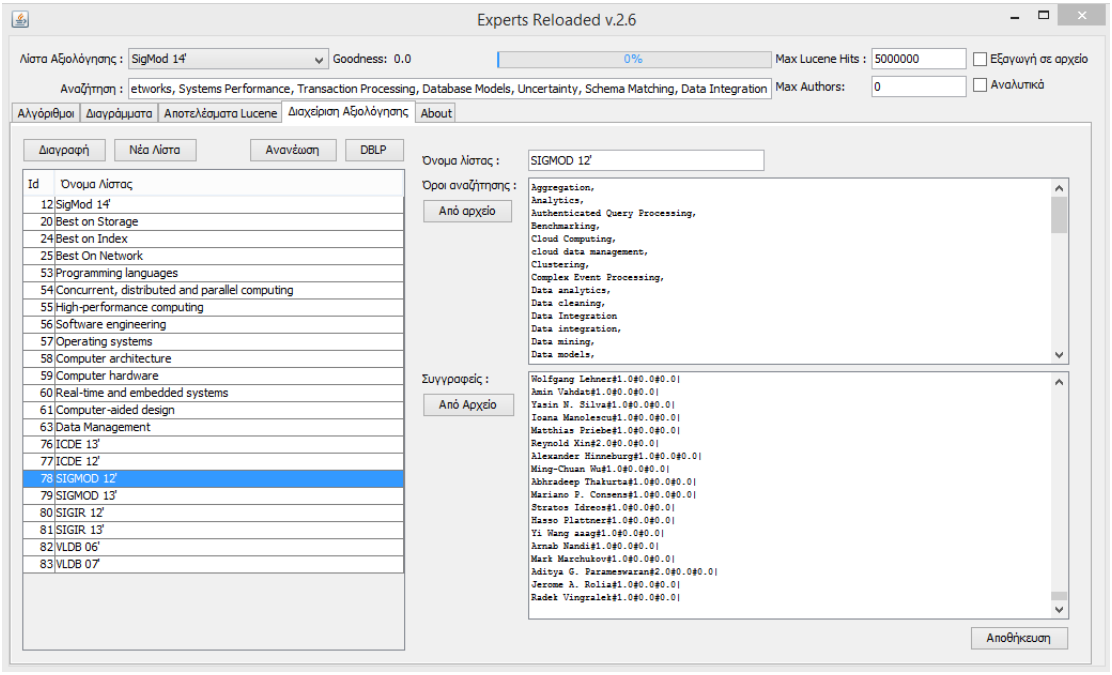
6.5.2 Αποθήκευση

Η αποθήκευση της λίστας από την οθόνη «DBLP» συνδέεται με την κύρια εφαρμογή και επιπλέον αποθηκεύεται αυτομάτως ως λίστα αξιολόγησης. Ο χρήστης έχει την δυνατότητα να εισάγει όνομα καθώς και σε ποιες λέξεις κλειδιά αντιστοιχεί η λίστα αυτή έτσι ώστε να μπορεί να γίνει η σύγκριση αποτελεσμάτων με την κύρια εφαρμογή.



Εικόνα 33 Οθόνη για την αποθήκευση λίστας αξιολόγησης

Μετά την αποθήκευση, η λίστα είναι διαθέσιμη στον πίνακα της αξιολόγησης. Σε περίπτωση που η λίστα δεν φαίνεται αμέσως, ο χρήστης θα πρέπει να πατήσει το κουμπί ανανέωση.



Εικόνα 34 Ενδεικτικά, αποθηκευμένες λίστες αξιολόγησης

7 Πειράματα

7.1 Διαδικασία

Κάθε διαδικασία αξιολόγησης απαιτεί τουλάχιστον ένα μέτρο σύγκρισης. Στην προκειμένη περίπτωση, χρησιμοποιείται το πλήθος των άρθρων που έχει γράψει κάποιος σε συναφή συνέδρια και περιοδικά (Goodness). Όσα περισσότερα άρθρα έχει κάποιος τόσο καλύτερος είναι. Το Goodness βασίζεται στην αξιοπιστία των συνεδρίων/περιοδικών για την ποιότητα των δημοσιεύσεων τους, εάν για παράδειγμα ένας συγγραφέας r_1 έχει τριάντα δημοσιεύσεις με θέμα τις βάσεις π.χ. στο SIGMOD και στο VLDB αυτό σημαίνει ότι είναι καλός στις βάσεις καθώς τα προαναφερθέντα συνέδρια είναι από τα κορυφαία στον τομέα τους και η βαθμολογία του είναι τριάντα.

Για τους σκοπούς της αξιολόγησης επιλέχθηκαν δύο ομάδες συνεδριών/περιοδικών (εφεξής, συνεδρίων, χάριν συντομίας).

Θεματολογία βάσεων δεδομένων (Ομάδα SIGMOD)

- SIGMOD Conference
- VLDB
- PVLDB
- ICDE
- EDBT
- VLDB J.
- ACM Trans. Database Syst. TODS
- IEEE Trans. Knowl. Data Eng. TKDE

Θεματολογία ανάκτησης πληροφορίας (Ομάδα ECIR)

- ECIR
- SIGIR
- CIKM
- Inf. Retr.
- Inf. Process. Manage.
- SIGIR Forum

Δείγμα από την ομάδα SIGMOD

Author	Goodness
Philip S. Yu	191
Divesh Srivastava	168
Jiawei Han	166
Hector Garcia-Molina	138
H. V. Jagadish	133
Surajit Chaudhuri	130
Beng Chin Ooi	129
Nick Koudas	109

Με τη χρήση της βάσης του DBLP, ανακτήθηκαν οι επίσημες λίστες (L') όλων των δημοσιεύσεων για τα συνέδρια:

- ICDE 13'
- SIGMOD 13'
- VLDB 13'
- ECIR 13'
- Ομάδα SIGMOD
- Ομάδα ECIR

Έστω L_{SIGMOD}= $\{\{r_1, G_1\},\ldots, \{r_j, G_j\}\}$ η συνολική λίστα της ομάδας SIGMOD και L'_{ICDE}= $\{r_1,\ldots, r_k\}$ η λίστα του συνεδρίου ICDE με j >k όπου r οι συγγραφείς, G το goodness. Το τελικό goodness προκύπτει από την αντιστοίχιση των συγγραφέων στο ICDE και υπολογίζεται από τον παρακάτω τύπο.

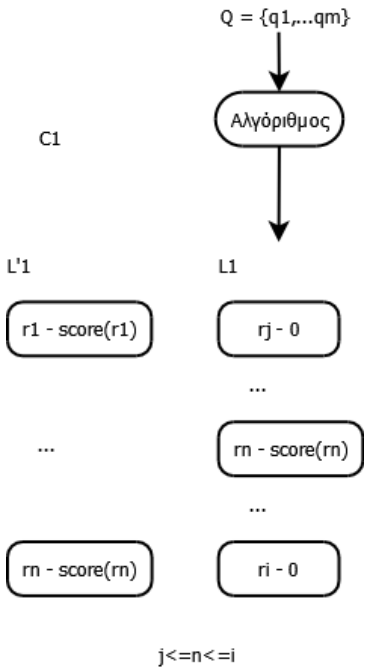
$$\text{Goodness}(L'_{ICDE}) = \sum_{i=1}^{k} G_i$$

Έχοντας πλέον τις αξιόπιστες λίστες ανά συνέδριο μαζί με μία ένδειξη για το πόσο καλός είναι ο κάθε συγγραφέας που περιέχουν θα πρέπει να γίνει σύγκριση με την έξοδο των μεθόδων που έχουν υλοποιηθεί στα πλαίσια της εργασίας. Καθώς οι μέθοδοί επίλυσης χρειάζονται τομείς ενδιαφέροντος ως είσοδο για να λειτουργήσουν και να παράξουν τις λίστες (L) αποτελεσμάτων, για κάθε ομάδα συνεδρίων δημιουργήθηκαν δύο σύνολα λέξεων κλειδιών. Τα σύνολα αυτά προέκυψαν ύστερα από σχετική έρευνα στις επίσημες ιστοσελίδες των συνεδρίων στην κατηγορία τομείς ενδιαφέροντος.

Έστω Q_{SIGMOD}, Q_{ECIR} τα σύνολα που κάθε στοιχείο τους αποτελείται από έναν όρο αναζήτησης.

$$Q_{SIGMOD} = \{q_1, \ldots, q_m\} \text{ και } Q_{ECIR} = \{q_1, \ldots, q_n\}$$

Οι L' λίστες αποτελούν την βάση σύγκρισης με τις λίστες L που ανακτώνται από την εφαρμογή. Όσο πιο πολλοί συγγραφείς της L βρίσκονται στην L' τόσο καλύτερη θεωρείται ότι είναι η μέθοδος. Εκτός από τις λίστες των συνεδρίων στις δοκιμές χρησιμοποιήθηκαν και ολόκληρες οι λίστες των δύο ομάδων

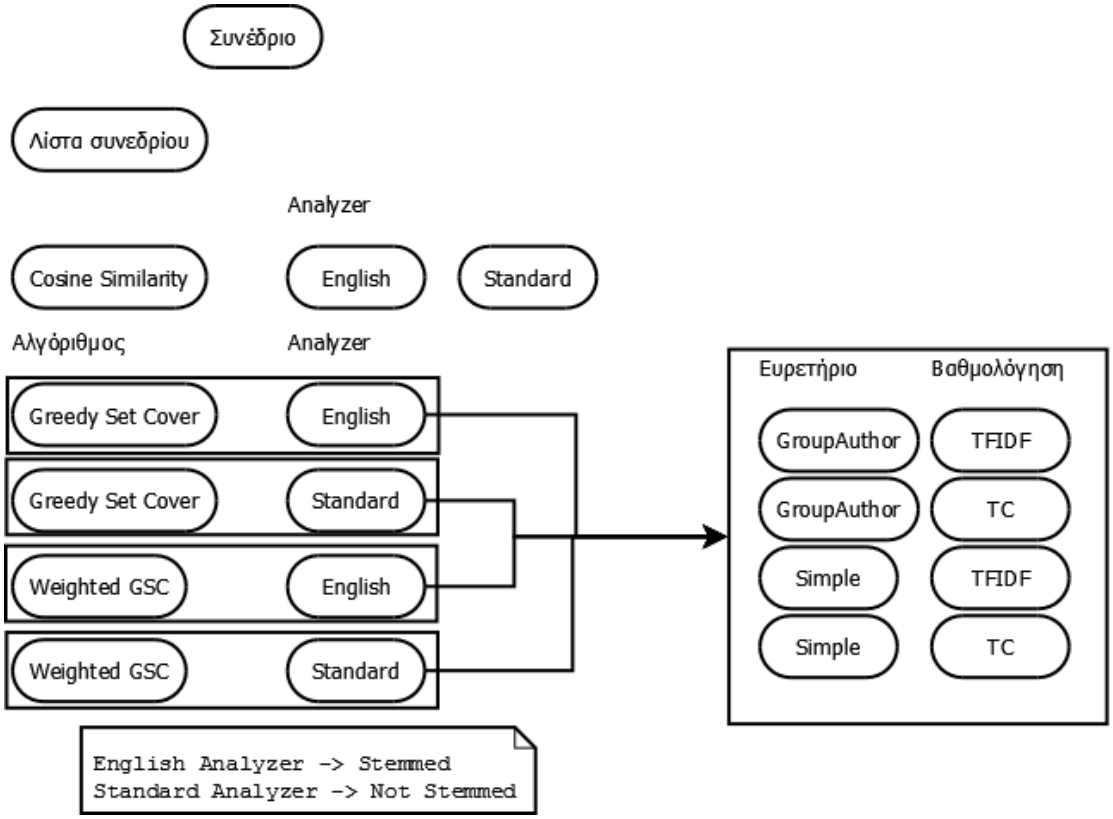


Εικόνα 35 Απεικόνιση λιστών αξιολόγησης

Στις περιπτώσεις όπου ένας συγγραφέας της λίστας L1 δεν υπάρχει στην λίστα L'1 τότε η βαθμολογία του είναι μηδέν(0), αλλιώς η βαθμολογία του είναι ίση με αυτήν από την λίστα L'1.

Οι προτεινόμενοι αλγόριθμοι για το πρόβλημα της εύρεσης ειδικών, είναι ο Greedy Set Cover, Weigted Set Cover, Custom Set Cover, Weighted Custom Greedy Set Cover. Για την πληρέστερη αποτύπωση των αποτελεσμάτων κάθε ένας από αυτούς δοκιμάστηκε με διαφορετικό συνδυασμό Analyzer, Ευρετήριο και βαθμολόγηση Lucene. Οι διαφοροποιήσεις αυτές δεν αναμένεται να δώσουν ουσιαστικό προβάδισμα σε κάποια μέθοδο, αποτελούν όμως μια πληροφορία που θα μπορούσε να αξιοποιηθεί σε μία εμπορική εφαρμογή.

Όλοι οι υπόλοιποι συνδυασμοί έχουν δευτερεύουσα σημασία και φαίνονται στο παρακάτω διάγραμμα.



Εικόνα 36 Συνδυασμοί μεθόδων προς αξιολόγηση

7.2 Βασική υλοποίηση (Cosine Similarity)

Η βασική υλοποίηση για την επίλυση του προβλήματος, είναι η χρήση του Cosine Similarity. Ο αλγόριθμος αυτός συγκρίνει δύο διανύσματα μετρώντας το συνημίτονο της γωνίας μεταξύ τους. Για την προσαρμογή του στο πρόβλημα της εύρεσης ειδικών θα πρέπει αρχικά να αναπαρασταθεί η πληροφορία του συνόλου Q ως διάνυσμα λέξεων. Στην συνέχεια για κάθε ερευνητή στο σύνολο R η πληροφορία d που τον συνοδεύει μετατρέπεται και αυτή σε διάνυσμα.

Για όλα τα ζευγάρια (d_q, d_i) υπολογίζεται η ομοιότητα συνημίτονου

$$\text{συν}(d_q, d_i) = (d_q * d_i)/||d_q||||d_i||$$

όπου * το εσωτερικό γινόμενο και $||d_q||$ το μήκος του d.

Το τελικό στάδιο του αλγορίθμου είναι η κατάταξη των ερευνητών έχοντας υπολογίσει την ομοιότητα συνημίτονου , από την μεγαλύτερη προς την μικρότερη.

7.3 Αποτελέσματα

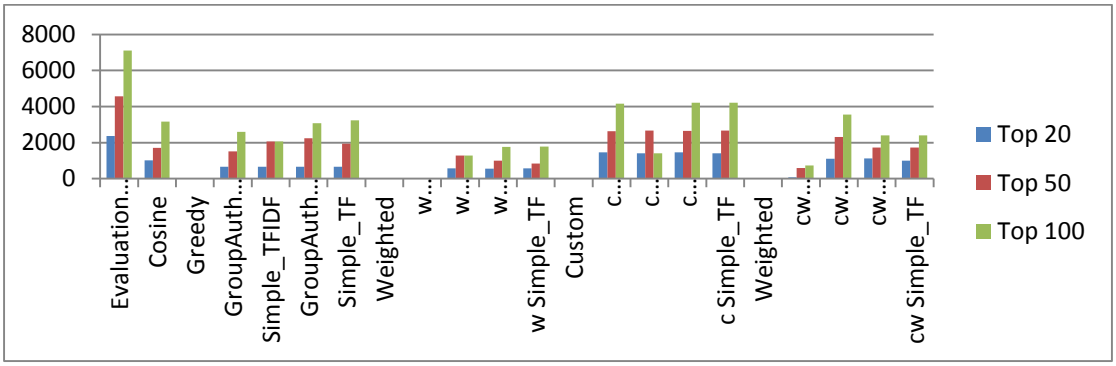
Η σύγκριση του Goodness παρουσιάζεται παρακάτω για τους πρώτους 20, 50 και 100 ειδικούς. Με κόκκινο χρώμα σημειώνονται οι τρεις βασικοί συνδυασμοί αλγορίθμων που συγκρίνονται.

7.3.1 Σύνολο λέξεων συνεδρίου

SIGMOD LIKE

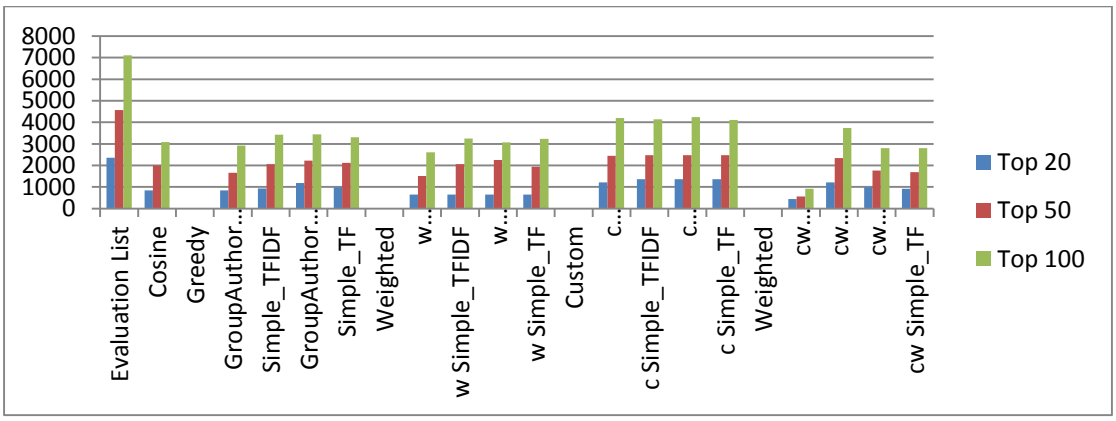
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	2361	4564	7104	
Cosine	1009	1710	3171	44,6368
Greedy				
GroupAuthor_TFIDF	656	1511	2605	36,6695
Simple_TFIDF	654	2064	2064	29,0541
GroupAuthor_TC	654	2250	3074	43,2714
Simple_TC	654	1937	3237	45,5659
Weighted				
w GroupAuthor_TFIDF	1	9	12	0,1689
w Simple_TFIDF	579	1286	1286	18,1025
w GroupAuthor_TC	547	1001	1771	24,9296
w Simple_TC	579	832	1772	24,9437
Custom				
c GroupAuthor_TFIDF	1468	2625	4158	58,5304
c Simple_TFIDF	1408	2661	1408	19,8198
c GroupAuthor_TC	1468	2655	4221	59,4172
c Simple_TC	1408	2661	4208	59,2342
Weighted				
cw GroupAuthor_TFIDF	71	594	734	10,3322
cw Simple_TFIDF	1102	2320	3550	49,9718
cw GroupAuthor_TC	1126	1722	2402	33,8119
cw Simple_TC	1004	1722	2404	33,8401



Stemmed

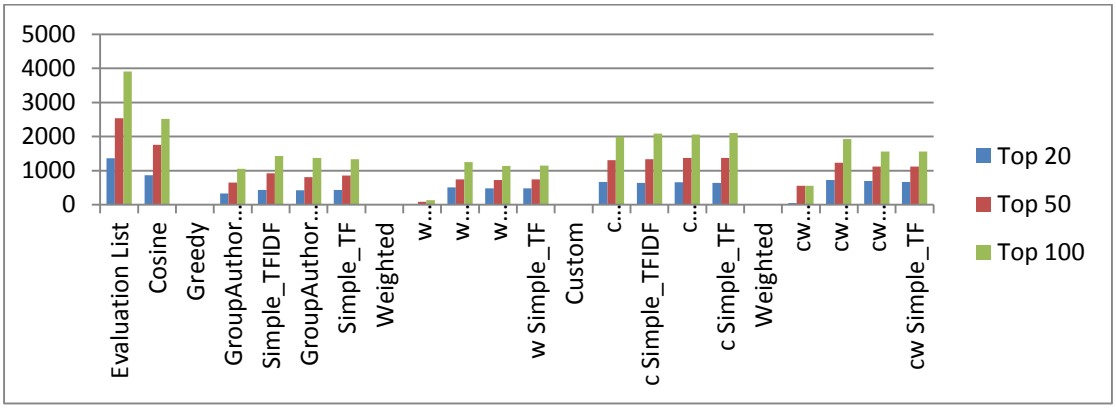
Goodness	Top 20	Top 50	Top 100	%
Cosine	849	2006	3081	43,36993
Greedy				
GroupAuthor_TFIDF	837	1652	2922	41,13176
Simple_TFIDF	932	2059	3419	48,12782
GroupAuthor_TC	1184	2228	3433	48,32489
Simple_TC	988	2111	3310	46,59347
Weighted				
w GroupAuthor_TFIDF	656	1511	2605	36,66948
w Simple_TFIDF	654	2064	3249	45,7348
w GroupAuthor_TC	654	2250	3074	43,2714
w Simple_TC	654	1937	3237	45,56588
Custom				
c GroupAuthor_TFIDF	1214	2442	4192	59,00901
c Simple_TFIDF	1360	2479	4140	58,27703
c GroupAuthor_TC	1360	2473	4235	59,6143
c Simple_TC	1360	2479	4109	57,84065
Weighted				
cw GroupAuthor_TFIDF	439	563	915	12,88007
cw Simple_TFIDF	1219	2338	3734	52,56194
cw GroupAuthor_TC	992	1765	2803	39,45664
cw Simple_TC	916	1686	2800	39,41441
Evaluation List	2361	4564	7104	



ECIR LIKE

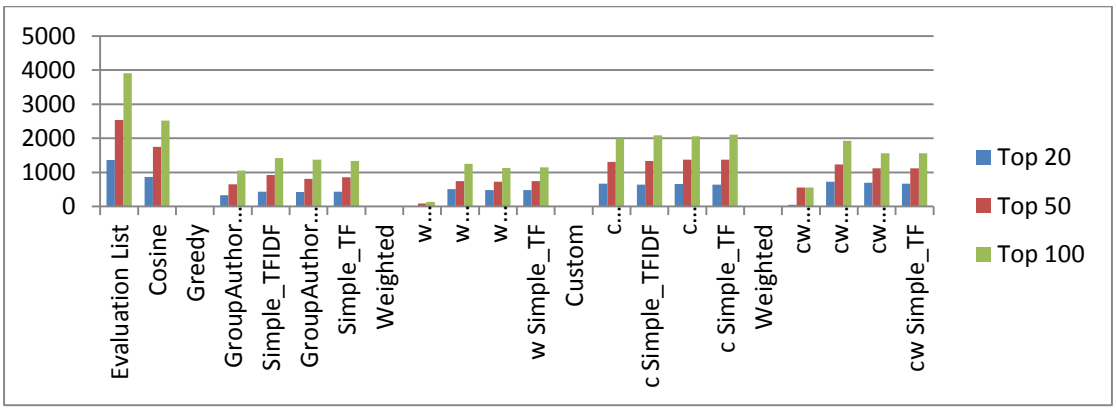
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Cosine	863	1753	2522	64,60041
Greedy				
GroupAuthor_TFIDF	326	646	1050	26,89549
Simple_TFIDF	436	921	1424	36,47541
GroupAuthor_TC	427	812	1372	35,14344
Simple_TC	435	856	1338	34,27254
Weighted				
w GroupAuthor_TFIDF	2	86	129	3,304303
w Simple_TFIDF	509	740	1251	32,04406
w GroupAuthor_TC	479	724	1133	29,02152
w Simple_TC	479	740	1144	29,30328
Custom				
c GroupAuthor_TFIDF	664	1305	1992	51,02459
c Simple_TFIDF	640	1335	2090	53,53484
c GroupAuthor_TC	662	1373	2055	52,63832
c Simple_TC	641	1376	2106	53,94467
Weighted				
cw GroupAuthor_TFIDF	45	553	559	14,31865
cw Simple_TFIDF	723	1235	1930	49,43648
cw GroupAuthor_TC	696	1123	1557	39,88217
cw Simple_TC	671	1115	1561	39,98463
Evaluation List	1363	2539	3904	



Stemmed

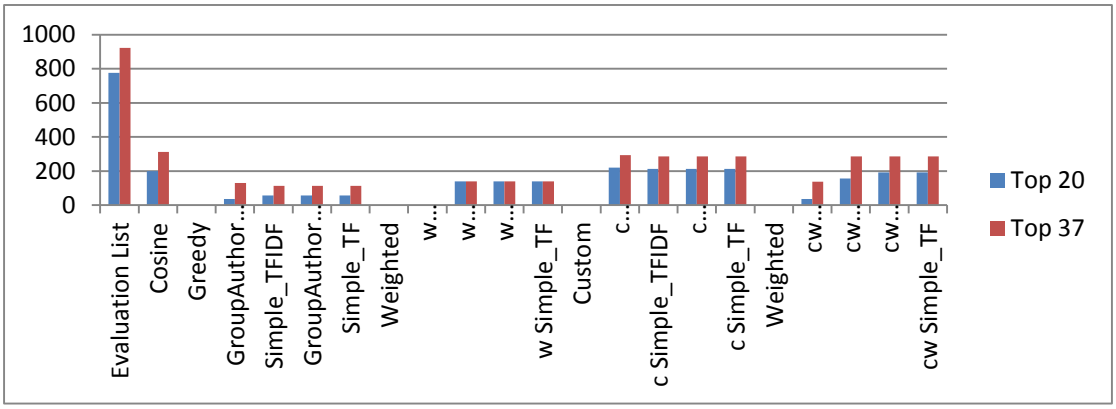
Goodness	Top 20	Top 50	Top 100	%
Cosine	802	1395	2197	56,27561
Greedy				
GroupAuthor_TFIDF	541	935	1287	32,96619
Simple_TFIDF	611	1002	1570	40,21516
GroupAuthor_TC	569	1003	1412	36,16803
Simple_TC	611	950	1365	34,96414
Weighted				
w GroupAuthor_TFIDF	326	646	1050	26,89549
w Simple_TFIDF	436	921	1424	36,47541
w GroupAuthor_TC	427	812	1372	35,14344
w Simple_TC	435	856	1338	34,27254
Custom				
c GroupAuthor_TFIDF	662	1192	2123	54,38012
c Simple_TFIDF	650	1230	2109	54,02152
c GroupAuthor_TC	684	1242	2104	53,89344
c Simple_TC	650	1225	2107	53,97029
Weighted				
cw GroupAuthor_TFIDF	390	648	665	17,03381
cw Simple_TFIDF	738	1346	2203	56,4293
cw GroupAuthor_TC	732	1363	2015	51,61373
cw Simple_TC	766	1344	1949	49,92316
Evaluation List	1363	2539	3904	



ECIR 2013

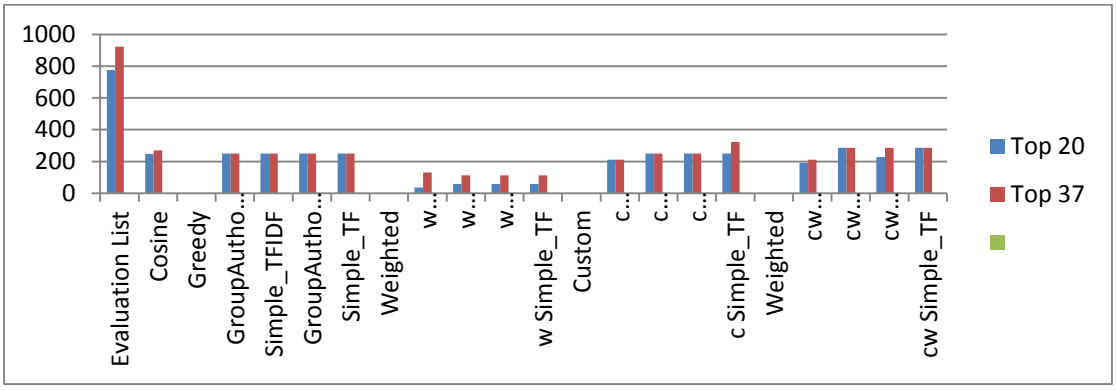
Not Stemmed

Goodness	Top 20	Top 37
Evaluation List	775	922
Cosine	200	311
Greedy		
GroupAuthor_TFIDF	36	130
Simple_TFIDF	57	113
GroupAuthor_TC	57	113
Simple_TC	57	113
Weighted		
w GroupAuthor_TFIDF	0	0
w Simple_TFIDF	139	139
w GroupAuthor_TC	139	139
w Simple_TC	139	139
Custom		
c GroupAuthor_TFIDF	220	293
c Simple_TFIDF	212	285
c GroupAuthor_TC	212	285
c Simple_TC	212	285
Weighted		
cw GroupAuthor_TFIDF	36	138
cw Simple_TFIDF	155	285
cw GroupAuthor_TC	192	285
cw Simple_TC	192	285



Stemmed

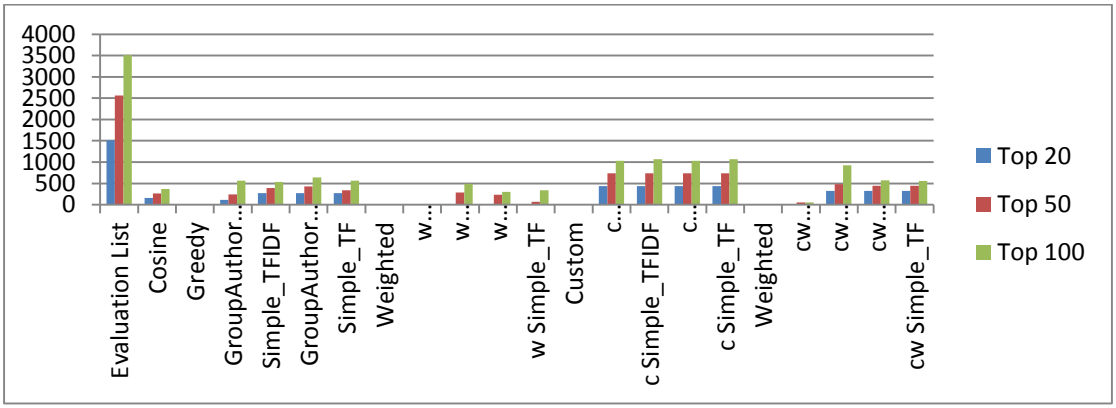
Goodness	Top 20	Top 37
Evaluation List	775	922
Cosine	247	270
Greedy		
GroupAuthor_TFIDF	249	249
Simple_TFIDF	249	249
GroupAuthor_TC	249	249
Simple_TC	249	249
Weighted		
w GroupAuthor_TFIDF	36	130
w Simple_TFIDF	57	113
w GroupAuthor_TC	57	113
w Simple_TC	57	113
Custom		
c GroupAuthor_TFIDF	212	212
c Simple_TFIDF	249	249
c GroupAuthor_TC	249	249
c Simple_TC	249	322
Weighted		
cw GroupAuthor_TFIDF	191	211
cw Simple_TFIDF	285	285
cw GroupAuthor_TC	228	285
cw Simple_TC	285	285



ICDE 2013

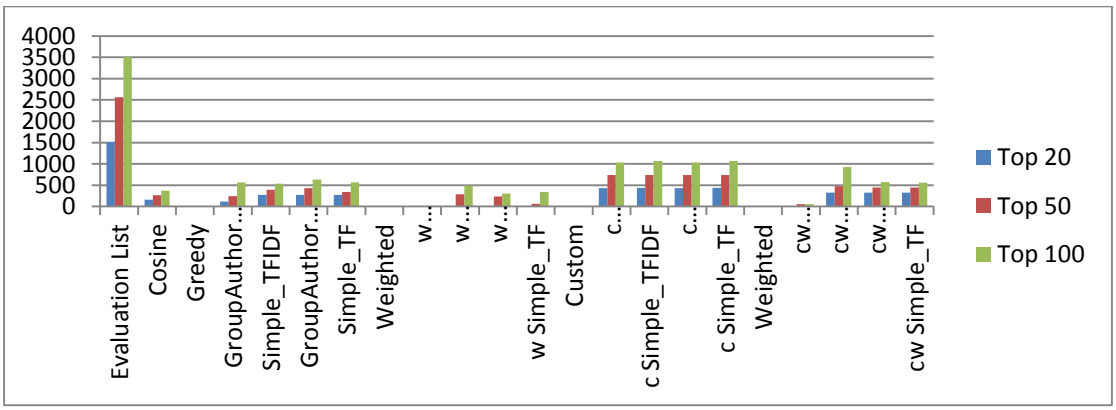
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1509	2561	3508	
Cosine	162	264	370	10,54732
Greedy				
GroupAuthor_TFIDF	114	240	564	16,07754
Simple_TFIDF	269	395	533	15,19384
GroupAuthor_TC	269	429	636	18,12999
Simple_TC	269	341	567	16,16306
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	11	288	490	13,96807
w GroupAuthor_TC	11	231	299	8,523375
w Simple_TC	11	65	342	9,749145
Custom				
c GroupAuthor_TFIDF	433	738	1028	29,30445
c Simple_TFIDF	434	738	1065	30,35918
c GroupAuthor_TC	433	738	1028	29,30445
c Simple_TC	434	738	1065	30,35918
Weighted				
cw GroupAuthor_TFIDF	0	51	52	1,482326
cw Simple_TFIDF	321	478	927	26,42531
cw GroupAuthor_TC	321	444	575	16,39111
cw Simple_TC	321	444	554	15,79247



Stemmed

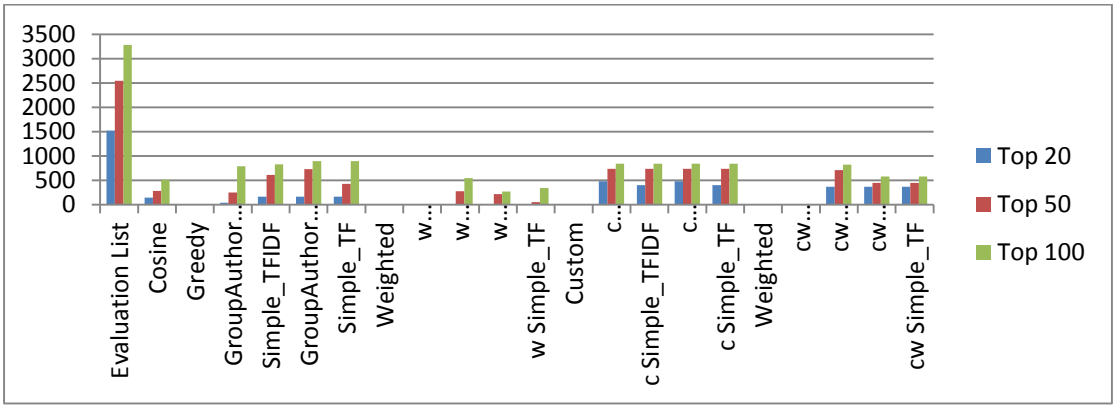
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1509	2561	3508	
Cosine	137	162	327	9,321551
Greedy				
GroupAuthor_TFIDF	151	423	864	24,62942
Simple_TFIDF	177	355	751	21,40821
GroupAuthor_TC	300	424	795	22,66249
Simple_TC	177	355	676	19,27024
Weighted				
w GroupAuthor_TFIDF	114	240	564	16,07754
w Simple_TFIDF	269	395	533	15,19384
w GroupAuthor_TC	269	429	636	18,12999
w Simple_TC	269	341	567	16,16306
Custom				
c GroupAuthor_TFIDF	374	727	946	26,96693
c Simple_TFIDF	540	727	978	27,87913
c GroupAuthor_TC	540	727	931	26,53934
c Simple_TC	540	727	931	26,53934
Weighted				
cw GroupAuthor_TFIDF	52	52	104	2,964652
cw Simple_TFIDF	321	532	1041	29,67503
cw GroupAuthor_TC	270	444	678	19,32725
cw Simple_TC	270	444	678	19,32725



SIGMOD 2013

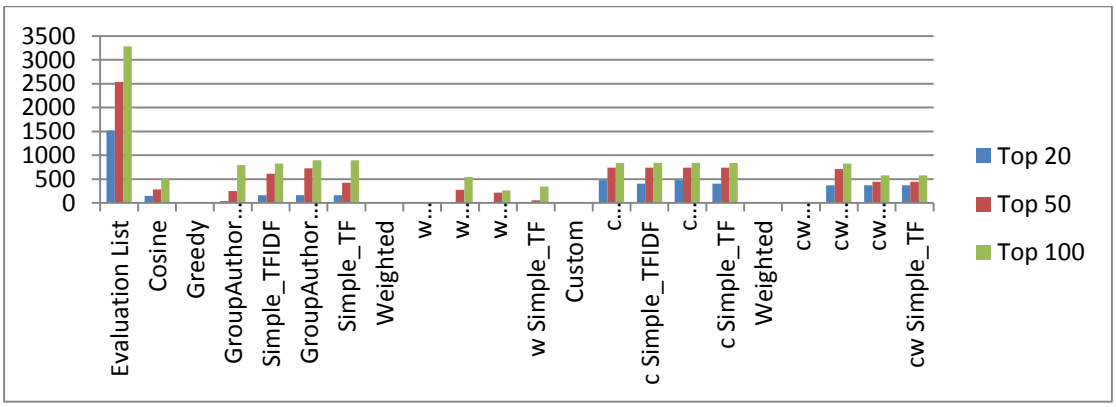
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1521	2542	3283	
Cosine	148	282	511	15,56503
Greedy				
GroupAuthor_TFIDF	42	252	792	24,12428
Simple_TFIDF	166	615	831	25,31221
GroupAuthor_TC	166	728	897	27,32257
Simple_TC	166	428	897	27,32257
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	0	276	546	16,63113
w GroupAuthor_TC	0	220	267	8,132805
w Simple_TC	0	54	343	10,44776
Custom				
c GroupAuthor_TFIDF	481	740	844	25,70819
c Simple_TFIDF	402	740	844	25,70819
c GroupAuthor_TC	481	740	844	25,70819
c Simple_TC	402	740	844	25,70819
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	369	712	825	25,12945
cw GroupAuthor_TC	369	448	579	17,63631
cw Simple_TC	369	448	579	17,63631



Stemmed

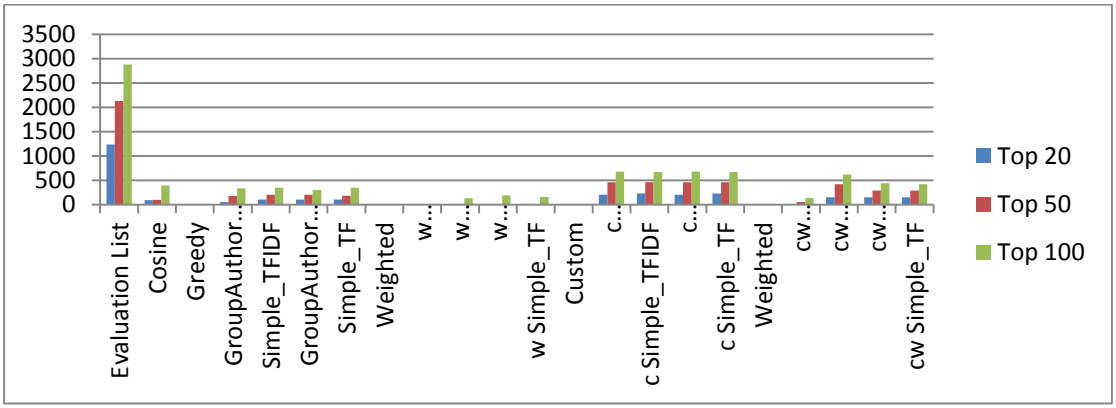
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1521	2542	3283	
Cosine	118	335	499	15,19951
Greedy				
GroupAuthor_TFIDF	164	416	749	22,8145
Simple_TFIDF	299	463	857	26,10417
GroupAuthor_TC	432	581	862	26,25647
Simple_TC	299	366	726	22,11392
Weighted				
w GroupAuthor_TFIDF	42	252	792	24,12428
w Simple_TFIDF	166	615	831	25,31221
w GroupAuthor_TC	166	728	897	27,32257
w Simple_TC	166	428	897	27,32257
Custom				
c GroupAuthor_TFIDF	268	643	882	26,86567
c Simple_TFIDF	434	643	840	25,58635
c GroupAuthor_TC	434	643	840	25,58635
c Simple_TC	434	643	840	25,58635
Weighted				
cw GroupAuthor_TFIDF	0	0	52	1,583917
cw Simple_TFIDF	369	581	918	27,96223
cw GroupAuthor_TC	369	448	764	23,2714
cw Simple_TC	369	369	764	23,2714



VLDB 2013

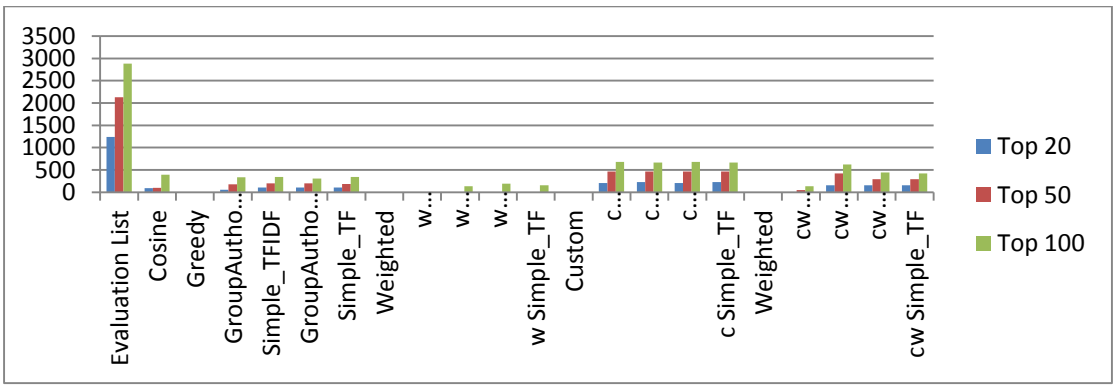
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1239	2128	2883	
Cosine	90	98	397	13,77038
Greedy				
GroupAuthor_TFIDF	54	180	335	11,61984
Simple_TFIDF	104	203	347	12,03607
GroupAuthor_TC	104	203	306	10,61394
Simple_TC	104	186	347	12,03607
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	0	0	133	4,61325
w GroupAuthor_TC	0	0	192	6,659729
w Simple_TC	0	0	159	5,515088
Custom				
c GroupAuthor_TFIDF	207	464	678	23,51717
c Simple_TFIDF	231	464	668	23,17031
c GroupAuthor_TC	207	464	678	23,51717
c Simple_TC	231	464	668	23,17031
Weighted				
cw GroupAuthor_TFIDF	0	51	139	4,821367
cw Simple_TFIDF	154	421	620	21,50538
cw GroupAuthor_TC	154	292	442	15,33125
cw Simple_TC	154	292	421	14,60284



Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1239	2128	2883	
Cosine	98	150	276	9,573361
Greedy				
GroupAuthor_TFIDF	63	115	350	12,14013
Simple_TFIDF	12	76	262	9,087756
GroupAuthor_TC	12	115	272	9,434617
Simple_TC	12	76	221	7,665626
Weighted				
w GroupAuthor_TFIDF	54	180	335	11,61984
w Simple_TFIDF	104	203	347	12,03607
w GroupAuthor_TC	104	203	306	10,61394
w Simple_TC	104	186	347	12,03607
Custom				
c GroupAuthor_TFIDF	207	335	520	18,03677
c Simple_TFIDF	207	464	597	20,7076
c GroupAuthor_TC	207	464	597	20,7076
c Simple_TC	207	464	597	20,7076
Weighted				
cw GroupAuthor_TFIDF	51	65	191	6,625043
cw Simple_TFIDF	154	421	708	24,55775
cw GroupAuthor_TC	160	235	509	17,65522
cw Simple_TC	160	235	509	17,65522



7.3.2 Συγκεντρωτικά

Στα συγκεντρωτικά αποτελέσματα δεν εμφανίζονται οι περιπτώσεις των αλγορίθμων με βάρη (weighted) καθώς οι επιδόσεις τους είναι εμφανώς χαμηλότερες.

SIGMOD LIKE	Top 20	Top 50	Top 100	%
Evaluation List	2361	4564	7104	
Cosine	849	2006	3081	43,36993
GroupAuthor_TC	1184	2228	3433	48,32489
c GroupAuthor_TC	1360	2473	4235	59,6143
ECIR LIKE				
Evaluation List	1363	2539	3904	
Cosine	802	1395	2197	56,27561
GroupAuthor_TC	569	1003	1412	36,16803
c GroupAuthor_TC	684	1242	2104	53,89344
ECIR 2013				
Evaluation List	775	922		
Cosine	247	270		
GroupAuthor_TC	249	249		
c GroupAuthor_TC	249	249		
ICDE 2013				
Evaluation List	1509	2561	3508	
Cosine	137	162	327	9,321551
GroupAuthor_TC	300	424	795	22,66249
c GroupAuthor_TC	540	727	931	26,53934
SIGMOD 2013				
Evaluation List	1521	2542	3283	
Cosine	118	335	499	15,19951
GroupAuthor_TC	432	581	862	26,25647
c GroupAuthor_TC	434	643	840	25,58635
VLDB 2013				
Evaluation List	1239	2128	2883	
Cosine	98	150	276	9,573361
GroupAuthor_TC	12	115	272	9,434617
c GroupAuthor_TC	207	464	597	20,7076

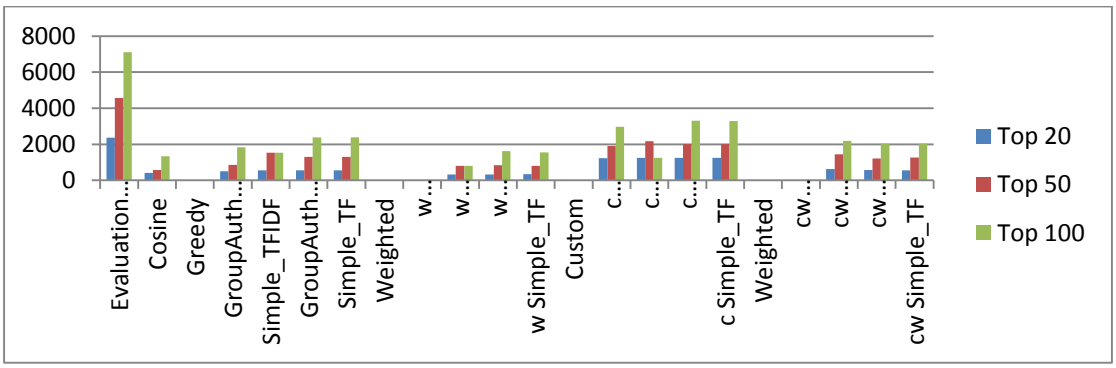
7.3.3 Τυχαίες λέξεις

Καθώς τα αποτελέσματα δεν ήταν πολύ ξεκάθαρα, κρίθηκε απαραίτητο να γίνει δοκιμή με λιγότερες λέξεις. Για τις παρακάτω δοκιμές επιλέχθηκαν τυχαία 15 λέξεις από το σύνολο των όρων κάθε συνεδρίου.

SIGMOD LIKE

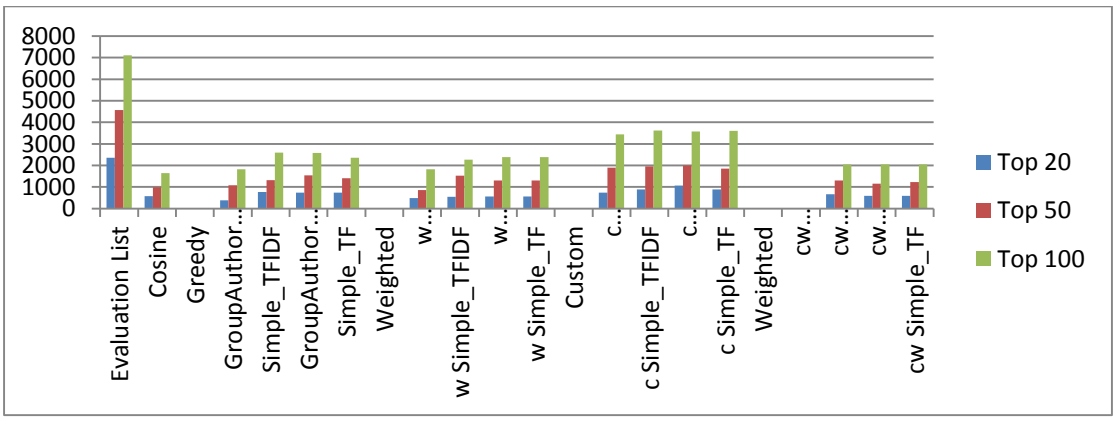
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	2361	4564	7104	
Cosine	413	573	1328	18,6937
Greedy				
GroupAuthor_TFIDF	492	853	1824	25,6757
Simple_TFIDF	544	1524	1524	21,4527
GroupAuthor_TC	558	1303	2380	33,5023
Simple_TC	558	1303	2380	33,5023
Weighted				
w GroupAuthor_TFIDF	2	4	11	0,1548
w Simple_TFIDF	314	798	798	11,2331
w GroupAuthor_TC	329	833	1613	22,7055
w Simple_TC	335	806	1556	21,9032
Custom				
c GroupAuthor_TFIDF	1232	1900	2964	41,7230
c Simple_TFIDF	1240	2174	1240	17,4550
c GroupAuthor_TC	1240	2012	3314	46,6498
c Simple_TC	1240	2009	3280	46,1712
Weighted				
cw GroupAuthor_TFIDF	3	9	13	0,1830
cw Simple_TFIDF	625	1443	2181	30,7010
cw GroupAuthor_TC	568	1216	2019	28,4206
cw Simple_TC	555	1270	2021	28,4488



Stemmed

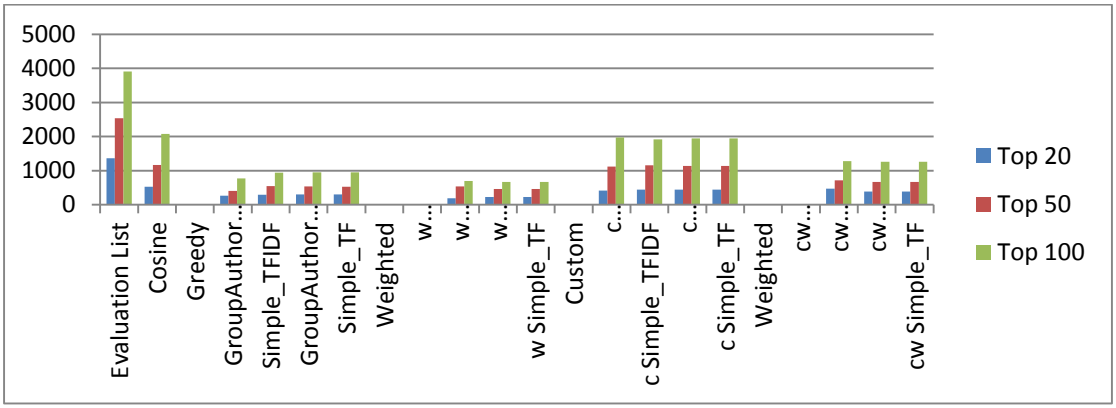
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	2361	4564	7104	
Cosine	579	997	1639	23,07151
Greedy				
GroupAuthor_TFIDF	378	1078	1818	25,59122
Simple_TFIDF	771	1317	2589	36,44426
GroupAuthor_TC	732	1534	2578	36,28941
Simple_TC	732	1400	2350	33,07995
Weighted				
w GroupAuthor_TFIDF	492	853	1824	25,67568
w Simple_TFIDF	544	1524	2268	31,92568
w GroupAuthor_TC	558	1303	2380	33,50225
w Simple_TC	558	1303	2380	33,50225
Custom				
c GroupAuthor_TFIDF	734	1899	3434	48,33896
c Simple_TFIDF	891	1948	3622	50,98536
c GroupAuthor_TC	1064	1998	3574	50,30968
c Simple_TC	891	1852	3606	50,76014
Weighted				
cw GroupAuthor_TFIDF	2	3	8	0,112613
cw Simple_TFIDF	670	1297	2043	28,75845
cw GroupAuthor_TC	590	1153	2044	28,77252
cw Simple_TC	590	1222	2046	28,80068



ECIR LIKE

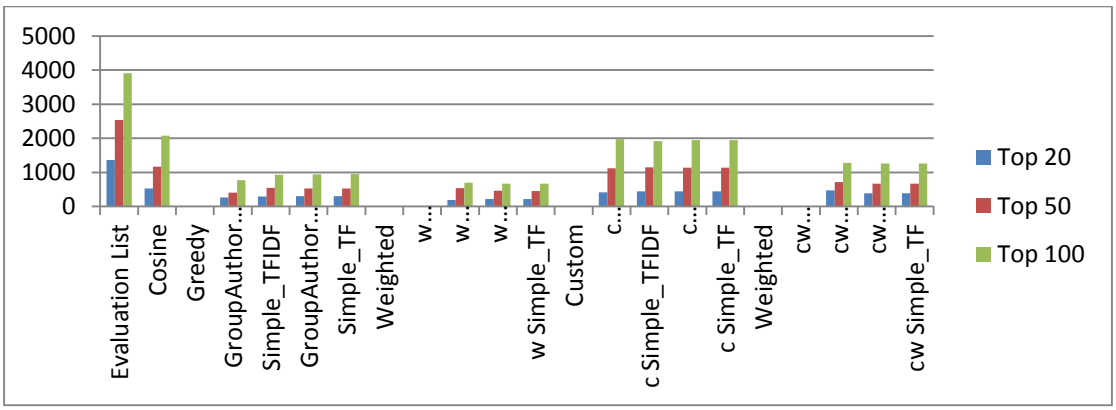
Not Stemmed

Evaluation List	1363	2539	3904	
Cosine	526	1170	2080	53,27869
Greedy				
GroupAuthor_TFIDF	264	403	775	19,85143
Simple_TFIDF	292	544	936	23,97541
GroupAuthor_TC	306	532	945	24,20594
Simple_TC	302	528	946	24,23156
Weighted				
w GroupAuthor_TFIDF	0	5	12	0,307377
w Simple_TFIDF	187	533	699	17,90471
w GroupAuthor_TC	222	461	668	17,11066
w Simple_TC	222	457	665	17,03381
Custom				
c GroupAuthor_TFIDF	414	1120	1974	50,56352
c Simple_TFIDF	446	1152	1921	49,20594
c GroupAuthor_TC	446	1135	1945	49,8207
c Simple_TC	446	1135	1945	49,8207
Weighted				
cw GroupAuthor_TFIDF	0	0	3	0,076844
cw Simple_TFIDF	475	712	1275	32,65881
cw GroupAuthor_TC	385	667	1262	32,32582
cw Simple_TC	385	667	1262	32,32582
Evaluation List	1363	2539	3904	



Stemmed

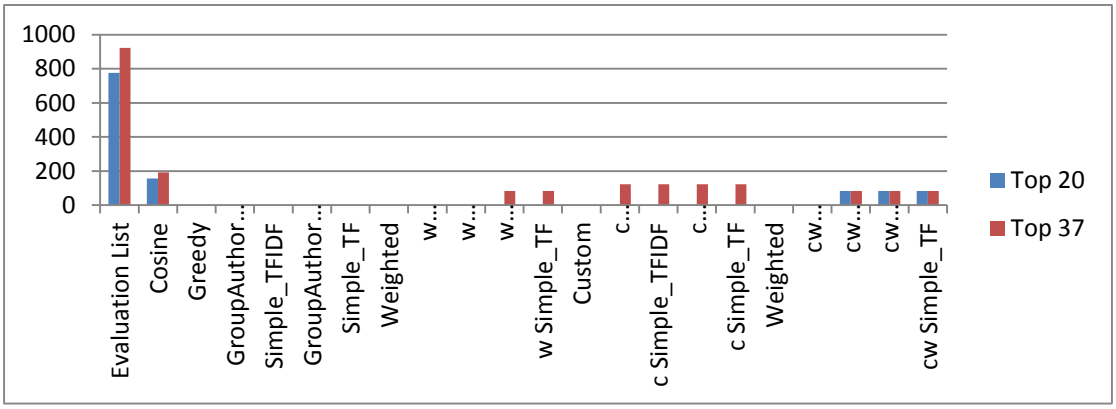
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1363	2539	3904	
Cosine	326	638	1028	26,33197
Greedy				
GroupAuthor_TFIDF	425	560	1063	27,22848
Simple_TFIDF	344	851	1294	33,14549
GroupAuthor_TC	356	788	1311	33,58094
Simple_TC	356	817	1307	33,47848
Weighted				
w GroupAuthor_TFIDF	264	403	775	19,85143
w Simple_TFIDF	292	544	936	23,97541
w GroupAuthor_TC	306	532	945	24,20594
w Simple_TC	302	528	946	24,23156
Custom				
c GroupAuthor_TFIDF	588	1192	1791	45,87602
c Simple_TFIDF	723	1218	1870	47,89959
c GroupAuthor_TC	739	1322	1851	47,41291
c Simple_TC	739	1318	1872	47,95082
Weighted				
cw GroupAuthor_TFIDF	3	3	8	0,204918
cw Simple_TFIDF	519	808	1330	34,06762
cw GroupAuthor_TC	646	1202	1680	43,03279
cw Simple_TC	646	1202	1680	43,03279



ECIR 2013

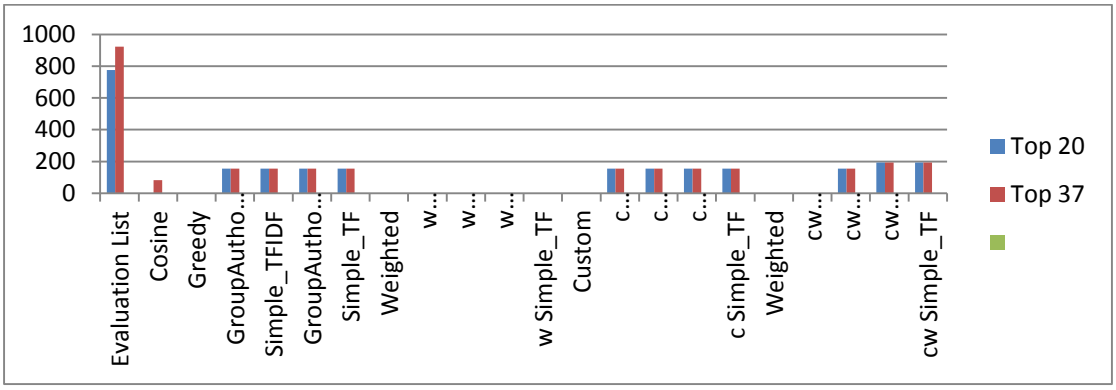
Not Stemmed

Goodness	Top 20	Top 37
Evaluation List	775	922
Cosine	155	191
Greedy		
GroupAuthor_TFIDF	0	0
Simple_TFIDF	0	0
GroupAuthor_TC	0	0
Simple_TC	0	0
Weighted		
w GroupAuthor_TFIDF	0	0
w Simple_TFIDF	0	0
w GroupAuthor_TC	0	82
w Simple_TC	0	82
Custom		
c GroupAuthor_TFIDF	0	122
c Simple_TFIDF	0	122
c GroupAuthor_TC	0	122
c Simple_TC	0	122
Weighted		
cw GroupAuthor_TFIDF	0	0
cw Simple_TFIDF	82	82
cw GroupAuthor_TC	82	82
cw Simple_TC	82	82



Stemmed

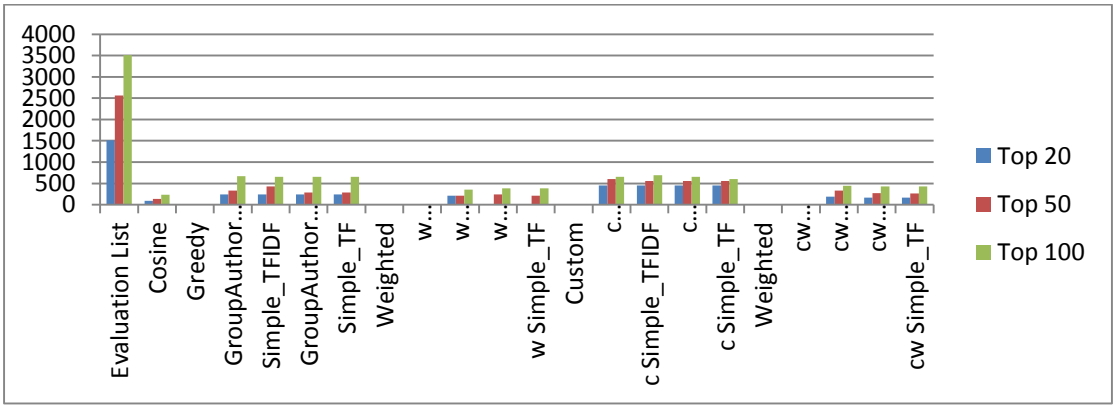
Goodness	Top 20	Top 37
Evaluation List	775	922
Cosine	0	82
Greedy		
GroupAuthor_TFIDF	155	155
Simple_TFIDF	155	155
GroupAuthor_TC	155	155
Simple_TC	155	155
Weighted		
w GroupAuthor_TFIDF	0	0
w Simple_TFIDF	0	0
w GroupAuthor_TC	0	0
w Simple_TC	0	0
Custom		
c GroupAuthor_TFIDF	155	155
c Simple_TFIDF	155	155
c GroupAuthor_TC	155	155
c Simple_TC	155	155
Weighted		
cw GroupAuthor_TFIDF	0	0
cw Simple_TFIDF	155	155
cw GroupAuthor_TC	192	192
cw Simple_TC	192	192



ICDE 2013

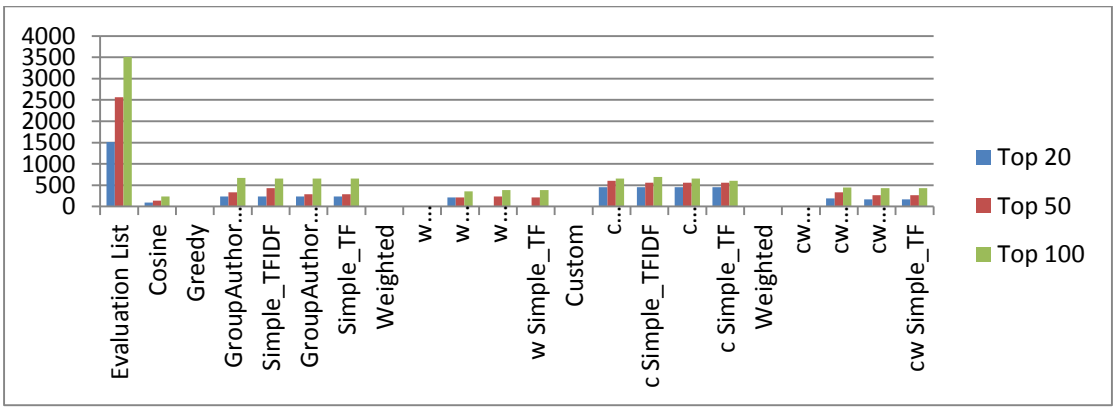
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1509	2561	3508	
Cosine	90	139	237	6,755986
Greedy				
GroupAuthor_TFIDF	238	331	669	19,0707
Simple_TFIDF	238	431	654	18,6431
GroupAuthor_TC	238	289	654	18,6431
Simple_TC	238	289	654	18,6431
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	211	211	357	10,17674
w GroupAuthor_TC	0	238	384	10,94641
w Simple_TC	0	211	384	10,94641
Custom				
c GroupAuthor_TFIDF	455	601	657	18,72862
c Simple_TFIDF	455	559	693	19,75485
c GroupAuthor_TC	455	559	657	18,72862
c Simple_TC	455	560	602	17,16078
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	190	330	444	12,65678
cw GroupAuthor_TC	166	268	432	12,31471
cw Simple_TC	166	265	432	12,31471



Stemmed

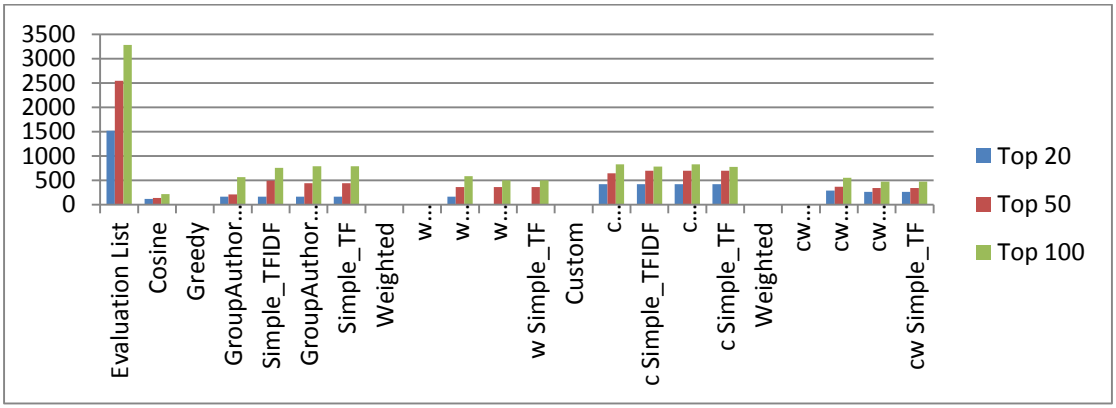
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1509	2561	3508	
Cosine	98	122	122	3,477765
Greedy				
GroupAuthor_TFIDF	72	238	497	14,16762
Simple_TFIDF	284	423	646	18,41505
GroupAuthor_TC	239	378	508	14,48119
Simple_TC	239	378	510	14,5382
Weighted				
w GroupAuthor_TFIDF	238	331	669	19,0707
w Simple_TFIDF	238	431	654	18,6431
w GroupAuthor_TC	238	289	654	18,6431
w Simple_TC	238	289	654	18,6431
Custom				
c GroupAuthor_TFIDF	377	585	638	18,187
c Simple_TFIDF	377	480	742	21,15165
c GroupAuthor_TC	377	562	742	21,15165
c Simple_TC	377	480	742	21,15165
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	242	330	465	13,25542
cw GroupAuthor_TC	218	357	468	13,34094
cw Simple_TC	218	357	465	13,25542



SIGMOD 2013

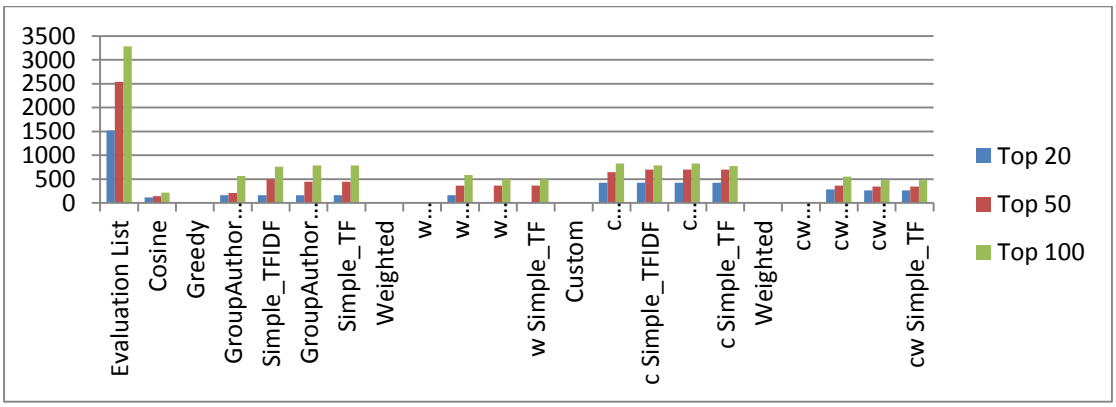
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1521	2542	3283	
Cosine	118	142	220	6,701188
Greedy				
GroupAuthor_TFIDF	166	208	568	17,30125
Simple_TFIDF	166	496	758	23,08864
GroupAuthor_TC	166	442	790	24,06336
Simple_TC	166	442	790	24,06336
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	166	363	588	17,91045
w GroupAuthor_TC	0	363	500	15,22997
w Simple_TC	0	363	500	15,22997
Custom				
c GroupAuthor_TFIDF	423	645	829	25,25129
c Simple_TFIDF	423	700	785	23,91106
c GroupAuthor_TC	423	700	829	25,25129
c Simple_TC	423	700	774	23,576
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	287	366	554	16,87481
cw GroupAuthor_TC	263	342	477	14,52939
cw Simple_TC	263	342	477	14,52939



Stemmed

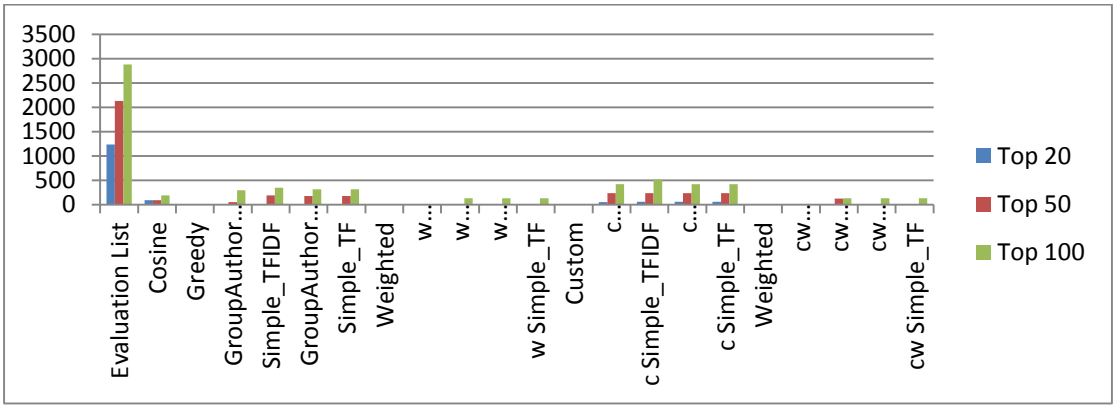
Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1521	2542	3283	
Cosine	79	103	103	3,137374
Greedy				
GroupAuthor_TFIDF	22	188	484	14,74261
Simple_TFIDF	166	205	690	21,01736
GroupAuthor_TC	166	302	636	19,37253
Simple_TC	166	263	690	21,01736
Weighted				
w GroupAuthor_TFIDF	166	208	568	17,30125
w Simple_TFIDF	166	496	758	23,08864
w GroupAuthor_TC	166	442	790	24,06336
w Simple_TC	166	442	790	24,06336
Custom				
c GroupAuthor_TFIDF	166	347	798	24,30704
c Simple_TFIDF	245	544	901	27,44441
c GroupAuthor_TC	245	520	857	26,10417
c Simple_TC	245	544	901	27,44441
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	290	466	466	14,19433
cw GroupAuthor_TC	266	387	466	14,19433
cw Simple_TC	266	466	466	14,19433



VLDB 2013

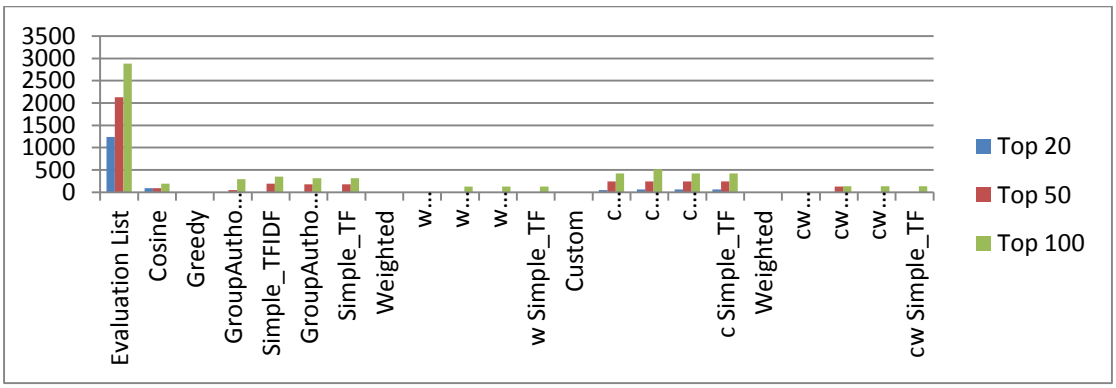
Not Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1239	2128	2883	
Cosine	90	90	192	6,659729
Greedy				
GroupAuthor_TFIDF	0	51	295	10,2324
Simple_TFIDF	0	192	352	12,2095
GroupAuthor_TC	0	180	318	11,03018
Simple_TC	0	180	318	11,03018
Weighted				
w GroupAuthor_TFIDF	0	0	0	0
w Simple_TFIDF	0	0	129	4,474506
w GroupAuthor_TC	0	0	129	4,474506
w Simple_TC	0	0	129	4,474506
Custom				
c GroupAuthor_TFIDF	51	240	424	14,7069
c Simple_TFIDF	63	240	515	17,86334
c GroupAuthor_TC	63	240	424	14,7069
c Simple_TC	63	240	424	14,7069
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	0	125	133	4,61325
cw GroupAuthor_TC	0	8	133	4,61325
cw Simple_TC	0	8	133	4,61325



Stemmed

Goodness	Top 20	Top 50	Top 100	%
Evaluation List	1239	2128	2883	
Cosine	59	59	59	2,046479
Greedy				
GroupAuthor_TFIDF	25	166	240	8,324662
Simple_TFIDF	129	217	331	11,4811
GroupAuthor_TC	129	240	292	10,12834
Simple_TC	129	215	240	8,324662
Weighted				
w GroupAuthor_TFIDF	0	51	295	10,2324
w Simple_TFIDF	0	192	352	12,2095
w GroupAuthor_TC	0	180	318	11,03018
w Simple_TC	0	180	318	11,03018
Custom				
c GroupAuthor_TFIDF	63	266	418	14,49879
c Simple_TFIDF	63	266	520	18,03677
c GroupAuthor_TC	63	192	452	15,67811
c Simple_TC	63	266	520	18,03677
Weighted				
cw GroupAuthor_TFIDF	0	0	0	0
cw Simple_TFIDF	51	125	161	5,584461
cw GroupAuthor_TC	51	51	161	5,584461
cw Simple_TC	51	51	161	5,584461



7.3.4 Συγκεντρωτικά

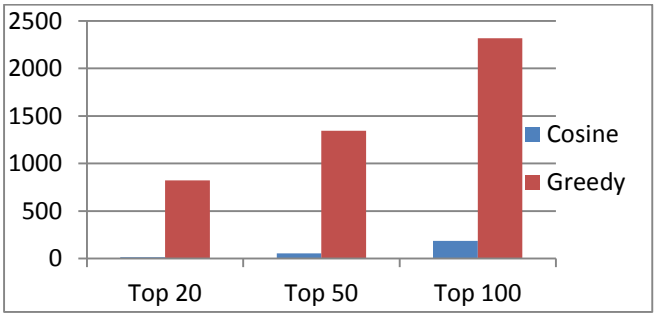
SIGMOD LIKE	Top 20	Top 50	Top 100	%
Evaluation List	2361	4564	7104	
Cosine	579	997	1639	23,07151
GroupAuthor_TC	732	1534	2578	36,28941
c GroupAuthor_TC	1064	1998	3574	50,30968
ECIR LIKE				
Evaluation List	1363	2539	3904	
Cosine	326	638	1028	26,33197
GroupAuthor_TC	356	788	1311	33,58094
c GroupAuthor_TC	739	1322	1851	47,41291
ECIR 2013				
Evaluation List	775	922		
Cosine	0	82		
GroupAuthor_TC	155	155		
c GroupAuthor_TC	155	155		
ICDE 2013				
Evaluation List	1509	2561	3508	
Cosine	98	122	122	3,477765
GroupAuthor_TC	239	378	508	14,48119
c GroupAuthor_TC	377	562	742	21,15165
SIGMOD 2013				
Evaluation List	1521	2542	3283	
Cosine	79	103	103	3,137374
GroupAuthor_TC	166	302	636	19,37253
c GroupAuthor_TC	245	520	857	26,10417
VLDB 2013				
Evaluation List	1239	2128	2883	
Cosine	59	59	59	2,046479
GroupAuthor_TC	129	240	292	10,12834
c GroupAuthor_TC	63	192	452	15,67811

7.3.5 Όροι με αύξουσα σειρά

Εκτός από τις δοκιμές με το σύνολο των τομέων ενδιαφέροντος από τα συνέδρια, κρίθηκε απαραίτητο να γίνουν δοκιμές με μεταβλητό πλήθος λέξεων κλειδιών στην αναζήτηση. Οι λέξεις αυτές έχουν επιλεχθεί τυχαία από την ομάδα του SIGMOD.

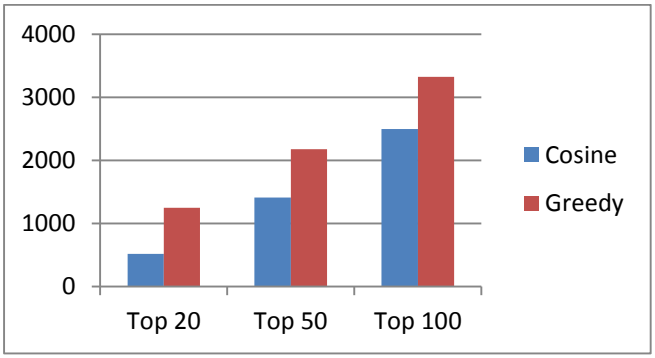
5 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	15	55	187
Greedy	823	1345	2318



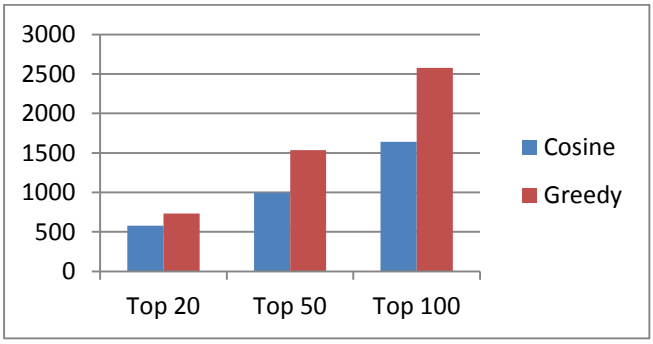
10 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	519	1415	2500
Greedy	1249	2177	3325



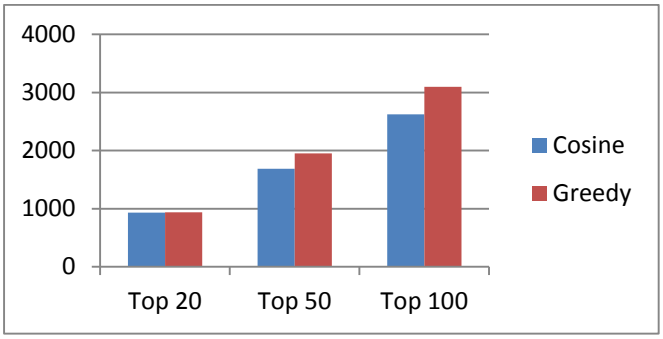
15 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	579	997	1639
Greedy	732	1534	2578



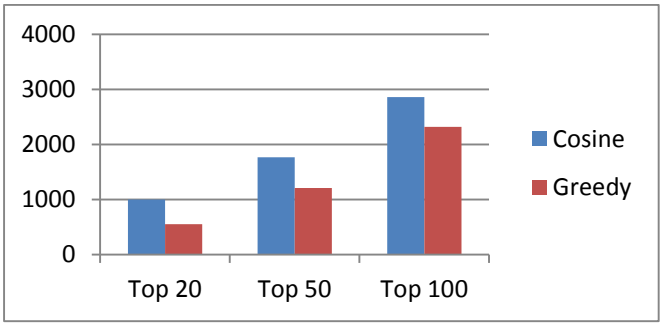
20 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	932	1686	2625
Greedy	936	1953	3096



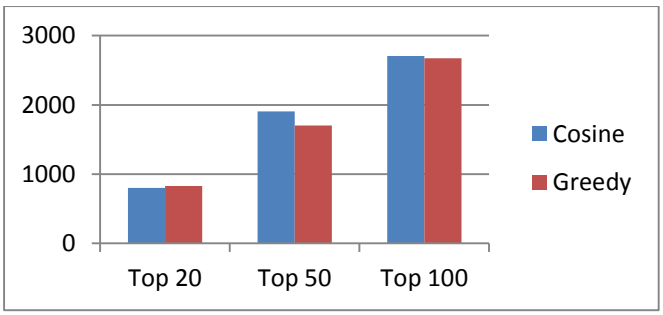
25 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	998	1767	2859
Greedy	548	1206	2319



30 λέξεις αναζήτησης

Goodness	Top 20	Top 50	Top 100
Cosine	802	1904	2705
Greedy	828	1703	2672



8 Συμπεράσματα και μελλοντική εργασία

8.1 Συμπεράσματα

Ξεκινώντας το κεφάλαιο θα γίνει αρχικά αναφορά στις διαφοροποιήσεις των αλγορίθμων με βάση των τύπο ευρετηρίων και βαθμολόγησης και αφού βγει το πόρισμα για τον καλύτερο συνδυασμό θα γίνει η τελική σύγκριση των αλγορίθμων.

8.1.1 Σύγκριση παραμέτρων

Καθώς η μέθοδος επίλυσης του προβλήματος αφήνει το περιθώριο παραμετροποίησης, δοκιμάστηκε μια πληθώρα συνδυασμών έτσι ώστε να βρεθεί ο αποδοτικότερος. Οι παράμετροι που χρησιμοποιήθηκαν είναι οι παρακάτω

- Ευρετήριο
    - GroupAuthor
        - Stemmed
        - Not Stemmed
    - Simple
        - Stemmed
        - Not Stemmed
- Βαθμολόγηση
    - TC
    - TFIDF

Τα ευρετήρια GroupAuthor και Simple παρά την δομική διαφορά που έχουν δίνουν εξίσου καλά αποτελέσματα για την μέθοδο με τον Set Cover. Αυτό σημαίνει ότι η μέθοδος λειτουργεί το ίδιο καλά είτε με την ύπαρξη ενός κειμένου(ψευδοκειμένου) που περιγράφει τον ειδικό είτε με την ύπαρξη πολλών κειμένων. Σε μία πραγματική εφαρμογή όπου έστω υπάρχει μια βάση δεδομένων με συγγράφεις και τους τίτλους των δημοσιεύσεων τους δεν είναι απαραίτητο το επιπλέον βήμα της συγκέντρωσης πληροφορίας ανά συγγραφέα σε ένα ψευδοκείμενο. Η διαδικασία αυτή είναι χρονοβόρα και επαναλαμβανόμενη και σίγουρα μη αποδοτική σε ένα περιβάλλον που συνεχώς θα προστίθενται ή θα αφαιρούνται δεδομένα. Ο Cosine Similarity από τον ορισμό του χρειάζεται ολόκληρο το κείμενο που περιγράφει τον συγγραφέα έτσι ώστε να λειτουργήσει. Δηλαδή για να κατασκευάσει το διάνυσμα λέξεων με το οποίο θα συγκριθεί το αντίστοιχο διάνυσμα των όρων αναζήτησης μπορεί να χρησιμοποιήσει μόνο το ευρετήριο GroupAuthor. Άρα για να γίνει η σύγκριση μεταξύ Cosine Similarity και Set Cover επί ίσοις όροις για την παράμετρο του ευρετηρίου επιλέγεται το GroupAuthor.

Η επόμενη παράμετρος αφορά την επιλογή της αποκοπής ή όχι των καταλήξεων από τις λέξεις του ευρετηρίου και της αναζήτησης. Το word stemming δίνει ένα προβάδισμα στις μεθόδους που το χρησιμοποιούν κάποιες φορές μικρό και κάποιες μεγάλο αλλά στην πλειοψηφία τους η χρήση stemmed ευρετηρίου δίνει καλύτερα αποτελέσματα. Παρόλα αυτά η παράμετρος εξαρτάται από τον τομέα που χρησιμοποιείται η εφαρμογή, δηλαδή εάν ο χρήστης θεωρεί σημαντικό να κρατήσει τις καταλήξεις των λέξεων για να κάνει αναζήτηση. Στην παρούσα περίπτωση επιλέγεται η αποκοπή των καταλήξεων καθώς οι πειραματικές μετρήσεις είναι καλύτερες με την επιλογή αυτή.

Τελευταία διαφοροποίηση αποτελεί ο τρόπος βαθμολόγησης των κειμένων. Δηλαδή πως εξάγεται η ένδειξη για το πόσο σχετικό είναι ένα κείμενο με μια λέξη κλειδί από την αναζήτηση. Η βαθμολόγηση TFIDF λειτουργεί σε γενικές γραμμές καλά και πιο συγκεκριμένα όταν υπάρχει μεγάλο σύνολο λέξεων αναζήτησης. Αυτό συμβαίνει λόγω του παράγοντα Inverse term frequency ο οποίος δίνει μεγαλύτερη σημασία σε λέξεις που είναι σπανιότερες μέσα σε ένα κείμενο. Ουσιαστικά επειδή αυτή η μέθοδος εξαρτάται και από το μέγεθος του ψευδοκειμένου είναι ασταθής στα αποτελέσματα που εξάγει καθώς στο ευρετήριο που χρησιμοποιείται το μέγεθος των κειμένων δεν είναι προκαθορισμένο. Στα μικρά σύνολα, δηλαδή αναζήτηση με 5-10 λέξεις η TFIDF αποτυγχάνει με μεγάλη διαφορά από την έτερη TC. Η βαθμολόγηση TC καθώς μετρά μόνο το πλήθος των εμφανίσεων ενός όρου και είναι ανεξάρτητη από το μέγεθος του ψευδοκειμένου δίνει καλά αποτελέσματα σε όλες τις περιπτώσεις.

8.1.2 Σύγκριση αλγορίθμων

Οι αλγόριθμοι οι οποίοι συγκρίνονται είναι οι παρακάτω

- Cosine Similarity - CS
- Greedy Set Cover - GSC
- Weighted Greedy Set Cover - WGSC
- Custom Greedy Set Cover - CGSC
- Weight Custom Greedy Set Cover - WCGSC

Για να επιτευχθεί αξιόπιστη σύγκριση, επιλέχθηκε βάσει των μετρήσεων ο καλύτερος δυνατός συνδυασμός όλων των παραμέτρων.

- Ευρετήριο: GroupAuthor & Stemmed
- Βαθμολόγηση: TC

Έχοντας ως μέτρο σύγκρισης το goodness, το πρώτο εμφανές συμπέρασμα είναι ότι ο Weighted Set Cover (και η Custom Weighted έκδοση) δεν αποδίδει καλά και έχει μεγάλες διακυμάνσεις. Σε κάποιες (λίγες) μετρήσεις είναι καλύτερος από την non weighted έκδοση αλλά υπάρχουν περιπτώσεις με μηδέν Goodness. Αυτή μεγάλη διαφορά που υπάρχει ανάμεσα σε όλο το φάσμα των μετρήσεων κάνει αναξιόπιστο τον αλγόριθμο.

Από την άλλη ο Greedy Set Cover χωρίς βάρη (και η Custom έκδοση) λειτουργεί καλά σε όλες τις περιπτώσεις και ειδικά με βαθμολόγηση TC. Επειδή όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο η custom αλγόριθμοι δεν είναι βέλτιστοι εξ' ορισμού θα θεωρηθεί ότι μόνο ο Greedy Set Cover θα συγκριθεί με τον βασικό αλγόριθμο Cosine Similarity. Στις πρώτες μετρήσεις με το σύνολο των λέξεων από συνέδρια ο cosine similarity είναι αρκετά ανταγωνιστικός και υπάρχει σχετική ισορροπία ανάμεσα στις φορές που κερδίζει και χάνει από τον Greedy Set Cover με το προβάδισμα όμως πάντα στον GSC. Οι όροι αναζήτησης που δοκιμάστηκαν ήταν 50-60 (ανάλογα με το συνέδριο) ένας αριθμός πολύ μεγάλος και μη ρεαλιστικός καθώς ένας χρήστης είναι σπάνιο, ακόμα και σε μία μηχανή αναζήτησης να εισάγει τόσες πολλές λέξεις κλειδιά. Το γεγονός αυτό κατέστησε αναγκαίο έναν δεύτερο γύρο μετρήσεων αυτή την φορά με μικρότερο πλήθος λέξεων αναζήτησης, δεκαπέντε στον αριθμό. Στις μετρήσεις αυτές υπάρχει ένα ξεκάθαρο προβάδισμα του Greedy Set Cover έναντι του

Cosine Similarity σε όλες τις περιπτώσεις. Ο CS δείχνει να ευνοείται από την ύπαρξη πολλών λέξεων κλειδιών καθώς μεγαλώνει το διάνυσμα σύγκρισης. Για να επιβεβαιωθεί η παραπάνω υπόθεση ξεκίνησε ένας επιπλέον γύρος μετρήσεων, εισάγοντας αρχικά πέντε λέξεις και προσθέτοντας πέντε, κάθε φορά, μέχρι τελικά τον αριθμό τριάντα. Οι μετρήσεις αυτές έδειξαν ότι μέχρι τις 15 λέξεις ο GSC δουλεύει καλύτερα σε όλες τις περιπτώσεις (Top 20,50,100) από τον CS. Στις 20 λέξεις ο CS είναι πολύ κοντά με τον GSC για τους Top 20 ειδικούς έχοντας μία τάση να τον ξεπεράσει. Αυτή η τάση εκδηλώνεται στην μέτρηση των 25 λέξεων όπου ο cosine ξεπερνά τον GSC σε όλες τις περιπτώσεις. Για την περίπτωση των 30 λέξεων τα αποτελέσματα είναι μεικτά και αποδεικνύεται ότι παρόλο που ο CS έχει καλή βαθμολογία για πολλές λέξεις αναζήτησης, παραμένει ασταθής και δε βρίσκεται συνεχώς μπροστά από τον GSC. Από την άλλη ο GSC διατηρεί καλή βαθμολογία κάτω από όλες τις συνθήκες και παραμένει ψηλά ανεξάρτητα της εισόδου. Τα αποτέλεσμα τον καθιστούν με διαφορά σαφώς πιο αξιόπιστο και αποδοτικό από τον βασικό αλγόριθμο CS.

Ολοκληρώνοντας, για την λύση του προβλήματος της εύρεσης ειδικών η καλύτερη επιλογή σύμφωνα με τις πειραματικές μετρήσεις είναι ο αλγόριθμος Greedy Set Cover. Θα πρέπει να επισημανθεί ότι για την επίτευξη βέλτιστου αποτελέσματος είναι απαραίτητος ο κατάλληλος συνδυασμός των παραμέτρων για ευρετήριο και βαθμολόγηση.

8.2 Μελλοντικές επεκτάσεις

Στην παρούσα διπλωματική παρουσιάστηκε μια μέθοδος για την επίλυση του προβλήματος της εύρεσης ειδικών. Παρόλο που τα αποτελέσματα είναι αρκετά καλά η παραδοχή ότι τα δεδομένα που υπάρχουν στην διάθεσή μας είναι στατικά εμποδίζει την χρήση του προγράμματος σε πραγματικό περιβάλλον. Μία πιθανή επέκταση θα μπορούσε να είναι η σύνδεση με βάση δεδομένων η οποία θα εμπλουτίζεται συνεχώς από το διαδίκτυο με τις νέες δημοσιεύσεις. Τα ευρετήρια που έχουν χρησιμοποιηθεί περιέχουν μόνο τίτλους δημοσιεύσεων. Η χρήση του προλόγου αντί των τίτλων ή ακόμα και ολόκληρων των δημοσιεύσεων ενδεχομένως να δώσει ακόμη καλύτερα αποτελέσματα. Επίσης το συνέδριο TREC (http://trec.nist.gov/) παρέχει δεδομένα για δοκιμές στα οποία δοκιμάζονται από κοινού πολλές μέθοδοι για ανάκτηση δεδομένων όχι απαραίτητα μόνο για την εύρεση ειδικών.

Ένας άλλος τομέας μέσω του οποίου μπορεί να ενισχυθεί η έρευνα είναι αυτός της σημασιολογίας (Semantics). Καθώς το Lucene κάνει αναζήτηση μέσα στο ευρετήριο δεν μπορεί να διακρίνει το πλαίσιο στο οποίο χρησιμοποιείται μια λέξη. Χρησιμοποιώντας σημασιολογία η αναζήτηση στο ευρετήριο γίνεται ακριβέστερη και δικαιότερη. Ένα χρήσιμο εργαλείο για το σκοπό θα μπορούσε να είναι το wordnet (http://wordnet.princeton.edu/).Για το κομμάτι της αξιολόγησης πέρα από το goodness, θα μπορούσε να δημιουργηθεί μια μέτρηση για κάθε κείμενο με βάση το πού έχει δημοσιευτεί , από ποιόν έχει γραφτεί και πόσες παραπομπές έχει. Η μέτρηση αυτή μοιάζει με τον τρόπο που χρησιμοποιεί το Google Scholar για να βαθμολογήσει τα κείμενα που αναρτώνται εκεί (H-Index).

Τέλος μια πολύ ενδιαφέρουσα συγχώνευση θα μπορούσε να γίνει με την πρόταση των Karimzadehgan, Xiang Zhai [21] όπου είσοδος θα είναι καθαυτές οι δημοσιεύσεις προς αξιολόγηση και επιπλέον ένα διαθέσιμο σύνολο ειδικών που θα πρέπει να κατανεμηθούν ανάλογα.

8.3 Επίλογος

Στην παρούσα εργασία παρουσιάστηκε μια πρόταση για την επίλυση του προβλήματος εύρεσης ειδικών. Δόθηκε ιδιαίτερη βαρύτητα στην περίπτωση όπου ζητείται να επιστραφεί συγκεκριμένος αριθμός Κ ειδικών των οποίων η συνδυασμένη εξειδίκευση καλύπτει πλήρως την είσοδο του χρήστη. Στο πλαίσιο αυτό δημιουργήθηκε μια εφαρμογή έτσι ώστε να μπορέσει να δοκιμαστεί η μέθοδος αλλά και να πραγματοποιηθεί σύγκριση με έναν βασικό αλγόριθμο. Τα δεδομένα των ευρετηρίων αποκτήθηκαν από την βιβλιοθήκη του DBLP η οποία περιλαμβάνει όλες τις δημοσιεύσεις που αφορούν την επιστήμη των υπολογιστών. Μετά από πολλές μετρήσεις αποδείχθηκε ότι η μέθοδος με τον Greedy Set Cover που προτείνεται είναι καλύτερη σε σχέση με τον βασικό αλγόριθμο Cosine Similarity.

Κλείνοντας, θα ήθελα να σημειώσω ότι κατά την διάρκεια της εργασίας απέκτησα σημαντικά εφόδια γύρω από έναν σύγχρονο και συνεχώς αναπτυσσόμενο τομέα, αυτόν της ανάκτησης δεδομένων. Η πολύπλευρη φύση της εργασίας μου έδωσε την δυνατότητα για συνδυασμό και εφαρμογή πολλών γνώσεων που αποκτήθηκαν κατά την διάρκεια των σπουδών αλλά και πρόσφερε σημαντική εμπειρία.

Βιβλιογραφία

[1] Henning Rode,Djoerd Hiemstra Pavel Serdyukov, "Modeling Multi-step Relevance Propagation," in *CIKM*, Napa Valley, 2008.

[2] Cheng Xiang Zhai, Geneva Belford Maryam Karimzadehgan, "Multi-Aspect Expertise Matching for Review Assignment," , Napa Valley, 2008.

[3] Peter B. Checkland, *Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students*. Chichester: Wiley, 2006.

[4] Scott W. Ambler, *The Elements of UML™ 2.0 Style*.: Cambridge University Press, 2005.

[5] Prabhakar Raghavan and Hinrich Schütze Christopher D. Manning, *Introduction to Information Retrieval*, Online ed. Cambridge: Cambridge University Press, 2009.

[6] Berthier Ribeiro-Neto Ricardo Baeza-Yates, *Modern Information Retrieval*.: Addison-Wesley, 1999.

[7] Anna Huang, "Similarity Measures for Text Document Clustering," in *NZCSRSC*, Christchurch, 2008.

[8] A. P. de Vries, and I. Soborof N. Craswell, "Overview of the trec-2005 enterprise track," in *TREC*, 2006, p. 05.

[9] K. Balog, and M. de Rijke L. Azzopardi, "Language modeling approaches for enterprise tasks," in *TREC*, 2006, p. 05.

[10] L. Azzopardi, and M. de Rijke K. Balog, "Formal models for expert finding in enterprise corpora," in *SIGIR*, 2006, p. 06.

[11] B. He, V. Plachouras, and I. Ounis. C. Macdonald, "University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier," in *TREC*, 2006, p. 05.

[12] W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma Y. Fu, "Thuir at trec 2005: Enterprise track," in *TREC*, 2006, p. 05.

[13] J. Liu, S. Bao, and H. Li Y. Cao, "Research on expert search at enterprise track of trec2005," in *TREC*, 2006, p. 05.

[14] Hui Fang and ChengXiang Zhai, "Probabilistic Models for Expert Finding," in *ECIR*, 2007, pp. 418-430.

[15] J. Lafferty and C. Zhai, "Probabilistic relevance models based on document and query generation," in *Language Modeling and Information Retrieval, Kluwer International Series on Information Retrieval*, 2003, p. 13.

[16] Maarten de Rijke Krisztian Balog, "Finding Experts and their Details in E-mail Corpora," in *WWW*, Edinburgh, 2006.

[17] Euripides G.M. Petrakis, Angelos Hliaoutakis Rena Peraki, "An Information Retrieval System for Expert and Consumer Users," , Larnaca, 2012.

[18] Krisztian Balog, Daan Odijk Edgar Meij, "Entity linking and retrieval for semantic search," in *WSDM*, New York, 2014, pp. 683-684.

[19] Dan Roth, Yuancheng Tu Nikhil Johri, "Experts' Retrieval with Multiword-Enhanced Author Topic Model," , Los Angeles, 2010.

[20] Maar ten de Rijke, Krisztian Balog, "Non-Local Evidence for Expert Finding," in *CIKM*, Napa Valley, 2008.

[21] Cheng Xiang Zhai Maryam Karimzadehgan, "Constrained Multi-Aspect Expertise Matching for Committee Review Assignment," , Hong Kong, 2009.

[22] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 256–278, 1974.

[23] Riccardo Silvestrib, Luca Trevisanc Pierluigi Crescenzia, "On Weighted vs Unweighted Versions of Combinatorial Optimization Problems," *Information and Computation*, vol. 167, no. 1, pp. 10-26, May 2001.

[24] Jens Vygen Bernhard Korte, *Combinatorial Optimization*, 5th ed. New York: Springer, 2008.

[25] Uriel Fiege, "A Threshold of ln n for Approximating Set Cover," *ACM*, 1998.

[26] Apache. http://lucene.apache.org.

[27] V. CHVATAL, "A greedy heyristic for the set-covering problem," *Mathematics of operations Research*, vol. 4, no. 3, pp. 233-235, August 1979.

[28] Anita Mirijamdotter, Andrew Basden Birgitta Bergvall-Kåreborn, "Basic Principles of SSM Modeling: An Examination of CATWOE from a Soft Perspective," *Systemic Practice and Action Research* , vol. 17, no. 2, pp. 55-73, April 2004.

[29] Vegard B. Havdal Else Lervik, *Java the UML Way*.: Wiley, 2002.

[30] Ariadne. UML Applied. [Online]. http://pesona.mmu.edu.my/~wruslan/SE3/Readings/detail/UML-Applied-Second-Edition.pdf

[31] Sartaj Sahni, *Data Structures, Algorithms, and Applications in C++*.: McGraw-Hill, 1998.

[32] Object Refinery Limited. jfree. [Online]. http://www.jfree.org/jfreechart/

[33] Sergei Vassilivtsky Ravi Kumar. (2010) Generalized distances between rankings. [Online]. http://theory.stanford.edu/~sergei/slides/www10-metrics.pdf

[34] sqlite. sqlite. [Online]. http://www.sqlite.org/

[35] DBLP Team. DBLP. [Online]. http://www.informatik.uni-trier.de/~ley/db/

[36] D. Petkova and W. B. Croft, "Umass notebook 2006: Enterprise track," in *TREC*, 2007, p. 06.

[37] Irwin King, and Michael R. Lyu Hongbo Deng, "Enhanced Models for Expertise Retrieval Using Community-Aware Strategies," , 2012.

Παράρτημα

Α. Λέξεις αναζήτησης

Παρατίθεται το σύνολο των λέξεων κλειδιών για τις δοκιμές που πραγματοποιήθηκαν.

Σύνολο SIGMOD

Aggregation, Analytics, Authenticated Query Processing , Benchmarking , Cloud Computing, cloud data management, Clustering, Complex Event Processing, Data analytics, Data cleaning , Data Integration, Data mining, Data models , Data privacy, Data provenance, Data security, Data streams, Data visualization, Data Warehouses, Database Models, Database monitoring, Database tuning, Distributed databases, graph databases, Indexing, Information extraction, Information retrieval, Keyword Search, knowledge discovery, MapReduce, Mobile databases, Multimedia Databases, OLAP, parallel databases, performance evaluation, physical database design, query languages, Query optimization, Query processing, Schema Matching, Scientific databases, semantics , Semi-structured data, sensor networks, Service-oriented computing, Social networks, Spatial Databases, Storage, Storage systems, Temporal Databases, Text Databases, text mining, Transaction management, Uncertainty, XML

Σύνολο ECIR

ad targeting, Advertising, Authority, Blog search, browsing, Building test collections, categorisation, clustering, collaborative filtering, Content-based filtering, Cross-language retrieval, Crowdsourcing for evaluation, data fusion, Desktop search, Digital libraries, Distributed IR, Enterprise Search, evaluation , Evaluation methods and metrics, Experimental design, filtering and indexing, Fusion,Combination, Genomic IR, Image and video retrieval, Interactive IR, Intranet search, legal IR, Link analysis, meta-searching, Mobile IR, Multilingual retrieval, novelty detection, online-community search, Opinion mining, patent search , Personalised, query expansion, Query log analysis, Query reformulation, Query representation, Question answering, radio retrieval , Ranking, recommender systems, Relevance feedback, Reputation, search results or content, Searching, Social Tagging , Spam detection, Spam filtering, Speech retrieval, Summarization , Task-based IR, Text Categorisation, Text clustering, Text data mining, Topic detection and tracking, User interfaces, User models, User studies, User-oriented

Το σύνολο των τυχαίων 15 λέξεων

Σύνολο SIGMOD

Analytics, Complex Event Processing, Data analytics, Data mining, Data models , Data Warehouses, Information extraction, OLAP, parallel databases, Service-oriented computing, Temporal Databases, Social networks, Spatial Databases, Storage systems, semantics

Σύνολο ECIR

Advertising, clustering, Crowdsourcing for evaluation, data fusion, Genomic IR, Intranet search, meta-searching, Multilingual retrieval, Opinion mining, patent search , Query reformulation, Question answering, Relevance feedback, Social Tagging , User models

Όροι με αύξουσα σειρά

5

Aggregation,Clustering,Distributed databases,Storage systems,XML

10

Data Mining,Distributed databases,graph databases,Indexing,Information extraction,OLAP,

query languages,Query processing,semantics ,Social networks

15

Analytics,Complex Event Processing,Data analytics,Data mining,Data models, Data Warehouses,Information extraction,OLAP,parallel databases,Service-oriented computing, Temporal Databases,Social networks,Spatial Databases,Storage systems,semantics

20

Aggregation,Analytics,Authenticated Query Processing,cloud data management,Complex Event Processing,Data Mining,Data privacy,Data visualization,Data Warehouses,Distributed databases,graph databases,MapReduce,Multimedia Databases,parallel databases,physical database design,query languages,Spatial Databases,Temporal Databases,Transaction management, Uncertainty

25

cloud data management,Clustering,Complex Event Processing,Data analytics,Data models ,Database Models,Distributed databases,Indexing,Information extraction, Keyword Search,knowledge discovery,Mobile databases,Multimedia Databases,OLAP,performance evaluation,Query optimization,Schema Matching,Scientific databases,semantics,Semi-structured data,sensor networks,text mining,Transaction management,Uncertainty,XML

30

performance evaluation,Aggregation,Cloud Computing,cloud data management,Clustering, Complex Event Processing,Data cleaning,Data models,Data provenance,Data visualization, Database Models,Database monitoring,Database tuning,OLAP,parallel databases,physical database design, Query optimization,Schema Matching, Scientific databases,semantics, Semi-structured data, sensor networks, Service-oriented computing, Spatial Databases,Storage,Storage systems,text mining,Transaction management,Uncertainty,XML

Β. Προσθήκη αλγορίθμου

Ο προγραμματιστής έχει την δυνατότητα να αλλάξει την υπάρχουσα ροή και να κατευθύνει τον κώδικα του οπουδήποτε χωρίς να δεσμεύεται από την υπάρχουσα οργάνωση σε κλάσεις και πακέτα. Στις οδηγίες που ακολουθούν θεωρείται ότι ο νέος αλγόριθμος που θα προστεθεί ακολουθεί στο σκεπτικό των ήδη εφαρμοσμένων αλγορίθμων.

Έστω ότι θέλουμε να προσθέσουμε έναν νέο αλγόριθμο EXAMPLE ο οποίος θα επιλέγει τους πρώτους Κ συγγραφείς με τους περισσότερους όρους εξειδίκευσης από την αναζήτηση.

Το πρώτο βήμα είναι η δήλωση του αλγορίθμου στην κλάση

```
public enum AlgorithmType {
    COSINE,
    GREEDY_SET_COVER,
    GREEDY_SET_COVER_CUSTOM,
    GREEDY_SET_COVER_WEIGHTED,
    GREEDY_SET_COVER_WEIGHTED_CUSTOM,
    COSINE_S,
    GREEDY_SET_COVER_S,
    GREEDY_SET_COVER_S_CUSTOM,
    GREEDY_SET_COVER_WEIGHTED_S,
    GREEDY_SET_COVER_WEIGHTED_S_CUSTOM,
    EXAMPLE
}
```

Εικόνα 37 com.fragos.experts.reloaded.utils. AlgorithmType

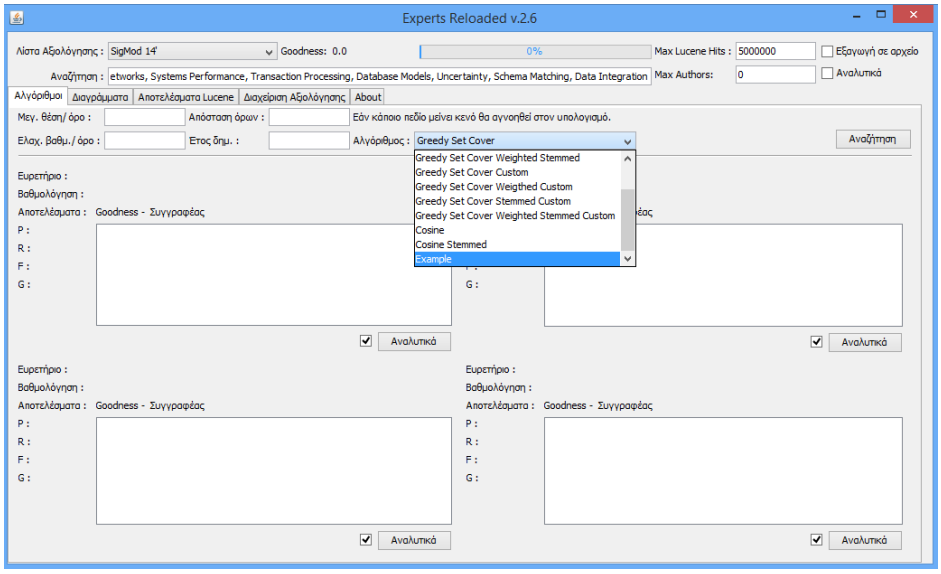
Εάν ο προγραμματιστής επιθυμεί να χρησιμοποιήσει το υπάρχον πακέτο UI για την γραφική διεπαφή του, τότε θα πρέπει να προσθέσει τον αλγόριθμο και στο Combox της κύριας οθόνης.

```
private void fillAlgorithms(){
    cmbAlgorithm.addItem("Greedy Set Cover");
    cmbAlgorithm.addItem("Greedy Set Cover Weigthed");
    cmbAlgorithm.addItem("Greedy Set Cover Stemmed");
    cmbAlgorithm.addItem("Greedy Set Cover Weighted Stemmed");
    cmbAlgorithm.addItem("Greedy Set Cover Custom");
    cmbAlgorithm.addItem("Greedy Set Cover Weigthed Custom");
    cmbAlgorithm.addItem("Greedy Set Cover Stemmed Custom");
    cmbAlgorithm.addItem("Greedy Set Cover Weighted Stemmed Custom");
    cmbAlgorithm.addItem("Cosine");
    cmbAlgorithm.addItem("Cosine Stemmed");
    cmbAlgorithm.addItem("Example");
}
```

Εικόνα 38 fill Algorithms

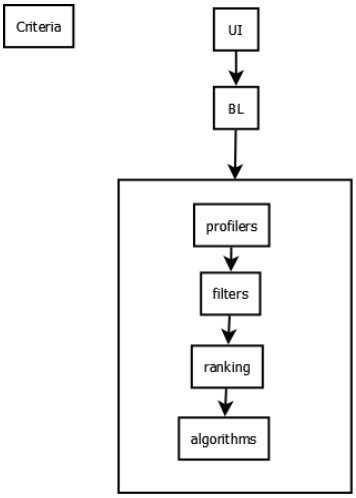
```
private AlgorithmType getSelectedAlgorithm(){
    String algorithmName = (String) cmbAlgorithm.getSelectedItem();
    AlgorithmType algorithm = null;

    switch(algorithmName){
        case  "Greedy Set Cover":
            algorithm = AlgorithmType.GREEDY_SET_COVER;
            break;
        case  "Greedy Set Cover Weigthed":
            algorithm = AlgorithmType.GREEDY_SET_COVER_WEIGHTED;
            break;
        case  "Greedy Set Cover Stemmed":
            algorithm = AlgorithmType.GREEDY_SET_COVER_S;
            break;
        case  "Greedy Set Cover Weighted Stemmed":
            algorithm = AlgorithmType.GREEDY_SET_COVER_WEIGHTED_S;
            break;
        case "Greedy Set Cover Custom":
            algorithm = AlgorithmType.GREEDY_SET_COVER_CUSTOM;
            break;
        case "Greedy Set Cover Weigthed Custom":
            algorithm = AlgorithmType.GREEDY_SET_COVER_WEIGHTED_CUSTOM;
            break;
        case "Greedy Set Cover Stemmed Custom":
            algorithm = AlgorithmType.GREEDY_SET_COVER_S_CUSTOM;
            break;
        case "Greedy Set Cover Weighted Stemmed Custom":
            algorithm = AlgorithmType.GREEDY_SET_COVER_WEIGHTED_S_CUSTOM;
        case "Cosine":
            algorithm = AlgorithmType.COSINE;
            break;
        case "Cosine Stemmed":
            algorithm = AlgorithmType.COSINE_S;
            break;
        case "Example":
            algorithm = AlgorithmType.EXAMPLE;
            break;
    }
    return algorithm;
```

Εικόνα 39 Get selected algorithm



Εικόνα 40 Κεντρική οθόνη

Η βασική ιδέα της είναι να υπάρχει ένα κεντρικό αντικείμενο (com.fragos.experts.reloaded.model.Criteria) το οποίο περιέχει όλες τις δυνατές παραμέτρους για την εκτέλεση ενός αλγορίθμου και αυτό το αντικείμενο αφού δημιουργηθεί να περνάει μέσα από όλα τα πακέτα.



Εικόνα 41 Σειρά κλήσεων πακέτων

Όταν ο χρήστης πατήσει το κουμπί ‘Αναζήτηση’ γίνεται ένας αρχικός έλεγχος των επιλογών που έχουν εισαχθεί και στην συνέχεια δημιουργείται το αντικείμενο Criteria.

```
private void btnSearchGSCActionPerformed(java.awt.event.ActionEvent evt) {
    int retainPostision;
    int distance;
    int year;
    float retainScore;
    int maxAuthors;

    retainPostision = validateIntField(txtRetainTopKPositionGSC, "Πρέπει να εισάγετε έναν ακέραιο στο πεδίο Μέγιστη θέση/όρο");
    distance = validateIntField(txtDistanceGSC,  "Πρέπει να εισάγετε έναν ακέραιο στο πεδίο Απόσταση όρων");
    year = validateIntField(txtYearGSC, "Πρέπει να εισάγετε έναν ακέραιο στο πεδίο Έτος δημοσίευσης");
    retainScore = validateFloatField(txtRetainTopKScoreGSC, "Πρέπει να εισάγετε έναν δεκαδικό στο πεδίο Ελάχιστη βαθμολογία/όρο");
    maxAuthors = validateIntField(txtMaxAuthors, "Πρέπει να εισάγετε έναν ακέραιο");

    Criteria criteria = new Criteria();
    criteria.setShowDetails(chkShowDetails.isSelected());
    criteria.setAlgorithm(getSelectedAlgorithm());
    criteria.setRetainTopKPosition(retainPostision);
    criteria.setRetainTopKScore(retainScore);
    criteria.setTermDistance(distance);
    criteria.setPublicationYear(year);
    criteria.setMaxAuthors(maxAuthors);
    search(criteria);
```

Εικόνα 42 Κλήση συνάρτησης αναζήτησης

```
private void search(Criteria criteria){
    criteria.setHitsPerPage(Integer.parseInt(txtHits.getText()));
    searchTerms = txtSearch.getText();
    criteria.setSearchTerms(searchTerms);

    task = new Task(criteria);
    task.addPropertyChangeListener(this);
    task.execute();
    //task.doInBackground();

}
```

Εικόνα 43 Συνάρτηση αναζήτησης

Επειδή η διεπαφή χρησιμοποιεί μια μπάρα προόδου ήταν απαραίτητη η δημιουργία μιας κλάσης η οποία κληρονομεί την SwingWorker.

```
private void executeProcess(){
    SetupCriteria sc = new SetupCriteria();

    //--------------------------------------------------------------------------
    setProgress(12);

    if(chkGSC2.isSelected()){
        sc.groupAuthor_TFIDF(criteria);
        runTask("Greedy",lblIndexGSC2,"GroupAuthor",lblLuceneSimilarityGSC2,"TFIDF",
                listResGSC2, "_GroupAuthorTFIDF.txt",lblPrecisionGSC2,lblRecallGSC2,
                lblFMeasureGSC2,lblGoodnessGSC2,criteria,2);
    }
    else{
        ExpertResultModel empty = new ExpertResultModel();
        expertsResultsHolder.add(empty);
    }
    //--------------------------------------------------------------------------
    setProgress(24);

    if(chkGSC3.isSelected()){
        sc.simple_TFIDF(criteria);
        runTask("Greedy",lblIndexGSC3,"Simple",lblLuceneSimilarityGSC3,"TFIDF",
                listResGSC3, "_SimpleTFIDF.txt",lblPrecisionGSC3,lblRecallGSC3,
                lblFMeasureGSC3,lblGoodnessGSC3,criteria,3);
    }
    else{
        ExpertResultModel empty = new ExpertResultModel();
        expertsResultsHolder.add(empty);
    }
```

Εικόνα 44 Συνάρτηση Swing Worker

Στην κλάση SetupCriteria έχει προ αποθηκευμένες κάποιες ρυθμίσεις οι οποίες σχετίζονται

Με τις τέσσερις διαφοροποιήσεις που έχουν υλοποιηθεί.

Stemming : Stemmed , Not Stemmed

Index : Simple, Group Author

```
public class SetupCriteria {

    public void groupAuthor_TFIDF(Criteria criteria){
        criteria.setIndexType(IndexType.GROUP_AUTHOR);
        criteria.setOutputFilename(Param.userDirectory+"\\Indexes\\GroupAuthor");
        criteria.setSimilarityType(SimilarityType.TFIDF);
        criteria.setAlgorithm(criteria.getAlgorithm());
        fixStemmed(criteria);
        fixWeighted(criteria);
    }

    public void simple_TFIDF(Criteria criteria){
        criteria.setIndexType(IndexType.SIMPLE);
        criteria.setOutputFilename(Param.userDirectory+"\\Indexes\\Simple");
        criteria.setSimilarityType(SimilarityType.TFIDF);
        criteria.setAlgorithm(criteria.getAlgorithm());
        fixStemmed(criteria);
        fixWeighted(criteria);
    }
```

Εικόνα 45 Setup Criteria

Τα σημεία του Swing Worker και του Setup Criteria δεν χρειάζεται να πειραχτούν εκτός και αν ο προγραμματιστής επιθυμεί να αλλάξει το γραφικό περιβάλλον καθώς ο κώδικας είναι συνδεδεμένος με τις 4 διαφοροποιήσεις που παρέχονται.

Η εκκίνηση της λογικής του προγράμματος ξεκινάει με την κλήση στην συνάρτηση

expertsBL.search(criteria)

```
private void runTask(String algorithm,JLabel lblIndex,String lblIndexText,JLabel lblLuceneSimilarity,
                     String lblLuceneSimilarityText,JList listResults,String postFix,
                     JLabel lblPrecision, JLabel lblRecall, JLabel lblFMeasure,JLabel lblGoodness, Criteria criteria, int y){
    List<Author> al;
    float goodness = 0.0f;

    expertRM = expertsBL.search(criteria);
    lblIndex.setText(lblIndexText);
    lblLuceneSimilarity.setText(lblLuceneSimilarityText);
    al = expertRM.getFinalResult();
    DefaultListModel dlm = new DefaultListModel();
    Metrics metrics;

    authorChangeIndex2 = 0;
```

Εικόνα 46 Run Task

```
@Override
public ExpertResultModel search(Criteria criteria) {
    ExpertResultModel em = null;
    Search_GSC gsc = new Search_GSC();
    Search_Cosine cosine = new Search_Cosine();
    Search_Example example = new Search_Example();

    switch(criteria.getAlgorithm())
    {
        case GREEDY_SET_COVER:
        case GREEDY_SET_COVER_CUSTOM:
        case GREEDY_SET_COVER_S:
        case GREEDY_SET_COVER_S_CUSTOM:
        case GREEDY_SET_COVER_WEIGHTED:
        case GREEDY_SET_COVER_WEIGHTED_CUSTOM:
        case GREEDY_SET_COVER_WEIGHTED_S:
        case GREEDY_SET_COVER_WEIGHTED_S_CUSTOM:
            em = gsc.search(criteria);
            break;
        case COSINE:
        case COSINE_S:
            em = cosine.search(criteria);
            break;
        case EXAMPLE:
            em = example.search(criteria);
        default:
    }
    return em;
}
```

Εικόνα 47 Search (Στο πακέτο BL)

Σύμφωνα με την στρατηγική που ακολουθήθηκε στην διπλωματική η διαδικασία επίλυσης χωρίζεται σε στάδια. Αρχικά γίνεται αναζήτηση στο ευρετήριο Lucene που έχει δημιουργηθεί και στην συνέχεια τα αποτελέσματα περνούν από φίλτρα τα οποία έχει προσθέσει ο χρήστης. Μετά από την διαδικασία του φιλτραρίσματος γίνεται κατάταξη σύμφωνα με ένα κριτήριο βαθμολόγησης και στο τέλος εφαρμόζεται ο αλγόριθμος.

Για τους σκοπούς του παραδείγματος θα αγνοήσουμε το στάδιο των περιορισμών και θα θεωρήσουμε ότι ο αλγόριθμος επιστρέφει τους K συγγραφείς με το υψηλότερο score που δίνει το Lucene.

```
*/
public class Search_Example {

    ExpertResultModel search(Criteria criteria){
        ExpertResultModel erm = new ExpertResultModel();

        //Προφίλ

        //Περιορισμοί

        //Κατάταξη

        //Αλγόριθμος

        return erm;
    }
}
```

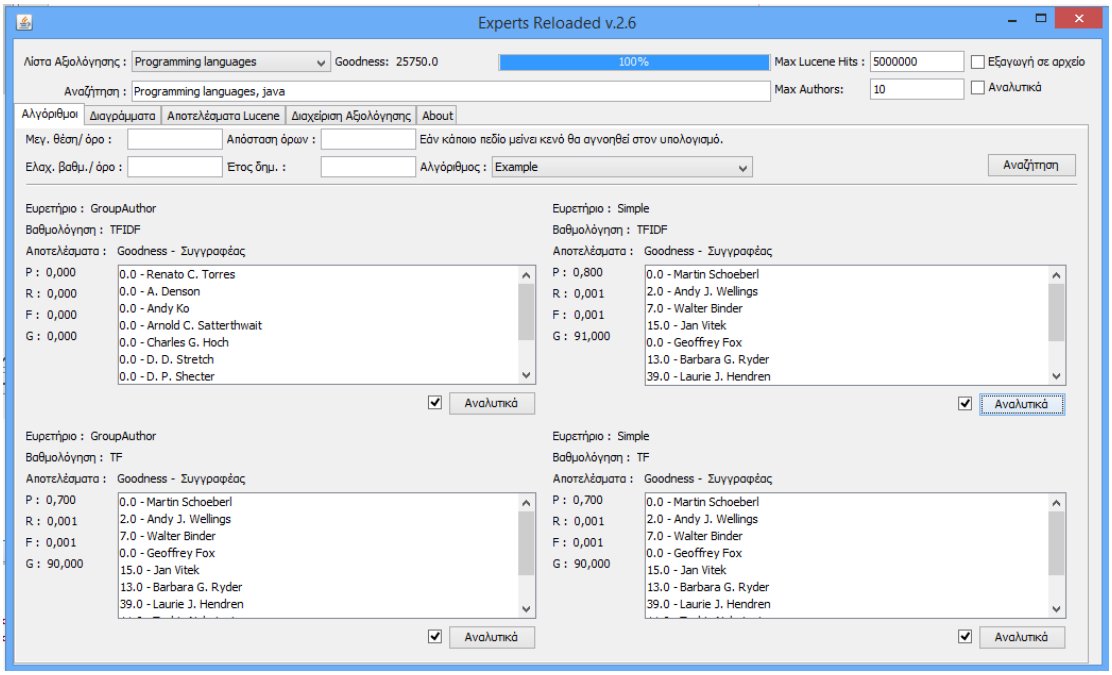
Εικόνα 48 Search Example

Ο κώδικας παρατίθεται παρακάτω

```
*/
public class Search_Example {

    ProfileGenerator cpg = new ProfileGenerator();
    Rank_Example topK = new Rank_Example();

    ExpertResultModel search(Criteria criteria){
        ExpertResultModel erm = new ExpertResultModel();

        //Προφίλ
        cpg.setErm(erm);
        cpg.createProfiles(criteria);
        erm = cpg.getErm();
        //Περιορισμοί
        //--
        //Κατάταξη
        topK.setErm(erm);
        topK.rank(criteria, cpg.getAuthorHashMap());
        erm = topK.getErm();
        //Αλγόριθμος
        PriorityQueue<Author> queue;
        queue = erm.getTopKScore();

        List<Author> result = new ArrayList<>();
        int count = 0;

        while(!queue.isEmpty()){
            Author author = queue.poll();
            result.add(author);
            count++;
            if(criteria.getMaxAuthors() == count){
                break;
            }
        }
        erm.setFinalResult(result);
        return erm;
    }
}
```

Εικόνα 49 Search Example 2

Θα πρέπει να σημειωθεί ότι στο τελευταίο βήμα του αλγορίθμου θα πρέπει να αναθέσουμε στο αντικείμενο ExpertResultModel το Final Result το οποίο είναι μία λίστα αντικειμένων Author.



Εικόνα 50 Αποτέλεσμα ενδεικτικού αλγορίθμου

Γ. Προγράμματα

Προγραμματισμός σε Java: Netbeans 8

Βάση δεδομένων : Ms SQL server 2012, MySQL server, Base X

Ευρετηριοποίηση : Apache Lucene

Σχεδιασμός διαγραμμάτων : Visual paradigm, DIA

Σύνταξη κειμένου : Word 2010