Κεφάλαιο 4. Latent Semantic Indexing (Λανθάνουσα Σημασιολογική Δεικτοδότηση)

4.1 Τι είναι Latent Semantic Indexing

Στο κεφάλαιο 3 ασχοληθήκαμε με το μοντέλο διανυσματικού χώρου (*vector space model*). Στο συγκεκριμένο κεφάλαιο θα ασχοληθούμε με μία προέκταση του μοντέλου αυτού, στην οποία οι εξαρτήσεις μεταξύ των όρων λαμβάνονται ρητά στην αναπαράσταση και αξιοποιούνται στην ανάκτηση. Η προέκταση αυτή είναι γνωστή ως *Latent Semantic Indexing* (*LSI*) και σε αυτή μοντελοποιούνται ταυτόχρονα όλες οι αλληλεξαρτήσεις μεταξύ των όρων και των εγγράφων. Υποτίθεται ότι υπάρχει μία «λανθάνουσα»(*latent*) δομή στο υπόδειγμα χρήσης λέξεων σε διάφορα έγγραφα, και γίνεται χρήση στατιστικών τεχνικών για να εκτιμηθεί αυτή. Μία περιγραφή των όρων, των εγγράφων και αναζητήσεων του χρήστη, βασισμένη στην υποκείμενη «λανθάνουσα σημασιολογική»(*latent semantic*) δομή (παρά την επιφανειακή δομή λέξεων) χρησιμοποιείται για την αναπαράσταση και την ανάκτηση πληροφορίας. Ένα πλεονέκτημα της *LSI* αναπαράστασης είναι ότι μία αναζήτηση του χρήστη μπορεί να είναι όμοια με ένα έγγραφο ακόμα και αν δεν έχουν κοινές λέξεις.

Το *Latent Semantic Indexing* (*LSI*) χρησιμοποιεί την διάσπαση σε ιδιάζουσες τιμές (*SVD*) προκειμένου να μοντελοποιήσει τις συνδυαζόμενες σχέσεις. Η *SVD* είναι πολύ συγγενική με την διάσπαση σε ιδιοδιανύσματα και την ανάλυση παραγόντων [CW85]. Ένας μεγάλος πίνακας όρων-εγγράφων αναλύεται σε ένα σύνολο από k, τυπικά από 100 ως 300, ορθοκανονικούς παράγοντες από τους οποίους ο αρχικός πίνακας μπορεί να προσεγγιστεί με γραμμικό συνδυασμό. Αντί να αναπαρίστανται τα έγγραφα και οι αιτήσεις των χρηστών απευθείας σαν σύνολα ανεξάρτητων λέξεων, το *LSI* τα αναπαριστά σαν συνεχείς τιμές σε κάθε μία από τις k ορθοκανονικές διαστάσεις δεικτοδότησης. Εφόσον ο αριθμός των παραγόντων ή των διαστάσεων είναι πολύ μικρότερος από τον αριθμό των μοναδικών όρων οι λέξεις δεν μπορεί να είναι ανεξάρτητες. Για παράδειγμα, αν δύο όροι χρησιμοποιούνται σε όμοια περιβάλλοντα (έγγραφα), θα έχουν παρόμοια διανύσματα στην αναπαράσταση. Η *SVD* μπορεί να συλλάβει καλύτερα μία τέτοια δομή από ότι οι απλές συσχετίσεις και ομαδοποιήσεις όρου με όρο και εγγράφου με έγγραφο. Το *LSI* μερικώς ξεπερνάει κάποιες από τις ελλείψεις της υπόθεσης για ανεξαρτησία των λέξεων, και παρέχει έναν αυτόματο τρόπο χειρισμού της συνωνυμίας χωρίς να υπάρχει ανάγκη ενός όχι αυτόματα κατασκευασμένου θησαυρού.

Η ανάλυση που διεξάγεται από την *SVD* μπορεί να αναπαρασταθεί και γεωμετρικά. Το αποτέλεσμα είναι ένα διάνυσμα που αναπαριστά τη θέση κάθε όρου και έγγραφου σε μία *LSI* αναπαράσταση διάστασης. Η θέση των διανυσμάτων των όρων αντανακλά τις συσχετίσεις στη χρήση τους για όλα τα έγγραφα. Σε αυτό το χώρο το συνημίτονο ή το εσωτερικό γινόμενο μεταξύ των διανυσμάτων ισοδυναμεί με την υπολογιζόμενη ομοιότητα τους. Η ανάκτηση τυπικά προχωράει χρησιμοποιώντας τους όρους σε μία αναζήτηση για να καθορίσει ένα σημείο στο χώρο, και όλα τα έγγραφα βαθμολογούνται με βάση την ομοιότητα τους με την αναζήτηση. Παρόλα αυτά εφόσον τα διανύσματα και των όρων και των εγγράφων αναπαριστούνται στον ίδιο χώρο, οι

ομοιότητες μεταξύ οποιουδήποτε συνδυασμού των όρων και των εγγράφων μπορούν εύκολα να εξαχθούν.

4.2 Αδυναμίες των Ισχύων Μεθόδων Αυτόματης Ανάκτησης

Μία βασική αδυναμία των προηγουμένων μοντέλων ανάκτησης πληροφορίας είναι ότι δεν μπορούν να αντιμετωπίσουν αποδοτικά τα φαινόμενα της συνωνυ*μίας*(*synonymy*) και της *πολυσημίας*(*polysemy*). Χρησιμοποιούμε τον όρο *συνωνυμία* για να περιγράψουμε το γεγονός ότι μπορούμε με πολλούς τρόπους να αναφερθούμε στο ίδιο αντικείμενο ή στην ίδια έννοια. Για παράδειγμα, δύο άνθρωποι διαλέγουν την ίδια λέξη για να περιγράψουν ένα αντικείμενο μόνο στο 20% των περιπτώσεων. Σχετικά μικρή συμφωνία έχει αναφερθεί στις μελέτες συνέπειας μεταξύ των συντακτών ευρετηρίου και στην γενιά όρων ψαξίματος από εκάστοτε ειδικούς μεσολαβητές, ή λιγότερο έμπειρους εξερευνητές. Η ύπαρξη της συνωνυμίας τείνει να μειώσει σημαντικά την απόδοση των συστημάτων ανάκτησης.

Επίσης πολλές φορές, λόγω του ότι η ίδια λέξη έχει περισσότερες από μια σημασίες, μπορεί να ανακτηθεί υλικό άσχετο από αυτό που ζητείται. Αυτό το γεγονός το περιγράφουμε με τον όρο *πολυσημία*. Σε διαφορετικές εκφράσεις ή σε διαφορετικούς ανθρώπους ο ίδιος όρος μπορεί να έχει ποικίλα νοήματα. Έτσι όταν ένα έγγραφο περιέχει έναν όρο αυτό δεν σημαίνει υποχρεωτικά ότι ο όρος αυτός έχει το νόημα που μας ενδιαφέρει. Η πολυσημία είναι ένας παράγοντας που προκαλεί μικρή ακρίβεια στα συστήματα.

Η αδυναμία των ισχύων μεθόδων δεικτοδότησης και ανάκτησης, να αντιμετωπίσουν αυτά τα προβλήματα οφείλεται κυρίως σε τρεις λόγους. Ο πρώτος είναι ο ατελής τρόπος με τον οποίο είναι αποθηκευμένα τα δεδομένα. Οι όροι που χρησιμοποιούνται για να περιγράψουν τα έγγραφα, είναι μόνο ένα κλάσμα των όρων τους οι χρήστες θα ψάξουν να βρουν σε αυτά. Αυτό συμβαίνει γιατί πολλές φορές τα έγγραφα δεν περιέχουν όλους τους όρους που ζητάει ο χρήστης ή γιατί οι διαδικασίες επιλογής σκόπιμα παραβλέπουν πολλούς από τους όρους σε ένα έγγραφο.

Ο δεύτερος παράγοντας είναι η έλλειψη μιας επαρκής αυτόματης μεθόδου που να αντιμετωπίζει την πολυσημία. Μια γνωστή προσπάθεια είναι η χρήση ελεγχόμενων λεξιλογίων και έμψυχοι μεσάζοντες που να δρουν ως μεταφραστές. Αυτή η λύση δεν είναι μόνο εξαιρετικά ακριβή αλλά και μη αποτελεσματική. Μια άλλη προσπάθεια είναι η χρήση Boolean διασταύρωσης και συντονισμού με άλλους όρους για αποσαφήνιση της σημασίας. Η επιτυχία εμποδίζεται από την αδυναμία των χρηστών να σκεφτούν κατάλληλους περιοριστικούς όρους, αν αυτοί υπάρχουν, και στο γεγονός ότι μερικοί οροί δεν εμφανίζονται στα έγγραφα ή δεν μπορούν να ανακτηθούν.

Ο τρίτος παράγοντας είναι περισσότερο τεχνικός και έχει σχέση με τον τρόπο με τον οποίο λειτουργούν τα συστήματα ανάκτησης. Στα συστήματα αυτά κάθε τύπος λέξης επεξεργάζεται ανεξάρτητα από τους υπόλοιπους. Έτσι το ταίριασμα (ή μη) δύο όρων οι οποίοι συνεχώς εμφανίζονται μαζί, μετράει το ίδιο με το ταίριασμα δυο όρων που σπάνια εμφανίζονται μαζί στο ίδιο έγγραφο. Έτσι η επιτυχία τόσο στη Boolean όσο και στη αναζήτηση επιπέδου συντεταγμένων, εμποδίζεται από την αδυναμία να περισσεύουν οι διαφορετικές αξίες και ως αποτέλεσμα καταστρέφει τα αποτελέσματα σε άγνωστο βαθμό. Το πρόβλημα αυτό μεγαλώνει την δυσκολία του χρήστη να χρησιμοποιήσει μία αναζήτηση σύνθετων όρων αποδοτικά ώστε να επεκτείνει ή να περιορίσει μία αναζήτηση.

4.3 Latent Semantic Indexing

Το LSI αποτελεί μια μέθοδο ανάκτησης πληροφορίας που χρησιμοποιεί μεθόδους της γραμμικής άλγεβρας για να μειώσει το μέγεθος του πίνακα εμφάνισης συχνοτήτων ώστε να είναι πιο εύκολη η επεξεργασία του. Το μοντέλο του LSI αναπτύχθηκε από τους: Deerwester, Dumais, Furnas, Landauer, Harshman το 1988 στα εργαστήρια της Bellcore ([DDFLH88].

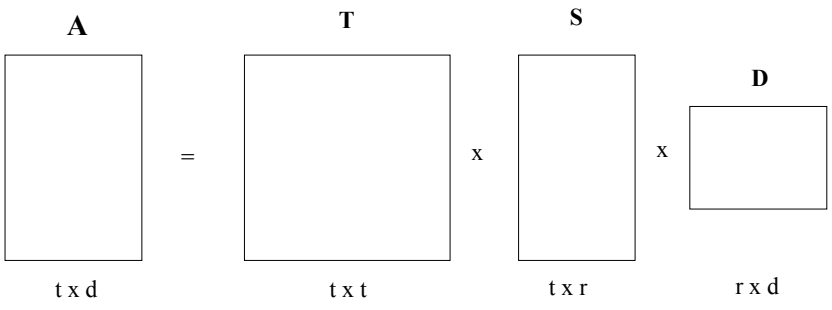
4.4 Singular Value Decomposition

Προκειμένου να εφαρμόσουμε το μοντέλο του *Latent Semantic Indexing* δημιουργούμε αρχικά έναν πίνακα όρων-κειμένων *Freq_Table* μεγέθους *t x d* , όπου *t* ο αριθμός των όρων και *d* ο αριθμός των κειμένων. Τα στοιχεία του πίνακα *Freq_Table* αποτελούν τις συχνότητες εμφάνισης κάθε όρου σε ένα συγκεκριμένο κείμενο, δηλαδή $Freq\_Table=[\alpha_{ij}]$ όπου το α_{ij} δηλώνει τη συχνότητα εμφάνισης του όρου *i* στο κείμενο *j*.

Στη συνέχεια ο πίνακας *Freq_Table* διασπάται στο γινόμενο 3 πινάκων με χρήση της μεθόδου *SVD* (singular value decomposition) ή αλλιώς "διάσπασης ιδιόμορφων τιμών" η οποία χρησιμοποιείται για την παραγοντοποίηση ιδιοτιμών-ιδιοδιανυσμάτων ενός συμμετρικού πίνακα.

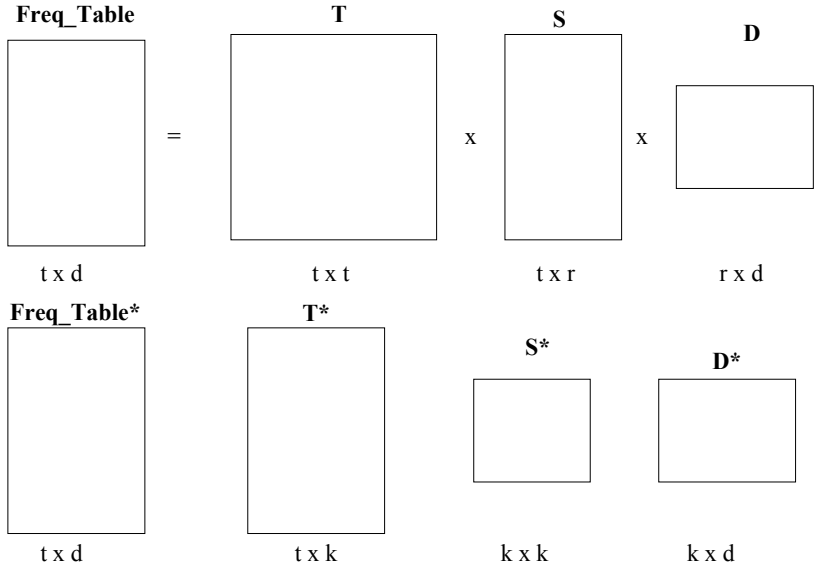
Σύμφωνα με τη μέθοδο *SVD* ένας συμμετρικός πίνακας *A* διασπάται σε γινόμενο τριών πινάκων $A=TxSxD^T$, όπου ο *T* είναι ένας ορθογώνιος πίνακας μεγέθους *t x t* και ονομάζεται πίνακας αριστερών ιδιοδιανυσμάτων, ο *S* είναι ένας διαγώνιος πίνακας μεγέθους *t x r* και τα στοιχεία της κυρίας διαγωνίου του αποτελούν τις ιδιοτιμές του *A,* ενώ ο πίνακας *D* είναι ένας ορθογώνιος πίνακας μεγέθους *r x d* και καλείται πίνακας δεξιών ιδιοδιανυσμάτων.

Ένα παράδειγμα διάσπασης πίνακα φαίνεται στο παρακάτω σχήμα.



Σχήμα 1: Διάσπαση πίνακα με τη μέθοδο SVD

Χρησιμοποιώντας τους πίνακες *T, S, D* μπορούμε να ανακατασκευάσουμε τον πίνακα *Freq_Table* χωρίς κανένα λάθος. Αυτό όμως που κάνει το *LSI* είναι να κρατά από τον πίνακα *S* τις *k* μεγαλύτερες ιδιόμορφες τιμές και να μηδενίζει τις υπόλοιπες. Έτσι δημιουργείται ένας καινούριος πίνακας S^* μεγέθους *k x k.* Σβήνοντας και τις αντίστοιχες στήλες και γραμμές των πινάκων *T* και *D* αντίστοιχα δημιουργούμε ένα νέο γινόμενο $Freq\_Table^*=T^*x\ S^*x\ D^{*T}$, όπου ο πίνακας T^* είναι μεγέθους *t x k* και ο πίνακας D^* είναι μεγέθους *k x d*. Έχουμε λοιπόν το ακόλουθο σχήμα :



Σχήμα 2: Διάσπαση του Freq_Table με SVD τεχνική και δημιουργία του Freq_Table κρατώντας τις k μεγαλύτερες ιδιοτιμές.*

Βέβαια η επιλογή του *k* αποτελεί κρίσιμο σημείο για την εφαρμογή του *LSI*, αφού θέλουμε ένα *k* τέτοιο ώστε να περιλαμβάνεται η απαραίτητη πληροφορία αλλά και να μη γίνονται λάθη κατά την ανακατασκευή του αρχικού πίνακα *Freq_Table*. Παρακάτω θα αναλύσουμε τη φυσική σημασία των πινάκων που προκύπτουν κατά την παραγοντοποίηση του πίνακα *Freq_Table*.

Προκειμένου να αποδώσουμε τη φυσική σημασία των πινάκων που προκύπτουν θα έχουμε υπόψη ότι θέλουμε να βρούμε : α) πόσο δυο όροι σχετίζονται β) πόσο σχετίζονται δυο κείμενα μεταξύ τους και γ) πώς σχετίζεται ένα κείμενο με ένα συγκεκριμένο όρο.

Σχέση μεταξύ δυο όρων.

Το εσωτερικό γινόμενο μεταξύ δύο γραμμών του πίνακα Freq_Table δείχνει κατά πόσο σχετίζονται μεταξύ τους δυο όροι, εννοώντας ότι αυτοί εμφανίζονται σε ένα σύνολο κειμένων με το ίδιο θέμα.

Πιο αναλυτικά ο πίνακας Freq_Table x Freq_Table^T είναι ένας συμμετρικός πίνακας που περιλαμβάνει όλα τα εσωτερικά γινόμενα που εκφράζουν τη σχέση μεταξύ όρων.

Σχέση μεταξύ δυο κειμένων.

Το εσωτερικό γινόμενο μεταξύ δύο στηλών του Freq_Table δείχνει κατά πόσο σχετίζονται μεταξύ τους δυο κείμενα, εννοώντας ότι σε αυτά εμφανίζονται οι ίδιοι όροι, άρα έχουν το ίδιο θέμα. Πιο αναλυτικά ο πίνακας Freq_Table^T x Freq_Table είναι ένας συμμετρικός πίνακας που περιλαμβάνει όλα τα εσωτερικά γινόμενα που εκφράζουν τη σχετικότητα μεταξύ κειμένων .

Σχέση μεταξύ κειμένου όρου.

Η συσχέτιση ανάμεσα σε ένα όρο και ένα κείμενο είναι εντελώς διαφορετική. Όπως έχουμε ήδη ορίσει κάθε στοιχείο του πίνακα Freq_Table αποτελεί τη συχνότητα εμφάνισης ενός

όρου σε ένα κείμενο. Εδώ όμως θα εκφράσουμε τον πίνακα Freq_Table με τη βοήθεια των άλλων πινάκων.

Ο πίνακας *T* αποτελείται από ορθοκανονικές στήλες οι οποίες ορίζουν ένα χώρο διανυσμάτων που εκφράζει σύνολα όρων. Επομένως κάθε κείμενο ορίζεται ως γραμμικός συνδυασμός κάποιων συνόλων όρων. Τα σύνολα αυτά είναι γραμμικά ανεξάρτητα (αφού οι στήλες είναι κάθετες μεταξύ τους) και μπορούμε να πούμε ότι το καθένα ορίζει και ένα θέμα διαφορετικό από τα άλλα.

Ο πίνακας *D* καθορίζει τη βαρύτητα που έχει κάθε θέμα μέσα στη συλλογή των κειμένων. Ο πίνακας *D* αποτελείται από ορθοκανονικές στήλες οι οποίες ορίζουν ένα χώρο διανυσμάτων που εκφράζει σύνολα κειμένων. Επομένως κάθε όρος ορίζεται ως ένας γραμμικός συνδυασμός κάποιων συνόλων κειμένων. Τα σύνολα αυτά είναι γραμμικά ανεξάρτητα και μπορούμε να πούμε ότι το καθένα ορίζει και μια διαφορετική ερμηνεία.

Εξαιτίας της ισότητας $Freq\_Table = T \times S \times D^T$ και εκμεταλευόμενοι το γεγονός ότι ο πίνακας S μπορεί να γραφεί ως γινόμενο δυο πινάκων $S^{1/2} \times S^{1/2}$ (όπου κάθε στοιχείο του πίνακα $S^{1/2}$ αποτελεί την τετραγωνική ρίζα των στοιχείων του $S^{1/2}$) γράφουμε την ισότητα : $Freq\_Table = T \times S^{1/2} \times S^{1/2} \times D^T$.

Έτσι το γινόμενο $T \times S^{1/2}$ ορίζει ένα σύνολο από θέματα τα οποία χρησιμοποιώντας τους συντελεστές του S αποκτούν διαφορετική βαρύτητα -σημασία, ενώ το γινόμενο $S^{1/2} \times D^T$ ορίζει ένα σύνολο από ερμηνείες όρων τα οποία χρησιμοποιώντας πάλι τους συντελεστές του *S* αποκτούν διαφορετική σημασία. Αφού είδαμε τη φυσική σημασία κάθε πίνακα θα δούμε πως μοντελοποιείται μια ερώτηση, query ενός χρήστη στο *LSI* και στη συνέχεια θα παρουσιάσουμε ένα παράδειγμα χρησιμοποίησης του.

4.5 Αναπαράσταση Query.

Το query ενός χρήστη αναπαρίσταται ως ένα διάνυσμα της μορφής : $q' = q^T \times T \times S^{-1}$, όπου *q* είναι το διάνυσμα των λέξεων που ο χρήστης χρησιμοποιεί για την ερώτηση του πολλαπλασιασμένες με τα αντίστοιχα βάρη όρων. Το διάνυσμα τοποθετείται στον *k*-διάστατο χώρο και συγκρίνεται με τα διανύσματα των κειμένων χρησιμοποιώντας ως συνάρτηση σχετικότητας τη συνάρτηση του συνημιτόνου ανάμεσα στο κάθε κείμενο και το *query*. Τα σχετικά κείμενα σύμφωνα με κάποιο κατώφλι επιστρέφονται στο χρήστη. Παρακάτω παραθέτουμε ένα παράδειγμα για να γίνει πιο κατανοητό το μοντέλο ανάκτησης του *Latent Semantic Indexing*.

4.6 Παράδειγμα Χρήσης του LSI.

Έστω ότι έχουμε μια συλλογή 9 κειμένων τα οποία χαρακτηρίζουμε με βάση τον τίτλο τους.

Τίτλος	
HCI_1	*Human* machine *interface* for *computer* applications
HCI_2	A *survey* of *user* opinion of *computer system response time*
HCI_3	The *user interface* of *EPS* management *system*
HCI_4	*System* and *human system* engineering testing of *EPS*
HCI_5	Relation of *user* perceived *response time* to error measurement
GR_1	The generation of random,binary,unordered *trees*
GR_2	The intersection *graph* of paths in trees
GR_3	*Graph minors:* Widths of *trees* and well-quasi ordering
GR_4	*Graph minors:*A *survey*

Πίνακας 1: Συλλογή κειμένων

Με πλάγια γράμματα χαρακτηρίζουμε στους τίτλους τις λέξεις που χρησιμοποιήθηκαν ως keywords .Στη συνέχεια με βάση αυτούς τους τίτλους κατασκευάζουμε έναν απλό πίνακα X (κειμένων-keywords), με χρήση του δυαδικού μοντέλου για τα βάρη (0 αν δεν εμφανίζεται ο όρος ,1 αν εμφανίζεται). Ο πίνακας είναι διαστάσεων 12 x 9 και είναι ο ακόλουθος :

	HCI_1	HCI_2	HCI_3	HCI_4	HCI_5	GR_1	GR_2	GR_3	GR_4
Human	1	0	0	1	0	0	0	0	0
Interface	1	0	1	0	0	0	0	0	0
Computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	1	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	0
Trees	0	0	0	0	0	1	1	0	1
Graph	0	0	0	0	0	0	1	1	0
Minors	0	0	0	0	0	0	0	1	1

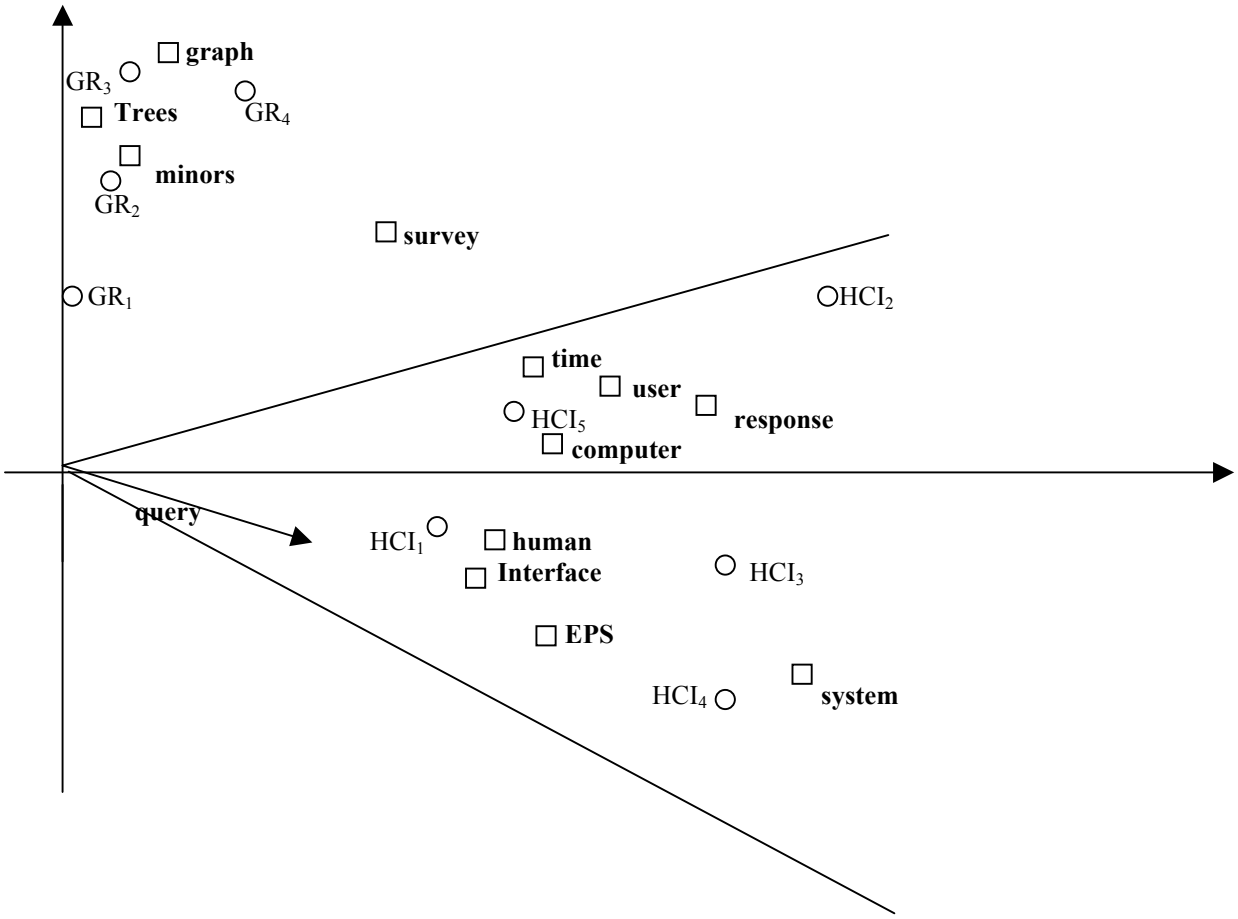
Πίνακας 2: Ο πίνακας εμφάνισης συχνοτήτων

Σε αυτή την περίπτωση βλέπουμε ότι κάθε κείμενο χαρακτηρίζεται από ένα διάνυσμα 12 διαστάσεων, με κάθε μια διάσταση να αντιστοιχεί σε ένα όρο. Επίσης βλέπουμε ότι υπάρχουν 2 κατηγορίες κειμένων στη συλλογή, μια η οποία αναφέρεται στο human-computer interaction και μια η οποία αναφέρεται σε θεωρία γράφων. Άρα θα ήταν επιθυμητό αν κάναμε μια αναζήτηση με βάση το *human computer interaction* να παίρναμε σαν απάντηση όλα τα κείμενα της κατηγορίας HCI. Βλέπουμε όμως ότι τα κείμενα HCI_3 και HCI_5 δεν θα περιέχονταν στην απάντηση του συστήματος καθώς δεν περιέχουν στην αναπαράστασή τους αυτούς τους όρους.

Αν τώρα εφαρμόσουμε τον αλγόριθμο LSI και εφαρμόσουμε SVD στον πίνακα X, κρατώντας μόνο δυο διαστάσεις (δηλαδή τις 2 πρώτες στήλες του πίνακα S) έχουμε την αναπαράσταση κειμένων και όρων στις 2 διαστάσεις που φαίνεται στο σχήμα 3.

Αντίστοιχα κάθε όρος αναπαρίσταται με τον πολλαπλασιασμό της αντίστοιχης στήλης του πίνακα *T* (που προκύπτει από τη μέθοδο *SVD*) με την ιδιόμορφη τιμή σ_1 για τη *x*-συντεταγμένη και με την δεύτερη ιδιόμορφη τιμή σ_2 για την *y*-συντεταγμένη. Έτσι τοποθετούμε τους όρους στο δισδιάστατο χώρο, (αν είχαμε κρατήσει *k* ιδιόμορφες τιμές του πίνακα *S*, τότε θα αναφερόμασταν στο *k*-διάστατο χώρο και κάθε *i*-συντεταγμένη θα υπολογιζόταν με πολλαπλασιασμό του διανύσματος-στήλης του *T* που εκφράζει κάθε όρο με την αντίστοιχη σ_i ιδιόμορφη τιμή).

Παρατηρούμε ότι τα κείμενα έχουν σαφώς διαχωριστεί στο χώρο σε δυο ομάδες, (η πρώτη ομάδα περιέχει τα κείμενα με θέμα το human computer interaction και η δεύτερη τα κείμενα με θέμα τους γράφους). Έτσι όταν θέσουμε το query "human computer interaction",αρχικά το query αυτό θα αντιστοιχηθεί σε διαστάσεις του *LSI* (με τον τρόπο που προηγουμένως αναπτύξαμε) και στη συνέχεια θα επιστραφούν όλα τα κείμενα τα οποία βρίσκονται "κοντά" σε αυτό σύμφωνα με τη συνάρτηση συνημιτόνου. Έτσι ακόμα και κείμενα τα οποία σχετίζονται με το θέμα αλλά δε μοιράζονται τα keywords θα επιστραφούν.



Σχήμα 3: Αντιστοίχηση κειμένων και όρων σε 2 διαστάσεις με χρήση του LSI

4.7 Θέματα για Συζήτηση στο LSI.

Τα αποτελέσματα πειραμάτων που έγιναν εφαρμόζοντας το LSI σε υπάρχουσες βάσεις κειμένων ήταν πολύ ενθαρρυντικά και έδειξαν ότι το LSI υπερέχει άλλων μεθόδων ανάκτησης που βασίζονται αποκλειστικά σε ταίριασμα των keywords. Όμως και η θεωρητική μελέτη του μοντέλου του LSI δείχνει ότι το LSI αντιμετωπίζει αρκετά καλά προβλήματα που εμφανίζονται στην ανάκτηση πληροφορίας.

Το LSI αντιμετωπίζει ικανοποιητικά το πρόβλημα της συνωνυμίας αφού κάθε όρος συσχετίζεται με τους άλλους και δημιουργούνται θέματα τα οποία χαρακτηρίζουν θεματικά τα κείμενα. Με αυτόν τον τρόπο επιστρέφονται κείμενα τα οποία δε μοιράζονται κανένα όρο δεικτοδότησης με το query που θέτει ένας χρήστης αλλά είναι σχετικά με αυτό.

Αντίθετα το LSI δε μπορεί να λύσει το πρόβλημα της πολυσημίας ,αφού κάθε όρος αναπαρίσταται από ένα μοναδικό σημείο στον k-διάστατο χώρο, οπότε οι διαφορετικές σημασίες μιας λέξης αντιστοιχίζονται στο ίδιο σημείο. Μάλιστα ένας όρος με πολλές σημασίες

αναπαρίσταται ως ο ζυγισμένος μέσος όρος των διαφορετικών εννοιών και όταν δεν υπάρχει ταύτιση μιας έννοιας με αυτήν που αναπαριστούμε υπάρχει σημαντικό πρόβλημα και προκύπτουν λάθος αποτελέσματα. Μια προφανής λύση σε αυτό το πρόβλημα είναι να εντοπίζουμε τις διαφορετικές σημασίες ενός όρου και να τον τοποθετούμε σε διαφορετικές περιοχές του χώρου ορίζοντας για κάθε έννοια μια υποκατηγορία. Όμως καμιά τέτοια μέθοδος δεν έχει ακόμα αναπτυχθεί.

Από όσα αναφέραμε παραπάνω το LSI αποτελεί ένα υποσχόμενο μοντέλο ανάκτησης το οποίο μάλιστα προσφέρει και μια οικονομική αναπαράσταση του πίνακα όρων-κειμένων αφού μειώνοντας τον αριθμό των διαστάσεων πετυχαίνουμε ικανοποιητική ακρίβεια στα αποτελέσματα (συνήθως ο αριθμός των διαστάσεων κυμαίνεται από 100-150). Παρόλα αυτά υπάρχουν κάποια μειονεκτήματα που καθιστούν δύσκολη τη χρήση του LSI ως γενικευμένου μοντέλου ανάκτησης.

Ένα από αυτά είναι η δυσκολία στην προσθήκη νέων κειμένων με αποτέλεσμα η αδυναμία χρήσης του μοντέλου σε δυναμικά περιβάλλοντα όπως το Διαδίκτυο. Ο πίνακας όρων -κειμένων θα πρέπει να ανανεώνεται εύκολα και γρήγορα με τις προσθήκες νέων κειμένων χωρίς να χρειάζεται να εφαρμόζεται ξανά στον ανανεωμένο πίνακα η μέθοδος SVD. Αυτή τη στιγμή υπάρχουν αλγόριθμοι που ενημερώνουν κατάλληλα τον πίνακα αν και γίνονται προσπάθειες για αποδοτικότερους τρόπους ενημέρωσης. Επίσης κρίσιμο θέμα αποτελεί και ο καθορισμός του αριθμού των διαστάσεων που προς το παρόν γίνεται μόνο έχοντας γνώση των χαρακτηριστικών της συλλογής κειμένων που έχουμε στη διάθεση μας.

Η επιλογή του αριθμού των διαστάσεων επηρεάζει την απόδοση του LSI . Η μείωση του αρχικού αριθμού διαστάσεων απομακρύνει το θόρυβο που προκαλείται από τις στατιστικές συσχετίσεις των λέξεων που εμφανίζονται στα κείμενα. Από την άλλη όμως αν μειώσουμε κατά πολύ τον αριθμό των διαστάσεων ο πίνακας που προκύπτει έχει χάσει ένα μεγάλο ποσοστό χρήσιμης πληροφορίας, με αποτέλεσμα τη μειωμένη απόδοση στα αποτελέσματα. Όπως αναφέραμε και προηγουμένως ένας αριθμός διαστάσεων ίσος με 150-200 δουλεύει σωστά και έχει ικανοποιητική ακρίβεια. Πιο αναλυτικά κατάλληλα πειράματα έδειξαν ότι η απόδοση στην ανάκτηση κειμένων αυξάνεται ως ένα σημείο καθώς αυξάνεται ο αριθμός των διαστάσεων και αγγίζει ένα μέγιστο. Στη συνέχεια αν αυξήσουμε τον αριθμό των διαστάσεων η ακρίβεια αρχίζει να μειώνεται και όταν φτάσουμε στο σημείο όπου κρατάμε τις αρχικές διαστάσεις τότε , η απόδοση του LSI ισούται με την απόδοση που έχει το διανυσματικό μοντέλο ανάκτησης. Άρα η επιλογή των διαστάσεων είναι ένα κρίσιμο θέμα για έρευνα.

Μια μέθοδος η οποία προσπαθεί να βελτιώσει την απόδοση του LSI είναι το relevant feedback. Σε αυτή τη μέθοδο, μετά την επιστροφή των αποτελεσμάτων από το query ο χρήστης εξετάζει τις απαντήσεις και ενημερώνει το σύστημα για τα αποτελέσματα, δηλαδή ποια ήταν σχετικά και ποια όχι με την ερώτηση του. Στη συνέχεια το σύστημα αλλάζει το query δίνοντας μεγαλύτερο βάρος στους όρους που περιέχονται στα κείμενα που χαρακτηρίστηκαν σχετικά με την ερώτηση και μικρότερο βάρος στους όρους που περιέχονταν στα κείμενα που ο χρήστης χαρακτήρισε ως μη σχετικά με την ερώτηση του . Ακολούθως το query επανεξετάζεται και έχουμε μια καινούρια απάντηση η οποία έχει μεγαλύτερη απόδοση από την προηγούμενη.

Όπως έχουμε ήδη αναφέρει το LSI δε μπορεί να χρησιμοποιηθεί σε δυναμικά περιβάλλοντα όπως το Internet αφού είναι πολύ δύσκολη η προσθήκη νέων κειμένων και όρων. Προκειμένου το σύστημα να έχει μεγάλη απόδοση θα πρέπει μετά την προσθήκη μικρού αριθμού κειμένων ή όρων να διασπάται ο αρχικός πίνακας και με χρήση της μεθόδου SVD να υπολογίζονται οι πίνακες T,S,D. Αυτό όμως είναι πολύ δύσκολο αφού έχει μεγάλο υπολογιστικό φόρτο και κόστος. Για αυτό το λόγο εφαρμόζονται τεχνικές ανανέωσης του μοντέλου οι οποίες όμως δίνουν ικανοποιητικά αποτελέσματα για μικρό αριθμό ενθέσεων νέων κειμένων. Σύμφωνα με αυτές τις τεχνικές κάθε νέο κείμενο ή όρος αντιμετωπίζεται ως ένα query και αναπαρίσταται στον k-διάστατο χώρο.

Πιο αναλυτικά κάθε νέο κείμενο που θέλουμε να ενθέσουμε εκφράζεται ως το ζυγισμένο άθροισμα των διανυσμάτων των όρων και τοποθετείται στο χώρο βάσει των συντεταγμένων που προκύπτουν από τη σχέση : $d^* = d^T \times D^* \times S^{*-1}$.

Αντίστοιχα κάθε νέος όρος που θέλουμε να ενθέσουμε εκφράζεται ως το ζυγισμένο άθροισμα των διανυσμάτων των κειμένων στα οποία εμφανίζεται και τοποθετείται στο χώρο βάσει των συντεταγμένων που προκύπτουν από τη σχέση: $t^* = txT^*xS^{*-1}$. Αυτός ο τρόπος ανανέωσης δίνει καλά αποτελέσματα για μικρό αριθμό ενθέσεων και στη συνέχεια πρέπει να εφαρμόσουμε την SVD τεχνική.

4.8 Εφαρμογές του LSI.

Το LSI αρχικά αναπτύχθηκε ως ένα σύστημα ανάκτησης πληροφορίας αλλά τα τελευταία χρόνια αναπτύσσονται πολλές εφαρμογές που βασίζονται στη φιλοσοφία του LSI, όπως το information filtering και η προσπάθεια μοντελοποίησης της ανθρώπινης σκέψης. Παρακάτω θα αναπτύξουμε κάθε μια από αυτές τις εφαρμογές δίνοντας έμφαση στο information filtering.

4.8.1 Πολυγλωσσικό λεξικό.

Το LSI μπορεί να χρησιμοποιηθεί για την ανάκτηση κειμένων σε διάφορες γλώσσες. Αρχικά απαιτείται μια συλλογή κειμένων σε παραπάνω από μια γλώσσες. Δημιουργούμε τον πίνακα όρων-κειμένων, όπου κάθε κείμενο αποτελεί ένα συνδυασμό των ίδιων κειμένων αλλά σε διαφορετικές γλώσσες, ενώ στο χώρο που δημιουργείται όροι που έχουν την ίδια σημασία αλλά γράφονται σε διαφορετικές γλώσσες αναπαρίστανται στο ίδιο σημείο. Έτσι υπάρχουν όροι όλων των γλωσσών στον k-διάστατο χώρο που δημιουργείται με την εφαρμογή της SVD μεθόδου. Στη συνέχεια ενθέτουμε μονογλωσσικά κείμενα, τα οποία αν αναφέρονται στα ίδια θέματα καταλαμβάνουν το ίδιο σημείο στο χώρο ανεξάρτητα από τη γλώσσα στην οποία έχουν γραφεί. Έτσι μπορούμε να κάνουμε ένα query σε οποιαδήποτε γλώσσα και να μας επιστραφούν τα αντίστοιχα κείμενα όλων των γλωσσών.

4.8.2 Μοντελοποίηση Ανθρώπινης Σκέψης.

Το LSI μπορεί να χρησιμοποιηθεί για να μοντελοποιήσει την ανθρώπινη σκέψη και ειδικότερα τον τρόπο με τον οποίο ο άνθρωπος συσχετίζει τους όρους. Περισσότερο ενδιαφέρει η συσχέτιση μεταξύ των συνώνυμων λέξεων, δηλαδή όρων που έχουν την ίδια σημασία ή σχετίζονται με το ίδιο θέμα. Όπως έχουμε ήδη αναφέρει στο μοντέλο του LSI οι όροι που σχετίζονται τοποθετούνται στην ίδια περιοχή ακόμα και αν δεν ανήκουν στο ίδιο κείμενο.

Το LSI προτείνεται ως ένας καλός τρόπος για την εύρεση συνώνυμων λέξεων. Μάλιστα οι Landauer και Dumais χρησιμοποίησαν ένα τεστ του TOEFL για την εύρεση συνώνυμων λέξεων, προκειμένου να δοκιμάσουν το LSI ,το οποίο είχε πάρα πολύ καλή απόδοση.

4.8.3 Αυτόματη Βαθμολόγηση Εκθέσεων.

Αυτή η εφαρμογή αν και ακούγεται περίεργη παρόλα αυτά έδωσε καλά αποτελέσματα. Βασίζεται στο γεγονός ότι η βαθμολογία μιας έκθεσης γίνεται βάσει κάποιων χαρακτηριστικών όπως: το πλήθος των λέξεων, τη σύνταξη κλπ. Προκειμένου να εφαρμοστεί η τεχνική του LSI δημιουργήθηκε ένας χώρος από προβαθμολογημένες εκθέσεις. Στη συνέχεια τοποθετήθηκαν οι αβαθμολόγητες εκθέσεις με τρόπο ώστε κάθε έκθεση να τοποθετείται κοντά σε εκείνη (τη βαθμολογημένη) με την ίδια ποιότητα ώστε να βαθμολογηθεί ανάλογα. Μάλιστα ο υπολογιστής εμφανίστηκε ως πιο αντικειμενικός κριτής στην πλειοψηφία των περιπτώσεων.

4.8.4 Information filtering.

Το *Information Filtering* αποτελεί μια διαδικασία κατά την οποία ο χρήστης δεν πρέπει να ζητά μόνος του την πληροφορία, αλλά η πληροφορία πρέπει να κατευθύνεται προς αυτόν. Γενικά κάθε χρήστης ορίζει ένα profile για τον ευατό του, το οποίο τοποθετείται στο χώρο. Κάθε νέο στοιχείο είτε με τη μορφή άρθρου ή απλά ενημερωτικού υλικού που εισάγεται

εξετάζεται από το σύστημα, αν βρίσκεται μέσα στο profile του χρήστη, ώστε να προωθηθεί προς αυτόν. Ο τρόπος με τον οποίο δημιουργήθηκε το profile έχει μεγάλη σημασία για την απόδοση του συστήματος και την ακρίβεια στο φιλτράρισμα της πληροφορίας. Το profile ενός χρήστη δεν είναι στατικό αλλά μπορεί να μεταβάλλεται με τις παρεμβάσεις του ίδιου ώστε να προσαρμόζεται ή να ανανεώνεται ανάλογα. Αλλά ας δούμε τη διαδικασία πιο αναλυτικά.

Ο κάθε άνθρωπος αποφασίζει να διαβάσει άρθρα ή να συλλέξει πληροφορίες που απευθύνονται στα ενδιαφέροντα του. Αυτή η διαδικασία είναι αρκετά χρονοβόρα αφού τα δεδομένα είναι πάρα πολλά. Όσο λοιπόν αφορά την ηλεκτρονική πληροφορία αυτό μπορεί να γίνει αυτόματα και μάλιστα από το ίδιο το πληροφοριακό σύστημα. Παρόλο που αυτή η ιδέα ακούγεται ενδιαφέρουσα η υλοποίηση της είναι αρκετά δύσκολη και σημαντικό ρόλο παίζει όχι μόνο το πως φιλτράρεται η πληροφορία αλλά και πως ο χρήστης εκφράζει τα ενδιαφέροντα του ή αυτό που θα ονομάζουμε user-profile. Το LSI αποδεικνύεται ένας αποδοτικός τρόπος για information filtering σύμφωνα με πειράματα που έγιναν στη Bellcore από τους Dumais και Foltz με σκοπό το φιλτράρισμα άρθρων και πληροφοριών που αναφέρονταν στα επαγγελματικά ενδιαφέροντα εργαζομένων της εταιρείας.

Αν και μπορούμε να πούμε ότι το information filtering βρίσκεται κοντά στη φιλοσοφία του information retrieval οι δυο διαδικασίες έχουν δυο σημαντικές διαφορές. Σκοπός του information retrieval είναι να βρει πληροφορία μέσα σε δεδομένα , ενώ αντίθετα το information filtering στοχεύει στο να αφαιρέσει πληροφορία από ένα σύνολο δεδομένων. Επίσης ένα query στο IR εκφράζει μια προσωρινή πληροφοριακή ανάγκη ενός χρήστη, ενώ στο IF εκφράζει τα ενδιαφέροντα ενός χρήστη. Πάντως και στις δυο διαδικασίες δημιουργείται ένας πίνακας όρων-κειμένων του οποίου κάθε στοιχείο εκφράζει τη συχνότητα εμφάνισης ενός όρου στο αντίστοιχο κείμενο.

Σε μοντέλα IF που χρησιμοποιήθηκαν παλιότερα ο χρήστης περιέγραφε τα ενδιαφέροντα του από ένα σύνολο keywords και με εφαρμογή lexical matching επιστρέφονταν οι πληροφορίες που χαρακτηρίζονταν από τα ίδια keywords. Σε αυτή τη φιλοσοφία βρίσκοταν το Information Lens System και SDI (selective dissemination of information) που χρησιμοποιήθηκαν παλιότερα.

Η βασική ιδέα για IF με τη βοήθεια του LSI είναι η εξής: αρχικά δημιουργείται ένας χώρος κειμένων (που μπορεί να είναι άρθρα ή άλλες πληροφορίες), τα οποία έχουν ήδη κριθεί από το χρήστη ως σχετικά ή μη με τα ενδιαφέροντα του. Αν ένα καινούριο κείμενο είναι σχετικά κοντά με κείμενα που ενδιαφέρουν το χρήστη τότε θεωρείται πιθανό να ενδιαφέρει το χρήστη. Αντίθετα αν βρίσκεται μακριά τότε πιθανώς δεν ενδιαφέρει το χρήστη. Με παρόμοιο τρόπο καθορίζεται κατά πόσο κοντά είναι ένα καινούριο κείμενο στα keywords που ορίζουν ένα user-profile.Τα σχετικά κείμενα επιστρέφονται τελικά στο χρήστη.

Τα πειράματα με χρήση του LSI έγιναν με μια συλλογή τεχνικών άρθρων της Bellcore με σκοπό να προωθηθούν στους χρήστες ανάλογα με τα ενδιαφέροντα τους. Οι χρήστες μπορούσαν να ορίσουν το profile τους με τη βοήθεια μιας λίστας από keywords και με τη δυνατότητα relevance feedback προκειμένου να βαθμολογούνται οι απαντήσεις και το profile να προσαρμόζεται για ακριβέστερα αποτελέσματα. Στην περίοδο των 6 μηνών που το πείραμα είχε διάρκεια οι εργαζόμενοι μπορούσαν να ανανεώνουν τη λίστα των keywords που αρχικά είχαν δώσει.

Με βάση έναν πίνακα όρων-κειμένων, ο οποίος περιελάμβανε τις συχνότητες εμφάνισης των όρων, εφαρμόστηκε η SVD τεχνική και τα κείμενα αναπαραστάθηκαν σε ένα χώρο και παράλληλα το κάθε profile με τη μορφή κειμένου τοποθετήθηκε στον ίδιο χώρο μειωμένων διαστάσεων. Σύμφωνα με τη συνάρτηση συνημιτόνου τα κείμενα βαθμολογήθηκαν και τα κοντινότερα σε ένα profile (συνήθως τα 7 πρώτα) επιστρέφονταν στον αντίστοιχο χρήστη. Σε σύγκριση με την απόδοση άλλων μεθόδων IF η χρήση του LSI έδωσε τα καλύτερα αποτελέσματα, τα οποία βελτιώνονταν κάθε μήνα με χρήση της επανατροφοδότησης από τη

μεριά του χρήστη (relevance feedback). Βέβαια εδώ δε θα πρέπει να ξεχνάμε ότι σημαντικό ρόλο στην απόδοση της μεθόδου παίζει ο αριθμός των καινούριων κάθε φορά άρθρων του μήνα που εισάγονταν στο σύστημα (θεωρώντας σταθερό τον αριθμό των άρθρων που πρέπει να επιστραφούν στο χρήστη). Μια ιδέα που θα μπορούσε να βελτιώσει παραπάνω τα αποτελέσματα είναι ο ορισμός και των αρνητικών ενδιαφερόντων του χρήστη μια και έτσι θα είχαμε ακριβέστερη περιγραφή του profile ενός χρήστη.

Βιβλιογραφία

[BDJ99] M. W. Berry, Z. Drmac, E. R. Jessup, *Matrices Vector Spaces and Informational Retrieval*, SIAM Review 1999, Vol. 41, pp.335-362.

[BDO95] M. W. Berry, S. T. Dumais, G.W. O'Brien, *Using linear algebra for intelligent information retrieval,* SIAM Review 1995, Vol. *37*, pp. 573-595.

[CW85] Cullum, J.K. & Willoughby, R.A. Lancoz algorithms for large symmetric eigenvalue computations- Vol 1 Theory, 1985

[DDFLH90] S. Deerwester, S. T. Dumais, G. Furnas, Th. K. Landauer, R. Harshman, "Indexing by Latent Semantic Analysis", *Journal of the Society for Information Science 1990*, Vol. 41, pp. 391-407.

[DLLL97] S. T. Dumais, T. A. Letsche, M. L. Littman, T. K. Landauer, *Automatic cross-language retrieval using Latent Semantic Indexing,* In AAAI Spring Symposuim on Cross-Language Text and Speech Retrieval, March 1997.

[FD92] P. W. Foltz, S. T. Dumais, *Personalized information delivery: An analysis of information filtering methods*, Communications of the ACM 1992,Vol. 35, pp. 51-60.

[PTRV98] Ch. H. Papadimitriou, H. Tamaki, P. Raghavan, S. Vempala, *Latent Semantic Indexing : A Probabilistic Analysis,* Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1998, Seattle, WA USA.

[WMB] Witten I. H., Moffat A., Bell T. C., Managing Gigabytes, Compressing and Indexing Documents and Images, Second Edition, MORGAN KAUFMANN PUBLISHERS INC. San francisco, California

Κεφάλαιο 5. Ανάκτηση Πληροφορίας στο Διαδίκτυο

5.1 Βασικές Έννοιες

Στην ενότητα αυτή παρουσιάζονται ορισμένες βασικές έννοιες σχετικά με την Ανάκτηση Πληροφορίας στο Διαδίκτυο. Αρχικά δίνεται μία σύντομη περιγραφή του διαδικτύου και κάποια στοιχεία για την ανάπτυξή του. Μετά αναφέρονται οι ιδιαιτερότητες της ανάκτησης πληροφορίας στο διαδίκτυο σε σχέση με τις συμβατικές εφαρμογές ανάκτησης πληροφορίας. Στη συνέχεια περιγράφονται τα βασικά χαρακτηριστικά μηχανών αναζήτησης που στηρίζονται μόνο στο περιεχόμενο των σελίδων, περιγράφεται περιληπτικά η γλώσσα HTML, που χρησιμοποιείται για τη δημιουργία σελίδων του διαδικτύου και τέλος αναφέρεται η σημασία των διασυνδέσεων μεταξύ των σελίδων αυτών και η αξία της πληροφορίας που μεταφέρουν.

5.1.1 Διαδίκτυο (World Wide Web)

Λίγα γεγονότα στην ιστορία των υπολογιστών έχουν επηρεάσει τόσο βαθιά την κοινωνία και την καθημερινή ζωή των ανθρώπων όσο η δημιουργία και η ανάπτυξη του διαδικτύου. Το διαδίκτυο είναι ένα σύνολο από ηλεκτρονικές σελίδες (web pages) μεγάλης πολυπλοκότητας οι οποίες εισάγονται και εξάγονται από αυτό με μία διαδικασία εντελώς αποκεντρωμένη και χαοτική.

Ο καθένας μπορεί να φτιάξει μία σελίδα όπως θέλει χωρίς κάποια καθορισμένη δομή και πρότυπα περιεχομένου. Οι σελίδες του διαδικτύου μπορούν να έχουν γραφτεί σε διάφορες γλώσσες, διαλέκτους ή μορφές από άτομα με διαφορετικό υπόβαθρο, μόρφωση, κουλτούρα, ενδιαφέροντα και κίνητρα. Επίσης κάθε σελίδα μπορεί να διαφέρει στο μέγεθος, να περιέχει είτε αλήθειες, είτε ψέματα, είτε ανοησίες, είτε προπαγάνδα.

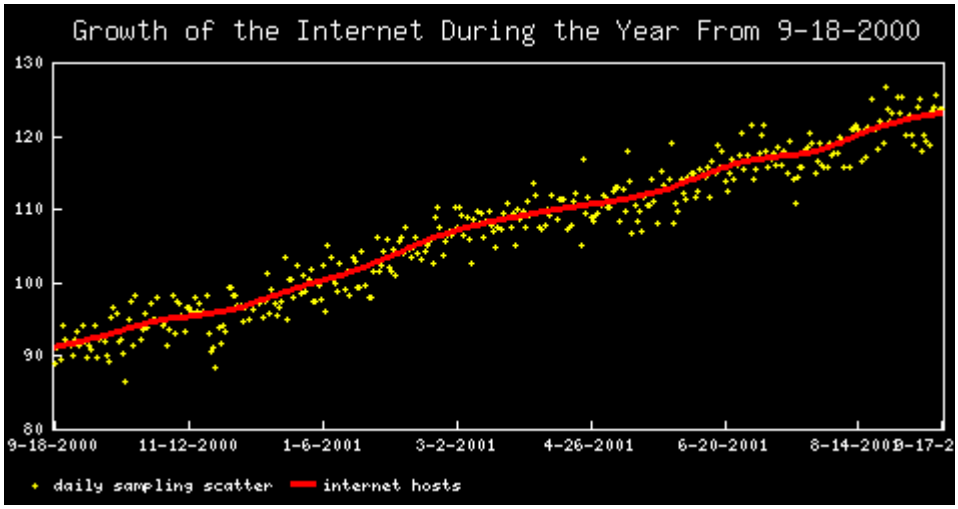
Καθημερινά στο διαδίκτυο προστίθενται περίπου ένα εκατομμύριο καινούριες σελίδες με αποτέλεσμα το μέγεθος του World Wide Web να αυξάνει διαρκώς και με ιδιαίτερα γρήγορους ρυθμούς. Ενδεικτικά στο Σχήμα 1 δίνεται ένας πίνακας με στοιχεία για το μέσο μηνιαίο αριθμό των hosts που υπάρχουν στο διαδίκτυο από το 1999 μέχρι σήμερα [Telcordia].

Μέσος Μηνιαίος Αριθμός Hosts σε Εκατομμύρια			
Μήνας	**1999**	**2000**	**2001**
Ιανουάριος	43.5996	69.4882	100.497
Φεβρουάριος	47.5934	72.2051	103.994
Μάρτιος	50.1986	74.3473	107.222
Απρίλιος	53.0498	76.3923	109.864
Μάιος	55.3170	79.2922	112.108
Ιούνιος	56.8890	81.8652	114.839
Ιούλιος	58.6299	84.7157	117.323
Αύγουστος	60.4827	87.4202	120.205
Σεπτέμβριος	62.3218	90.2854	-
Οκτώβριος	64.5587	92.9333	-
Νοέμβριος	66.2607	95.1675	-
Δεκέμβριος	67.7035	97.7895	-

Σχήμα 1 –Μέσος μηνιαίος αριθμός hosts στο διαδίκτυο

Στο Σχήμα 2 δίνεται μία εικόνα της ανάπτυξης του διαδικτύου από τις 18 Σεπτεμβρίου του 2000 μέχρι τις 14 Αυγούστου του 2001 [Telcordia].

Με την κόκκινη γραμμή παριστάνεται ο αριθμός των hosts στο διαδίκτυο. Στον οριζόντιο άξονα δίνονται κάποιες ενδεικτικές χρονικές στιγμές και στον κάθετο άξονα δίνεται ο αριθμός των hosts σε εκατομμύρια.



Σχήμα 2 – Ανάπτυξη του διαδικτύου κατά τη χρονιά 2001

Από τα παραπάνω στοιχεία προκύπτει ότι ο όγκος πληροφορίας που είναι αποθηκευμένος στο διαδίκτυο είναι πολύ μεγάλος και ιδιαίτερα χρήσιμος και διαρκώς αυξάνεται. Αυτό είναι και το βασικό προτέρημα του διαδικτύου, όπως και οι τρομερές δυνατότητες που δίνονται από τον άμεσο τρόπο με τον οποίο προσθέτονται καινούριες σελίδες σε αυτό.

Από την άλλη πλευρά όμως ο άναρχος τρόπος εισαγωγής των σελίδων είναι και το βασικό μειονέκτημα του. Η έλλειψη προτύπων και η παντελής έλλειψη δομής και οργάνωσης των σελίδων αυτών έχει οδηγήσει σε ένα χαοτικό σύνολο στο οποίο η πρόσβαση στην πληροφορία είναι ιδιαίτερα δυσχερής.

5.1.2 Ιδιαιτερότητες της Ανάκτησης Πληροφορίας στο Διαδίκτυο

Οι τεχνικές που πρέπει να ακολουθηθούν κατά την ανάκτηση πληροφορίας από το διαδίκτυο είναι αρκετά διαφορετικές από αυτές που χρησιμοποιούνται στα συμβατικά συστήματα ανάκτησης πληροφορίας. Οι κύριοι λόγοι που συμβαίνει αυτό είναι οι ακόλουθοι:

- Οι χρήστες του διαδικτύου συνηθίζουν να δίνουν πολύ μικρά ερωτήματα, της τάξης των τριών λέξεων το πολύ, και δεν είναι πρόθυμοι να δώσουν επιπλέον πληροφορίες. Επίσης δε δίνουν ιδιαίτερη σημασία στην ακριβή διατύπωση της πληροφοριακής του ανάγκης με αποτέλεσμα τα ερωτήματά τους να είναι αρκετά ασαφή. Έτσι συχνά τα αποτελέσματα δεν ταιριάζουν με την πραγματική πληροφοριακή ανάγκη του χρήστη και ο χρήστης απογοητεύεται.
- Η συλλογή των σελίδων αλλάζει συνέχεια, καθώς χιλιάδες καινούριες σελίδες εμφανίζονται καθημερινά στο διαδίκτυο και άλλες εμφανίζονται διαφορετικές. Έτσι το σύστημα επιβαρύνεται καθώς πρέπει να ενημερώνει διαρκώς τις ήδη αποθηκευμένες σελίδες και να προσθέτει συνέχεια καινούριες.
- Η χρησιμότητα των σελίδων ποικίλλει. Μερικές σελίδες εστιάζουν ιδιαίτερα σε κάποιο θέμα, άλλες δίνουν διάφορες πληροφορίες για πολλά, και πολλές φορές άσχετα μεταξύ τους θέματα. Επίσης υπάρχουν σελίδες που σκοπός τους δεν είναι καν η παροχή πληροφοριών.
- Η ποιότητα της πληροφορίας που περιέχεται στις σελίδες δεν εξασφαλίζεται, καθώς μπορεί να αναφέρονται ανακρίβειες ή ακόμα και να γίνεται προπαγάνδα μέσω κάποιων από αυτές.

- Η προεπεξεργασία όλων των σελίδων του διαδικτύου απαιτεί μεγάλο κόστος χρόνου και χώρου και είναι ουσιαστικά ανέφικτη λόγω του μεγάλου όγκου της υπάρχουσας πληροφορίας.

Οι μηχανές που δημιουργήθηκαν για να βοηθούν τους χρήστες να βρουν κάποια χρήσιμη για αυτούς πληροφορία στο διαδίκτυο και έχουν λάβει υπόψη τους παραπάνω παράγοντες, ονομάζονται μηχανές αναζήτησης (Web Search Engines).

5.1.3 Μηχανές Αναζήτησης (Search Engines)

Μία μηχανή αναζήτησης αποτελείται από τρία βασικά τμήματα [K97, BR99]:

- Έναν *crawler*, ο οποίος ξεκινώντας από ένα αρχικό σύνολο σελίδων, συλλέγει σελίδες διατρέχοντας το διαδίκτυο με χρήση των υπερδεσμών (links). Καθώς ο κάθε crawler ακολουθεί διαφορετικές τεχνικές σχετικά με το ποιους υπερδεσμούς θα ακολουθήσει, είναι φυσικό επακόλουθο οι σελίδες που συλλέγονται από διαφορετικούς crawlers να είναι διαφορετικές.
- Έναν *δεικτοδοτητή* (indexer), ο οποίος επεξεργάζεται τις σελίδες που συνέλεξε ο crawler. Αρχικά αποφασίζει ποιες από τις σελίδες θα ταξινομήσει, καθώς είναι πιθανό μερικές σελίδες να εμφανίζονται περισσότερες από μία φορά στο σύνολο που πήρε από τον crawler. Στη συνέχεια σχεδιάζεται η δομή δεικτοδότησης που θα αναπαραστήσει τις σελίδες, όπου η πιο συχνή επιλογή είναι η χρήση της δομής ενός ανεστραμμένου δείκτη (inverted index). Ο indexer μπορεί να δημιουργήσει και άλλες επιπρόσθετες δομές για την αποθήκευση επιπρόσθετης πληροφορίας όπως αναπαραστάσεις του γράφου του διαδικτύου, αυθεντικά κείμενα σελίδων κ.ο.κ.
- Έναν *επεξεργαστή ερωτήματος* (query processor), ο οποίος επεξεργάζεται τα διάφορα ερωτήματα και επιστρέφει τις σχετικές σελίδες σε μία σειρά που καθορίζεται από έναν αλγόριθμο αξιολόγησης. Ο επεξεργαστής μετασχηματίζει τα ερωτήματα που δέχεται από τους χρήστες, σε μία πρότυπη μορφή, χρησιμοποιώντας τις δομές δεδομένων του δεικτοδοτητή για να εντοπίσει τις κατάλληλες σελίδες που ταιριάζουν με τα δεδομένα του χρήστη και τέλος τις διατάζει σύμφωνα με τον αλγόριθμο αξιολόγησης.

Στη συνέχεια παρουσιάζονται βασικές λειτουργίες των τριών αυτών τμημάτων.

Crawler

Ο crawler πρέπει να καθορίσει ποιες σελίδες του διαδικτύου θα συλλέξει για δεικτοδότηση και ίσως αποθήκευση. Συνήθως σε κάθε σελίδα του διαδικτύου ανατίθεται ένας αριθμός προτεραιότητας, ο οποίος καθορίζει τη σημαντικότητα μίας σελίδας προς συλλογή. Αυτός ο αριθμός προτεραιότητας μπορεί να χρησιμοποιηθεί είτε για την αξιολόγηση της ποιότητας των σελίδων είτε ως κριτήριο για την επιλογή των σελίδων του δεικτοδοτητή που θα πρέπει να ενημερωθούν. Επίσης ο crawler πρέπει να λαμβάνει υπόψην του, κατά τη λειτουργία του, διάφορους παράγοντες απόδοσης έτσι ώστε να μην επιβαρύνει τα συστήματα (servers) που επισκέπτεται.

Indexer

Ο σκοπός του δεικτοδοτητή (indexer), όπως αναφέρθηκε και προηγουμένως, είναι η δημιουργία όλων των δομών δεδομένων που απαιτούνται για την δεικτοδότηση σελίδων, όπως δομές ανεστραμμένων δεικτών, βάσεις δεδομένων με τις διευθύνσεις στο διαδίκτυο των δεικτοδοτούμενων σελίδων (URLs), βάσεις δεδομένων με τα αυθεντικά κείμενα των σελίδων, και μοντέλα αναπαράστασης του διαδικτύου.

Η δομή ανεσταμμένου δείκτη συνίσταται σε ένα κατάλογο που περιέχει για κάθε όρο διεκτοδότησης μία λίστα με τους κωδικούς των σελίδων που περιέχουν τον όρο αυτό και τις

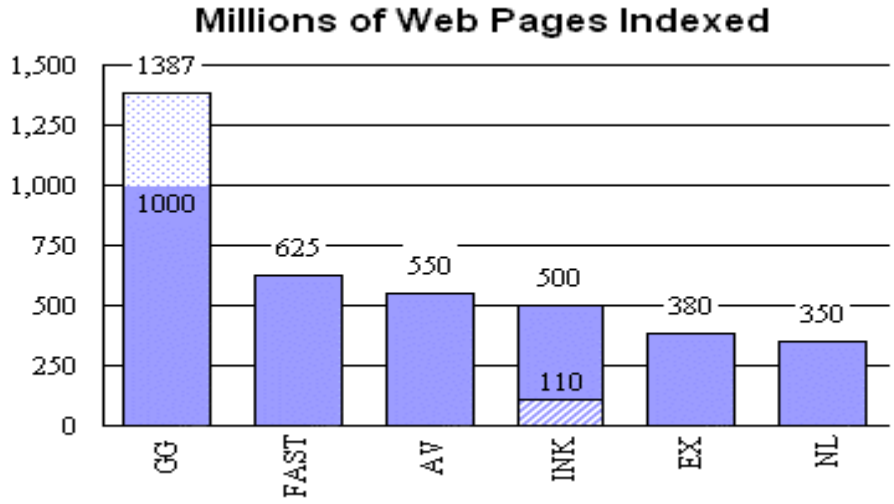
θέσεις των σελίδων στις οποίες βρίσκεται η λέξη. Η λίστα κάθε λέξης είναι ταξινομημένη λεξικογραφικά βάση του ζευγαριού του κωδικού της σελίδας και της θέσης της λέξης στη σελίδα αυτή.

Για εξοικονόμηση χώρου οι σελίδες αναπαριστάνονται με τους κωδικούς τους σε αυτόν τον κατάλογο και στις άλλες δομές δεδομένων, κάτι που πραγματοποιείται μέσω μίας βάσης δεδομένων που αποθηκεύει τα URLs των σελίδων μαζί με τους κωδικούς τους.

Πριν ο δεικτοδοτητής αρχίσει να γεμίζει με δεδομένα τις ανωτέρω δομές θα πρέπει να καθορίσει ποιες σελίδες θα ταξινομήσει και θα αποθηκεύσει. Αυτό για παράδειγμα μπορεί να γίνει βαθμολογώντας τις σελίδες και επιλέγοντας τελικά αυτές με τη μεγαλύτερη βαθμολογία. Με αυτό τον τρόπο μπορεί να δίνει μηδενική βαθμολογία στα διπλότυπα των σελίδων.

Ενδεικτικά στο Σχήμα 3 δίνεται ο αριθμός των σελίδων που έχει ταξινομήσει κάθε μηχανή αναζήτησης. Τα στοιχεία αυτά έχουν ανακοινωθεί από τις αντίστοιχες μηχανές αναζήτησης μέχρι τις 15 Αυγούστου του 2001 [Search Engine Watch].

GG=Google, FAST=FAST, AV=AltaVista, INK=Inktomi, WT=WebTop.com, NL=Northern Light, EX=Excite.



Σχήμα 3 – Μέγεθος καταλόγων μηχανών αναζήτησης

Επεξεργαστής Ερωτήματος

Ο στόχος του επεξεργαστή ερωτήματος είναι η αξιολόγηση των σελίδων που έχουν σχέση με το ερώτημα που έθεσε ο χρήστης και η επιστροφή μίας διατεταγμένης λίστας από σελίδες (με κάποια διάταξη αξιολόγησης). Η διάταξη αυτή πραγματοποιείται με τη χρήση κριτηρίων που είναι είτε εξαρτώμενα από το εκάστοτε ερώτημα είτε ανεξάρτητα από αυτό. Κριτήρια για μία σελίδα εξαρτώμενα από το εκάστοτε ερώτημα είναι το μέγεθος της σελίδας, ο διαδικτυακός κόμβος που ανήκει, η γλώσσα, το λεξιλόγιο κ.ο.κ. Κριτήρια εξαρτώμενα πο το ερώτημα είναι η συχνότητα εμφάνισης των όρων ερωτήματος στη σελίδα, οι θέσεις εμφάνισης κ.ο.κ.

Μειονεκτήματα Κλασσικών Μηχανών Αναζήτησης

Το βασικό μειονέκτημα των μηχανών αυτών είναι ο μεγάλος όγκος πληροφορίας που επιστρέφουν στα ερωτήματα που διατυπώνονται από τους χρήστες κάτι που καθιστά την

πληροφορία ουσιαστικά ανεκμετάλλευτη. Κάθε χρήστης κατά μέσο όρο κοιτάει τις πρώτες είκοσι το πολύ σελίδες που θα του επιστρέψει η μηχανή αναζήτησης και συνήθως επισκέπτεται τις δέκα από αυτές.

Μία απλή μέθοδος για την αντιμετώπιση του προβλήματος αυτού είναι η αξιολόγηση των ανακτώμενων σελίδων με βάση τη συχνότητα εμφάνισης του όρου δεικτοδότησης στο κείμενο των σελίδων. Παραλλαγές αυτής της τεχνικής είναι η παροχή αυξημένου βάρους σε σελίδες που περιέχουν τον όρο αναζήτησης στον τίτλο της σελίδας ή στις επικεφαλίδες ή είναι γραμμένες με τέτοιο τρόπο ώστε να του δίνεται μία ιδιαίτερη σημασία.

Όμως οι μηχανές αναζήτησης που ενσωματώνουν τέτοιες μεθόδους είναι πολύ εύκολο να ξεγελαστούν από τους σχεδιαστές των σελίδων, γιατί, όπως περιγράφηκε και παραπάνω, τα αποτελέσματα τους βασίζονται μόνο στο περιεχόμενο των σελίδων. Έτσι εμφανίστηκε, ιδιαίτερα παλαιότερα, το φαινόμενο οι σχεδιαστές των σελίδων να γράφουν πολλές φορές κάποιες λέξεις κλειδιά στις σελίδες τους έτσι ώστε να βαθμολογούνται περισσότερο από τέτοιες μηχανές αναζήτησης. Για να μην γίνεται αυτό εμφανές στο χρήστη αυτές οι λέξεις κλειδιά ήταν γραμμένες με τέτοιο τρόπο ώστε να μην είναι ορατές από το χρήστη.

Ακόμα η αξιολόγηση των σελίδων βάση και μόνο του περιεχομένου τους, καταλήγει και σε ένα άλλο βασικό πρόβλημα. Πολλές σελίδες δεν αναφέρουν ρητά στο περιεχόμενό τους το θέμα στο οποίο ανήκει η πληροφορία που παρέχουν. Για παράδειγμα σπάνια στη σελίδα μίας μηχανής αναζήτησης θα εμφανίζεται κάπου ο όρος "μηχανή αναζήτησης".

Ένα άλλο σημαντικό πρόβλημα που προκύπτει από αυτή τη μεθοδολογία είναι ότι πολλές φορές ο όρος που δίνει ο χρήστης σαν είσοδο έχει διάφορες σημασίες. Για παράδειγμα ο όρος jaguar μπορεί να αναφέρεται είτε στη συγκεκριμένη μάρκα αυτοκινήτου είτε και στο ζώο. Έτσι αν ο ίδιος ο χρήστης δε δώσει επιπλέον πληροφορία στη μηχανή για το τι ψάχνει θα του επιστραφούν σχεδόν σίγουρα και σελίδες για ένα θέμα που όχι μόνο δεν τον ενδιαφέρει, αλλά ούτε το είχε σκεφτεί.

Ένας βασικός ακόμα παράγοντας που επηρεάζει ιδιαίτερα το αποτέλεσμα που θα επιστραφεί στο χρήστη είναι οι συνώνυμες λέξεις. Για παράδειγμα εάν ο χρήστης δώσει σαν είσοδο τη λέξη "αμάξι" είναι πολύ πιθανό να μην του επιστραφούν σελίδες που να αναφέρουν τη λέξη "αυτοκίνητο" και να μην αναφέρουν τη λέξη "αμάξι". Έτσι χάνεται ένα σημαντικό μέρος χρήσιμης για το χρήστη πληροφορίας.

Τέλος με τον τρόπο που γίνεται η αξιολόγηση των σελίδων δεν δίνεται στο χρήστη κάποια εγγύηση ότι οι σελίδες που του επιστρέφονται είναι αξιόπιστες και αληθινές ούτε και του δίνεται η αίσθηση ότι αυτές έχουν αξιολογηθεί βάση κάποιας μορφής ανθρώπινης κρίσης.

Έτσι το ερώτημα που προκύπτει είναι πως μπορούν οι χρήστες να εντοπίζουν εύκολα και γρήγορα μόνο την πληροφορία που επιθυμούν και να έχουν τη δυνατότητα να πιστέψουν ότι είναι αληθινή και αξιόπιστη.

Όπως αναφέρθηκε παραπάνω το διαδίκτυο δεν έχει κάποια δομή ή οργάνωση. Όμως το βασικό στοιχείο που παρέχει κάποια μορφή οργάνωσης και δομής είναι οι υπερδεσμοί (hyperlinks) μεταξύ των σελίδων. Οι υπερδεσμοί αυτοί δίνουν μία χαλαρή μορφή συνοχής στο μεγάλο όγκο πληροφορίας που είναι αποθηκευμένος στο διαδίκτυο. Έτσι είναι εμφανές ότι αποτελούν σημαντική πηγή πληροφορίας για το διαδίκτυο και για την πληροφορία που αυτό περιέχει την οποία όμως δεν εκμεταλλεύονται καθόλου οι κλασσικές μηχανές ψαξίματος.

5.1.4 Η Γλώσσα HTML

Ο όρος HTML προκύπτει από τα αρχικά των λέξεων HyperText Markup Language και σημαίνει γλώσσα περιγραφής σελίδων του διαδικτύου.

Βασικό στοιχείο της γλώσσας αυτής είναι έννοια του “tag” (ετικέτα). Κάθε στοιχείο των σελίδων HTML εμφανίζεται ανάμεσα σε ετικέτες (tags) και οι ετικέτες αυτές καθορίζουν την τοποθεσία μέσα στη σελίδα των στοιχείων αυτών και τη μορφή με την οποία θα εμφανίζονται και θα φαίνονται.

Όλες οι σελίδες HTML ξεκινούν με την ετικέτα <html> και τελειώνουν με την αντίστοιχη ετικέτα τέλους </html>. Επίσης κάθε σελίδα HTML αποτελείται από δύο τμήματα. Το πρώτο τμήμα καθορίζεται από τις ετικέτες <head> και </head> και το δεύτερο από τις <body> και </body>.

Το πρώτο τμήμα αποτελεί και την κεφαλή του κειμένου και καθορίζει διάφορες παραμέτρους της συγκεκριμένης σελίδας. Για παράδειγμα καθορίζει τον τίτλο της, τη σχέση της με άλλες σελίδες, τη μορφή που μπορεί να έχει ή ακόμα και τη scripting γλώσσα που μπορεί να χρησιμοποιήσει. Από τα παραπάνω στοιχεία αυτό που χαρακτηρίζει κατά κάποιο τρόπο της συγκεκριμένη σελίδα είναι ο τίτλος της ο οποίος εμφανίζεται ανάμεσα στις ετικέτες <title> και </title>.

Το δεύτερο τμήμα αποτελεί το σώμα της σελίδας και είναι αυτό που περιέχει όλη την πληροφορία που επιθυμεί ο δημιουργός της σελίδας να παρουσιάσει, είτε με τη μορφή κειμένου, είτε εικόνων, είτε ακόμα και με διασυνδέσεις προς άλλες σελίδες δικές του ή άλλων ατόμων. Από τα στοιχεία που μπορούν να τοποθετηθούν στο σώμα μίας σελίδας HTML τα περισσότερα μπορούν να ενταχθούν σε δύο κατηγορίες.

Η πρώτη αποτελείται από τα στοιχεία ορισμού περιοχής τα οποία είναι οι επικεφαλίδες, οι παράγραφοι και οι οριζόντιες γραμμές. Σημαντικό ενδιαφέρον παρουσιάζουν οι επικεφαλίδες, οι οποίες καθώς τονίζονται από το δημιουργό της σελίδας υποδηλώνουν ότι περιέχουν κάποια πληροφορία η οποία πρέπει να προσεχτεί. Υπάρχουν έξι επίπεδα από επικεφαλίδες, το *H1* είναι το πιο σημαντικό και το *H6* το λιγότερο σημαντικό. Το κείμενο που εμφανίζεται σαν επικεφαλίδα περιέχεται ανάμεσα σε ετικέτες της μορφής *<h1>* και *</h1>*, δηλαδή ανάλογα με την επικεφαλίδα καθορίζεται και η ετικέτα.

Η δεύτερη βασική κατηγορία αποτελείται από τα στοιχεία ορισμού κειμένου τα οποία ορίζουν τύπους χαρακτήρων στο κείμενο. Βασική υποκατηγορία αυτών των στοιχείων αποτελούν τα στοιχεία τύπου γραμματοσειράς. Ανάλογα με τα στοιχεία που επιλέγει ο δημιουργός της σελίδας μπορεί να παρουσιάσει κάποιο κείμενο, είτε με πιο έντονα γράμματα ,είτε με πλάγιους χαρακτήρες, είτε να είναι υπογραμμισμένο. Ένα κείμενο για να εμφανίζεται με έντονα γράμματα πρέπει να βρίσκεται είτε ανάμεσα στις ετικέτες ** και ** είτε ανάμεσα στις ** και **. Για να εμφανίζεται με πλάγιους χαρακτήρες πρέπει να βρίσκεται ανάμεσα στις ετικέτες ** και **. Τέλος για να είναι υπογραμμισμένο πρέπει να βρίσκεται ανάμεσα στις ετικέτες *<u>* και *</u>*.

Όπως αναφέρθηκε και παραπάνω ο δημιουργός της σελίδας μπορεί να εμφανίζει στη σελίδα του διάφορες εικόνες. Για να το επιτύχει αυτό χρειάζεται ένα άλλο στοιχείο της γλώσσας HTML, το img, το οποίο ανήκει στα στοιχεία ορισμού κειμένου. Το στοιχείο αυτό καθορίζεται από την ετικέτα *<img>* και δεν έχει ετικέτα τέλους. Για να εμφανιστεί μία εικόνα σε μία σελίδα *HTML* πρέπει να υπάρχει στον κώδικα της σελίδας το στοιχείο img με την παρακάτω μορφή: *<img src=”imagepath/imagename.jpeg” alt=”comment”>*. Στο πεδίο *«src»* του στοιχείου *img* δίνεται πρώτα το μονοπάτι που δείχνει τον κατάλογο στον οποίο είναι αποθηκευμένη η εικόνα που θα εμφανιστεί και μετά δίνεται το όνομα της εικόνας με την κατάληξη του τύπου της. Στο πεδίο «alt» δίνεται το κείμενο που επιθυμεί ο δημιουργός να φαίνεται μέχρι να εμφανιστεί η εικόνα, είναι εμφανές ότι το κείμενο αυτό περιγράφει κατά κάποιο τρόπο την εικόνα και το θέμα της.

Τέλος ένα ακόμα πολύ χρήσιμο και θεμελιώδες στοιχείο της γλώσσας HTML είναι οι υπερδεσμοί. Οι υπερδεσμοί περιέχονται ανάμεσα στις ετικέτες *<a>* και ** και έχουν την ακόλουθη μορφή: *hyperlink - text*. Στο πεδίο *«href»* του στοιχείου *a* δίνεται η διεύθυνση της σελίδας προς την οποία δείχνει ο συγκεκριμένος υπερδεσμός. Ανάμεσα στις δύο ετικέτες δίνεται το κείμενο που παρουσιάζει ο υπερδεσμός και η σελίδα προς την οποία δείχνει, επομένως κατά κάποιο τρόπο παρουσιάζει και το περιεχόμενο της σελίδας αυτής.

5.1.5. Υπερδεσμοί (links)

Όπως έχει αναφερθεί οι υπερδεσμοί μεταξύ των σελίδων μεταφέρουν μία πολύ σημαντική πληροφορία η οποία έχει να κάνει με τη σχέση των σελίδων που συνδέονται μέσω αυτών. Από τη δομή που σχηματίζεται από τις διασυνδέσεις αυτές κωδικοποιείται (εμπεριέχεται) ένα σημαντικό ποσό άδηλης ανθρώπινης κρίσης. Ακριβώς αυτός ο παράγοντας ανθρώπινης κρίσης είναι που λείπει από τις term-based search engines για να καθοριστεί η έννοια της ποιότητας μίας σελίδας.

Συγκεκριμένα, η δημιουργία ενός υπερδεσμού στο διαδίκτυο παρουσιάζει μία σημαντική ένδειξη του ακόλουθου τύπου κρίσης: ο κατασκευαστής της σελίδας *A* περιλαμβάνει στη σελίδα του ένα υπερδεσμό προς τη σελίδα Β αν πιστεύει ότι η Β περιέχει σημαντική και αξιόλογη πληροφορία και πιθανόν σχετική με αυτή που περιέχεται στη σελίδα Α. Έτσι μπορεί να χρησιμοποιηθεί ο αριθμός των υπερδεσμών που δείχνουν σε μία σελίδα (in-degree) ως ένα μέτρο για την αξιολόγηση της ποιότητας της σελίδας.

Ακολουθώντας την ίδια λογική με παραπάνω μπορεί να θεωρηθεί ότι εάν η σελίδα *A* έχει υπερδεσμούς σε πολλές καλές και ποιοτικές σελίδες τότε η άποψη και η κρίση του κατασκευαστή της σελίδας Α αποκτά μεγαλύτερη σημασία και γίνεται αξιοπρόσεκτη. Έτσι κατά συνέπεια το γεγονός ότι η σελίδα Α έχει ένα υπερδεσμό προς τη σελίδα Β υποδηλώνει ότι ίσως και η σελίδα Β είναι μία ποιοτική σελίδα.

Από την άλλη πλευρά όμως η θεώρηση και η χρήση των υπερδεσμών ως μία σημαντική πηγή πληροφορίας ελλοχεύει πολλούς κινδύνους και παγίδες. Αυτοί προκύπτουν από τη διαφορετική σημασία που μπορεί να έχει κάποιος υπερδεσμός. Δηλαδή πολλοί από τους υπερδεσμούς που εμφανίζονται στο διαδίκτυο δεν έχουν σκοπό να υποδηλώσουν την ποιότητα και το περιεχόμενο μίας σελίδας, αλλά έχουν δημιουργηθεί για να ικανοποιήσουν κάποιους άλλους σκοπούς. Για παράδειγμα έχουν δημιουργηθεί για καθαρά και μόνο λόγους πλοήγησης, ώστε να διευκολύνουν την κίνηση του χρήστη μέσα στο διαδικτυακό κόμβο. Ακόμα οι λόγοι μπορεί να είναι εμπορικοί ή και διαφημιστικοί, καθώς πολλές επιχειρήσεις παρουσιάζονται και διαφημίζονται μέσω του διαδικτύου, επακόλουθο της ανάπτυξης του.

Λαμβάνοντας επομένως υπόψη και τις διασυνδέσεις μεταξύ των σελίδων και όχι μόνο το περιεχόμενο αυτών ως πηγή πληροφορίας το διαδίκτυο μπορεί να αναπαρασταθεί σαν ένας γράφος με πολλούς διαφορετικούς τρόπους. Οι μηχανές αναζήτησης που έχουν χρησιμοποιήσει αυτή τη θεώρηση είναι γνωστές σαν *connectivity-based search engines*.

Οι περισσότερες από αυτές τις μηχανές αναζήτησης έχουν θεωρήσει την πιο εμφανή αναπαράσταση, δηλαδή ο γράφος περιέχει έναν κόμβο για κάθε σελίδα *u* και υπάρχει μία κατευθυνόμενη ακμή (u,v) αν και μόνο αν η σελίδα *u* περιέχει ένα υπερδεσμό προς τη σελίδα *v*.

5.2. Link-based Τεχνικές Αναζήτησης στο Διαδίκτυο

Σε αυτή την ενότητα περιγράφονται διάφορες τεχνικές που έχουν προταθεί για αποτελεσματική αναζήτηση στο διαδίκτυο με χρήση των διασυνδέσεων μεταξύ των σελίδων. Ιδιαίτερο βάρος δίνεται σε δύο από αυτές καθώς ήταν από τις πρώτες που παρουσιάστηκαν. Η μία από αυτές είναι ο αλγόριθμος HITS, που πρότεινε ο Kleinberg ([K97]) και χρησιμοποιείται στο σύστημα CLEVER, και η άλλη είναι αυτή των Brin και Page ([BP98], που χρησιμοποιείται στο σύστημα GOOGLE. Η άλλη τεχνική που παρουσιάζεται χρησιμοποιεί και συνδυάζει στοιχεία και των δύο παραπάνω προσεγγίσεων.

5.2.1 Ο αλγόριθμος HITS (Hyperlink-Induced Topic Search)

Το κεντρικό θέμα στο οποίο στηρίζεται η τεχνική είναι η διύλιση ευρέων θεμάτων αναζήτησης μέσω της εύρεσης των *authoritative* (αξιόπιστων) πηγών πληροφορίας για αυτά τα θέματα. Προτείνεται και ελέγχεται μία αλγοριθμική διατύπωση της έννοιας του authority στηριζόμενη στη σχέση μεταξύ ενός συνόλου σχετικών authority σελίδων και ενός συνόλου hub σελίδων που τις ενώνει όπως φαίνεται στη δομή των διασυνδέσεων. Αυτή η διατύπωση σχετίζεται με τις ιδιοτιμές κάποιων πινάκων που σχετίζονται με τον πίνακα γειτνίασης που αντιστοιχεί στο γράφο των σελίδων.

Εισαγωγικά Στοιχεία

Το πρόβλημα της αναζήτησης στο διαδίκτυο, μπορεί να οριστεί ως η διαδικασία για την εύρεση σελίδων που είναι σχετικές με τα διάφορα ερωτήματα που θέτει ο χρήστης. Η ποιότητα μίας μεθόδου αναζήτησης απαιτεί απαραίτητα μία μορφή ανθρώπινης εκτίμησης, λόγω της έμφυτης υποκειμενικότητας της έννοιας της ομοιότητας. Όπως έχει αναφερθεί οι term-based μηχανές αναζήτησης ταξινομούν ένα αρκετά μεγάλο κομμάτι του διαδικτύου και ανταποκρίνονται σε δευτερόλεπτα, αναλύοντας το κείμενο των σελίδων αυτών, αλλά τα αποτελέσματά τους δεν είναι ιδιαίτερα καλά για το χρήστη. Επομένως θα ήταν ιδιαίτερα σημαντικό το όφελος από ένα εργαλείο αναζήτησης που αν και θα είχε μεγαλύτερο χρόνο απόκρισης, σε σχέση με τις κλασσικές μηχανές, τα αποτελέσματα του όμως θα εμφάνιζαν αποτελέσματα καλύτερης *ποιότητας* για το χρήστη και θα ανταποκρίνονται στην ανθρώπινη έννοια της ποιότητας.

Η αναζήτηση, σε μία κλασσική μηχανή, ξεκινάει από την παροχή ενός ερωτήματος (query) από το χρήστη. Είναι προτιμότερο να μην θεωρηθεί μία γενικότερη μορφή του ερωτήματος, καθώς υπάρχουν περισσότεροι από ένας τύποι ερωτημάτων, και ο χειρισμός του κάθε τύπου είναι πιθανό να απαιτεί διαφορετικές τεχνικές. Μπορούμε να διακρίνουμε τις ακόλουθες κατηγορίες ερωτημάτων:

- Ειδικά ερωτήματα. πχ. «Ο Netscape υποστηρίζει το JD 1.1 code-signing API;»
- Ερωτήματα ευρέων θεμάτων. πχ. «Βρες πληροφορίες για την προγραμματιστική γλώσσα JAVA.»
- Ερωτήματα ομοιότητας σελίδων. πχ. «Βρες σελίδες όμοιες με την java.sun.com»

Παρατηρώντας τους πρώτους δύο μόνο τύπους ερωτημάτων παρουσιάζονται δύο τελείως διαφορετικοί τύποι προβλημάτων. Η δυσκολία στο χειρισμό ειδικών ερωτημάτων σχετίζεται με το λεγόμενο Scarcity Problem (πρόβλημα σπανιότητας), το οποίο εμφανίζεται όταν υπάρχουν πολύ λίγες σελίδες που περιέχουν την απαιτούμενη πληροφορία και είναι συχνά πολύ δύσκολο να βρεθεί η ταυτότητα αυτών των σελίδων.

Για τα ερωτήματα ευρέων θεμάτων από την άλλη πλευρά, είναι αναμενόμενο να βρεθούν πολλές χιλιάδες σχετικές σελίδες στο διαδίκτυο. Σε αυτή την περίπτωση προφανώς δεν τίθεται θέμα έλλειψης. Η κυριότερη δυσκολία στην περίπτωση αυτή προκύπτει από το πρόβλημα που αναφέρεται ως Abundance Problem (πρόβλημα αφθονίας), το οποίο εμφανίζεται

όταν ο αριθμός των σελίδων που λογικά μπορούν να επιστραφούν ως σχετικές είναι πολύ μεγάλος για να τις αφομοιώσει ένας άνθρωπος. Για να βρεθούν αποτελεσματικές μέθοδοι αναζήτησης κάτω από αυτές τις συνθήκες, χρειάζεται ένας τρόπος να φιλτραριστεί το τεράστιο σύνολο σχετικών σελίδων για να προκύψει ένα μικρό σύνολο των πιο σημαντικών σελίδων.

Η έννοια της authority (αξιόπιστη-αντιπροσωπευτική) σελίδας, χρησιμοποιείται σαν μία βασική έννοια στην τεχνική αυτή. Ένα από τα βασικά θέματα εδώ είναι ο προσδιορισμός των χαρακτηριστικών μία σελίδας που θα μπορούσαν να την καταστήσουν αντιπροσωπευτική και ποιοτική για ένα θέμα.

Αξίζει να αναφερθούν μερικά από τα προβλήματα που προκύπτουν. Το πρώτο πρόβλημα είναι ότι θεωρείται αναμενόμενο, να χαρακτηριστεί η κεντρική σελίδα του πανεπιστημίου του Harvard, www.harvard.edu, σαν μία από τις πιο authoritative σελίδες για το ερώτημα Harvard. Δυστυχώς όμως υπάρχουν εκατομμύρια σελίδες στο διαδίκτυο που χρησιμοποιούν τον όρο Harvard και η www.harvard.edu δεν είναι αυτή που χρησιμοποιεί τον όρο πιο συχνά, ούτε με τον πιο φανερό τρόπο ούτε και με κανέναν άλλο τρόπο ο οποίος θα μπορούσε να την ευνοήσει αν χρησιμοποιηθεί μία συνάρτηση αξιολόγησης που στηρίζεται στο κείμενο. Έτσι υποδηλώνεται το γεγονός ότι δεν υπάρχει μία απόλυτη ενδογενής αξιολόγηση της σελίδας που θα μπορούσε να επιτρέψει σε κάποιο εργαλείο απλά αναλύοντας τη σελίδα να εκτιμήσει σωστά το κατά πόσο είναι μία καλή authority σελίδα.

Το δεύτερο πρόβλημα είναι, για παράδειγμα, το να βρεθούν οι κεντρικές σελίδες των βασικών μηχανών αναζήτησης. Ξεκινώντας με το ερώτημα «μηχανή αναζήτησης», προκύπτει μία άμεση δυσκολία που στηρίζεται στο γεγονός ότι πολλές από τις πραγματικά authoritative σελίδες όπως το Yahoo!, Excite, AltaVista, δεν χρησιμοποιούν αυτό τον όρο στις σελίδες τους. Αυτό είναι ένα ουσιώδες και επαναλαμβανόμενο φαινόμενο. Ακόμα ένα παράδειγμα του φαινομένου αυτού είναι το ότι δεν είναι αναμενόμενο οι σελίδες της Honda ή της Toyota να περιέχουν τον όρο «κατασκευαστές αυτοκινήτων».

Ένα τρίτο πρόβλημα είναι ότι υπάρχουν πολλές φορές σελίδες που αναφέρονται στο ίδιο περιεχόμενο, αλλά δεν υπάρχει διασύνδεση μεταξύ τους, γεγονός το οποίο μπορεί να οφείλεται σε διάφορους λόγους. Ένας βασικός λόγος της μη ύπαρξης των διασυνδέσεων αυτών είναι η άγνοια ύπαρξης των σελίδων από τους δημιουργούς. Πιο συγκεκριμένα είναι σχεδόν αδύνατο ένα άτομο που δημιουργεί μία σελίδα για κάποιο θέμα να μπορεί να γνωρίζει όλες τις σελίδες που περιέχουν σχετικό περιεχόμενο. Ένας άλλος βασικός λόγος είναι ότι μπορεί να μην είναι επιθυμητή η ύπαρξη των διασυνδέσεων αυτών και αυτό οφείλεται σε επαγγελματικούς, ανταγωνιστικούς ή και διαφημιστικούς ακόμα λόγους. Για παράδειγμα είναι σχεδόν απίθανο να επιθυμεί η Apple να βρίσκεται σε μία σελίδα της μία διασύνδεση προς μία σελίδα της IBM, παρόλο που το περιεχόμενο και των δύο σελίδων είναι πιθανό να σχετίζεται άμεσα και να αναφέρεται στο ίδιο θέμα.

Περιγραφή της Τεχνικής

Κάθε σύνολο από διασυνδεδεμένες σελίδες μπορεί να αναπαρασταθεί σαν ένας κατευθυνόμενος γράφος $G = (V, E)$, όπου για κάθε σελίδα του συνόλου υπάρχει ένας κόμβος στο γράφο και για κάθε διασύνδεση από τη σελίδα p στην q υπάρχει μία κατευθυνόμενη ακμή από τον κόμβο p στον q. Έτσι ο βαθμός εξόδου (*out-degree*) ενός κόμβου p είναι ο αριθμός των σελίδων προς τις οποίες η σελίδα p έχει υπερδεσμό και βαθμός εισόδου (in-degree) ενός κόμβου p είναι ο αριθμός των σελίδων που έχουν υπερδεσμό προς τη σελίδα p.

Από το γράφο G είναι δυνατό να απομονωθεί ένας υπογράφος ακολουθώντας την ακόλουθη διαδικασία. Εάν το W είναι ένα υποσύνολο των σελίδων V του γράφου τότε ως $G[W]$ ορίζεται ο γράφος που προκύπτει από αυτό το σύνολο των σελίδων. Έτσι ο $G[W]$ περιέχει τόσους κόμβους όσες οι σελίδες του W και ακμές αυτές που προκύπτουν από τις αντίστοιχες διασυνδέσεις μεταξύ των σελίδων του W.

Ας υποθέσουμε ότι δίνεται ως είσοδος στο σύστημα ένα ερώτημα με ευρύ θέμα καθορισμένο από τον όρο αναζήτησης σ. Καθώς η τεχνική δεν έχει νόημα να εφαρμοστεί σε

όλες τις σελίδες του διαδικτύου, αλλά σε ένα μόνο κομμάτι του σχετικό με το θέμα του ερωτήματος, πρώτα πρέπει να επιλεγεί αυτό το υποσύνολο σελίδων του διαδικτύου.

Μία πρώτη ιδέα θα ήταν να επιλεγεί το σύνολο όλων των σελίδων Q_σ που περιέχουν τον όρο του ερωτήματος. Αυτή η μέθοδος έχει όμως δύο σημαντικά μειονεκτήματα. Πρώτον αυτό το σύνολο είναι πολύ πιθανό να περιέχει περισσότερες από ένα εκατομμύριο σελίδες και να αυξήσει τόσο το υπολογιστικό κόστος που να είναι ανέφικτη η εκτέλεση του αλγορίθμου, και δεύτερον πιθανότατα πολλές από τις authorities σελίδες να μην περιέχονται σε αυτό το σύνολο.

Βάση των παραπάνω προκύπτει το ότι πρέπει να αποκτηθεί πρώτα ένα σύνολο από σελίδες, το S_σ, το οποίο θα ικανοποιεί τις ακόλουθες απαιτήσεις:

i. Να είναι σχετικά μικρό.
ii. Να είναι πλούσιο σε σχετικές με το θέμα σελίδες.
iii. Να περιέχει τις περισσότερες ή έστω πολλές από τις authority σελίδες.

Καθώς το $\underline{S_\sigma}$ διατηρείται μικρό σε μέγεθος το υπολογιστικό κόστος για την εφαρμογή του αλγορίθμου σε αυτό διατηρείται σε σχετικά μικρά μεγέθη. Επίσης εξασφαλίζοντας ότι το σύνολο αυτό είναι πλούσιο σε σχετικές με το θέμα σελίδες γίνεται πιο εύκολη η εύρεση καλών authorities. Επομένως το πρόβλημα που προκύπτει είναι η εύρεση ενός τέτοιου συνόλου.

Εύρεση Βασικού Συνόλου Σελίδων

Για να δημιουργηθεί το βασικό σύνολο σελίδων που απαιτείται, αρχικά δημιουργείται ένα αρχικό σύνολο το R_σ, το οποίο περιέχει τις πρώτες t (έστω ότι το t είναι 200) σελίδες που δίνει σαν αποτέλεσμα μία *term-based* μηχανή αναζήτησης όπως για παράδειγμα η AltaVista δίνοντάς της σαν είσοδο τον όρο σ. Αυτό το σύνολο είναι εμφανές ότι ικανοποιεί την απαίτηση (i), καθώς το μέγεθός του μπορεί εύκολα να καθοριστεί από την παράμετρο t. Επίσης και η απαίτηση (ii) ικανοποιείται καθώς το R_σ είναι ένα υποσύνολο του Q_σ που είναι η συλλογή όλων των σελίδων που περιέχουν τον όρο σ. Από την άλλη πλευρά όμως το σύνολο αυτό απέχει πολύ από το να ικανοποιεί και την τρίτη απαίτηση, καθώς ακόμα και το σύνολο Q_σ δεν την ικανοποιεί.

Όμως δεν είναι ιδιαίτερα δύσκολο να καταλήξουμε σε ένα σύνολο S_σ, χρησιμοποιώντας το R_σ, το οποίο να ικανοποιεί και την τρίτη απαίτηση. Θεωρώντας ότι ένα καλό authority για το συγκεκριμένο θέμα δεν περιέχεται στο σύνολο R_σ είναι πολύ πιθανό να δείχνεται από τουλάχιστον μία σελίδα του R_σ. Έτσι ο αριθμός των καλών authorities μπορεί να αυξηθεί επεκτείνοντας το σύνολο R_σ προσθέτοντας τις σελίδες που δείχνουν σε σελίδες αυτού του συνόλου και αυτές που δείχνονται από σελίδες του R_σ. Η διαδικασία που ακολουθείται περιγράφεται στη συνέχεια.

Subgraph(σ,E,t,d)

σ: a query string

E: a text-based search engine

t, d: natural numbers

Let R_σ denote the top t results of E on σ

Set $S_\sigma = R_\sigma$

For each page p that belongs to R_σ

    Let $\Gamma^+(p)$ denote the set of all pages p points to

    Let $\Gamma^-(p)$ denote the set of all pages pointing to p

    Add all pages in $\Gamma^+(p)$ to S_σ

    If $\Gamma^-(p) <= d$ then

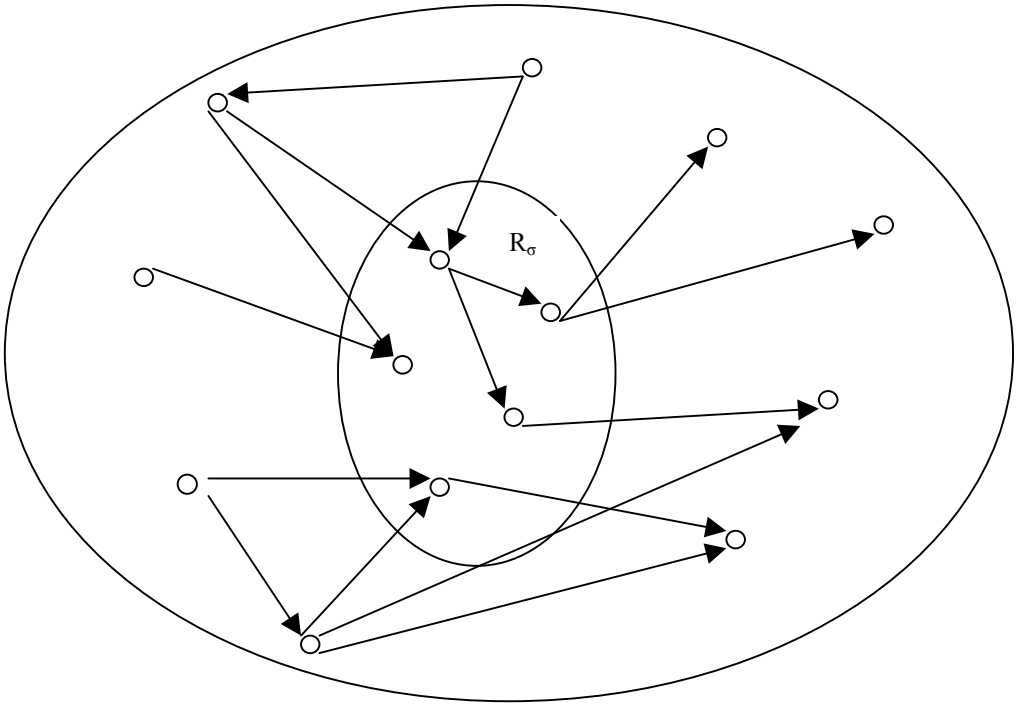
Add all pages in Γ(p) to S_σ

Else

Add an arbitrary set of d pages from Γ(p) to S_σ

End

Return S_σ



Σχήμα 4 – Επέκταση του αρχικού συνόλου σελίδων στο βασικό σύνολο

Έτσι τελικά προκύπτει το σύνολο S_σ μεγαλώνοντας το αρχικό σύνολο R_σ, περιλαμβάνοντας σε αυτό κάθε σελίδα προς την οποία υπάρχει διασύνδεση από σελίδα του συνόλου R_σ και κάθε σελίδα από την οποία υπάρχει διασύνδεση προς κάποια σελίδα του συνόλου R_σ, με την προϋπόθεση ότι μέχρι *d* σελίδες μπορούν να προστεθούν στο σύνολο S_σ που δείχνουν σε μία μόνο σελίδα του R_σ. Η προϋπόθεση αυτή είναι ιδιαίτερα σημαντική, καθώς μπορεί να υπάρχει ένας πολύ μεγάλος αριθμός σελίδων που περιέχουν διασύνδεση προς μία σελίδα, και είναι εμφανές ότι δεν είναι δυνατό όλες αυτές οι σελίδες να συμπεριληφθούν στο σύνολο S_σ, του οποίου το μέγεθος απαιτείται να είναι σχετικά μικρό. Το σύνολο S_σ αναφέρεται σαν το base set του ερωτήματος *σ*.

Από τις σελίδες που ανήκουν στο σύνολο S_σ προκύπτει ένας γράφος ο $G[S_\sigma]$, στον οποίο κόμβοι είναι οι σελίδες και ακμές οι διασυνδέσεις που συνδέουν τις σελίδες του συνόλου. Καθώς υπάρχουν στις σελίδες διασυνδέσεις οι οποίες δεν μεταφέρουν κάποια σημαντική πληροφορία, αλλά απλά διευκολύνουν την πλοήγηση του χρήστη, προτείνεται μία ευρεστική μέθοδος, η οποία σκοπό έχει να αντισταθμίσει το αποτέλεσμα των διασυνδέσεων αυτών. Επομένως οι ακμές που υπάρχουν στο γράφο $G[S_\sigma]$ χωρίζονται σε δύο κατηγορίες.

Μία ακμή χαρακτηρίζεται εγκάρσια (*transverse*) εάν συνδέει δύο σελίδες οι οποίες ανήκουν σε διαφορετικό *domain*, και *φυσική* (*intrinsic*) εάν συνδέει δύο σελίδες που βρίσκονται στο ίδιο *domain*. Το *domain* είναι το πρώτο επίπεδο της διεύθυνσης, η οποία σχετίζεται με κάποια σελίδα. Καθώς οι φυσικές ακμές είναι αυτές που διευκολύνουν την πλοήγηση μέσα σε έναν διαδικτυακό κόμβο, προκύπτει ότι μεταφέρουν πολύ λιγότερη πληροφορία για τη σπουδαιότητα και την ποιότητα της σελίδας προς την οποία δείχνουν από ότι οι εγκάρσιες ακμές. Για αυτό το λόγο οι φυσικές ακμές του γράφου αφαιρούνται με αποτέλεσμα να μένουν σε αυτόν μόνο οι εγκάρσιες ακμές. Ο γράφος που προκύπτει τελικά είναι ο G_σ. Η μέθοδος αυτή της διαγραφής των φυσικών ακμών, είναι μεν ιδιαίτερα απλή, είναι όμως και αποτελεσματική.

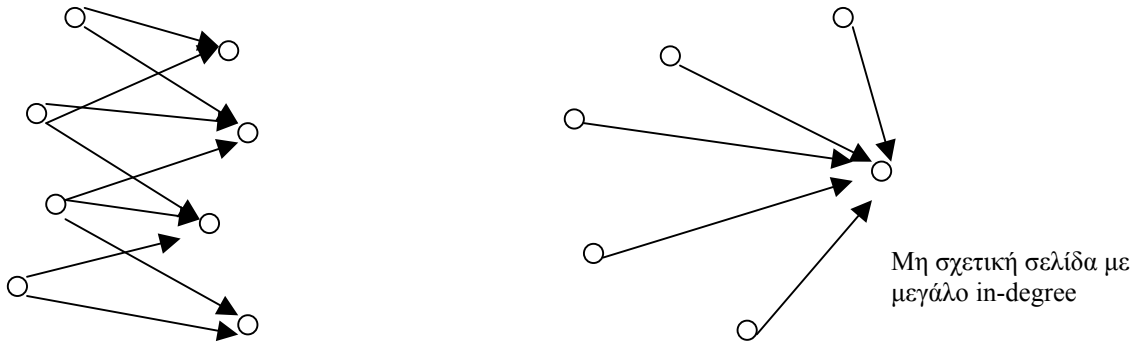
Υπολογισμός των Βαρών των Hubs και των Authorities

Ο γράφος G_σ που έχει δημιουργηθεί περιέχει πολλές σχετικές με το ερώτημα σελίδες και αρκετές σημαντικές σελίδες. Αυτό που χρειάζεται στη συνέχεια είναι να βρεθούν αυτές οι σημαντικές σελίδες, αναλύοντας τη δομή των ακμών του γράφου αυτού.

Μία απλή προσέγγιση είναι η ταξινόμηση των σελίδων βάση του in-degree, του αριθμού των ακμών που δείχνουν στη συγκεκριμένη σελίδα. Η ιδέα αυτή είχε απορριφθεί για το σύνολο όλων των σελίδων που περιέχουν το ερώτημα *σ*. Αλλά σε αυτή τη φάση ο γράφος που έχει δημιουργηθεί είναι χαρακτηριστικά μικρότερος και περιέχει πολύ περισσότερες σημαντικές σελίδες, προς τις οποίες υπάρχουν πολλές ακμές.

Παρόλο που αυτή η προσέγγιση δίνει καλύτερα αποτελέσματα για το γράφο από ότι για το σύνολο όλων των σελίδων, εφαρμόζοντάς τη στο γράφο μπορεί να δημιουργήσει σημαντικά προβλήματα. Αυτό συμβαίνει γιατί δεν διαχωρίζει τις σημαντικές σελίδες, σε σχέση με το ερώτημα, που υπάρχουν στο γράφο, από τις γενικότερα δημοφιλείς σελίδες, καθώς και οι δύο αυτοί τύποι σελίδων έχουν μεγάλο in-degree.

Το πρόβλημα αυτό μπορεί να αντιμετωπιστί με την παρατήρηση ότι οι authoritative σελίδες που είναι σχετικές με το ερώτημα *σ* του χρήστη δεν απαιτείται να έχουν μόνο μεγάλο in-degree, αλλά και να έχουν αρκετά κοινά χαρακτηριστικά με τα σύνολα των σελίδων που δείχνουν προς αυτές. Επομένως εκτός από τις authoritative σελίδες θα πρέπει να προσδιοριστούν και ένα σύνολο άλλων σελίδων, οι λεγόμενες hub σελίδες, οι οποίες έχουν διασυνδέσεις προς τις authoritative σελίδες. Οι σελίδες αυτές συνενώνουν κατά κάποιο τρόπο τις authorities σε ένα κοινό θέμα, αγνοώντας σελίδες που απλά έχουν μεγάλο in-degree. Ένα παράδειγμα αυτής της συνένωσης των authorities σελίδων από τις hub σελίδες φαίνεται στο Σχήμα 5.



Σχήμα 5 – Ένα ισχυρά συνδεδεμένο σύνολο σελίδων hubs και authorities

Οι σελίδες hubs και authorities υποδηλώνουν ένα είδος σχέσης αμοιβαίας ενίσχυσης, καθώς ένα καλό hub είναι μία σελίδα που δείχνει σε πολλά καλά authorities και ένα καλό authority είναι μία σελίδα που δείχνεται από πολλά καλά hubs. Επομένως για να βρεθούν αυτά τα σύνολο σελίδων πρέπει να βρεθεί μία μέθοδος η οποία να μπορεί να ανιχνεύσει αυτή τη σχέση, στο σύνολο G_σ.

Ένας Επαναληπτικός Αλγόριθμος

Ο επαναληπτικός αλγόριθμος που θα περιγραφεί στη συνέχεια, και ο οποίος υπολογίζει και ενημερώνει τα βάρη των τιμών hub και authority για κάθε σελίδα, εκμεταλλεύεται αυτή την αμοιβαία σχέση των hubs και authorities σελίδων. Με κάθε σελίδα του γράφου συνδέονται δύο μη αρνητικά βάρη, το authority βάρος $x^{<p>}$ και το hub βάρος $y^{<p>}$. Κανονικοποιώντας τα βάρη κάθε τύπου ξεχωριστά έτσι ώστε το άθροισμα των τετραγώνων τους να είναι ίσα με τη μονάδα, παρατηρείται ότι οι σελίδες με τις μεγαλύτερες τιμές για αυτά τα βάρη είναι τα καλύτερα *authorities* και *hubs* αντίστοιχα.

Αριθμητικά αυτή η σχέση αμοιβαίας ενίσχυσης αναπαριστάνεται ως εξής: εάν η σελίδα *p* δείχνει σε πολλές σελίδες με μεγάλες τιμές για το βάρος *x* (*authority*), τότε είναι αναμενόμενο να αποκτήσει μεγάλη τιμή για το βάρος *y* (*hub*). Ανάλογα εάν η σελίδα *p* δείχνεται από πολλές σελίδες με μεγάλες τιμές για το βάρος *y* είναι αναμενόμενο να αποκτήσει μεγάλη τιμή για το βάρος *x*. Αυτή η προσέγγιση ωθεί προς τον ορισμό δύο συναρτήσεων που εφαρμόζονται στα βάρη *x* και *y,* οι οποίες αναφέρονται σαν *I* και *O*. Έτσι η συνάρτηση *I* ενημερώνει τα βάρη *x* ως εξής:

$$x^{<p>} \longleftarrow \sum_{q:(q,p)\in E} y^{<q>}$$

Ενώ η συνάρτηση *O* ενημερώνει τα βάρη y ως εξής:

$$y^{<p>} \longleftarrow \sum_{q:(q,p)\in E} x^{<q>}$$

Για να βρεθούν οι τιμές ισορρόπησης για τα βάρη, αρκεί να εφαρμοστούν οι συναρτήσεις *I* και *O* διαδοχικά αρκετές φορές μέχρι οι τιμές να σταθεροποιηθούν. Το σύνολο των βαρών *x* αναπαριστάνεται με ένα διάνυσμα όπου κάθε συντεταγμένη αντιστοιχεί σε μία σελίδα, και αντίστοιχα το σύνολο των βαρών *y* με ένα άλλο διάνυσμα. Ο αλγόριθμος που καλείται είναι ο ακόλουθος.

Iterate(G, k)

G: a collection of n linked pages

k: a natural number

Let z denote the vector (1, 1, 1, ..., 1) $\in R^n$.

Set x_0 :=z.

Set y_0 :=z.

For i =1, 2, ..., k

Apply the I operation to (x_{i-1} , y_{i-1}) ,obtaining new -weights x_i' .

Apply the O operation to (x_i' ,y_{i-1}) ,obtaining new y -weights y_i' .

Normalize x_i',obtaining x_i .

Normalize y_i' ,obtaining y_i .

End

Return (x_k ,y_k).

Στη συνέχεια εφαρμόζεται μία συνάρτηση φιλτραρίσματος η οποία επιστρέφει τις *c* καλύτερες *authority* σελίδες και τις *c* καλύτερες *hub* σελίδες. Ο αλγόριθμος για αυτό το φιλτράρισμα είναι ο ακόλουθος:

Filter(G, k, c)

G: a collection of n linked pages

k,c: natural numbers

(x_k, y_k) := Iterate(G, k)

Report the pages with the c largest coordinates in x_k as authorities

Report the pages with the c largest coordinates in y_k as hubs

Χρησιμοποιώντας τεχνικές της γραμμικής άλγεβρας αποδεικνύεται ότι όσο αυξάνεται ο αριθμός *k*, δηλαδή όσο πιο πολλές φορές εκτελείται ο αλγόριθμος Iterate, τόσο οι τιμές των βαρών τείνουν να σταθεροποιηθούν.

Από τις τεχνικές αυτές προκύπτει επίσης μία σημαντική παρατήρηση σε σχέση με τα «τελικά» βάρη. Έστω *A,* ο πίνακας γειτνίασης του γράφου των σελίδων δηλαδή ο πίνακας που προκύπτει θέτοντας την τιμή 1 στη θέση *(i,j)* αν υπάρχει ακμή στο γράφο G_σ από τη σελίδα p_i στη σελίδα p_j και την τιμή 0 στις υπόλοιπες θέσεις του πίνακα. Εύκολα μπορεί να δειχτεί ότι οι συναρτήσεις *I* και *O* χρησιμοποιώντας τον πίνακα *A* μπορούν να γραφτούν ως εξής:

$$x \leftarrow A^T y \text{ και } y \leftarrow Ax \text{ αντίστοιχα.}$$

Έτσι παρατηρείται ότι το τελικό διάνυσμα *x* στο οποίο σταθεροποιείται ο αλγόριθμος *Iterate* είναι το πρωτεύον ιδιοδιάνυσμα του πίνακα A^TA και αντίστοιχα το τελικό διάνυσμα *y* είναι το πρωτεύον ιδιοδιάνυσμα του πίνακα AA^T.

Μετά από μερικές εκτελέσεις του αλγόριθμου *Iterate* προκύπτει ότι ο αλγόριθμος συγκλίνει αρκετά γρήγορα στις τελικές τιμές των διανυσμάτων *x* και *y*, καθώς αρκούν 20 επαναλήψεις. Από την παρατήρηση που προέκυψε παραπάνω μπορεί κανείς να θεωρήσει ότι αρκεί να βρεθούν τα ιδιοδιανύσματα των ανωτέρω πινάκων για να βρεθούν και οι τελικές τιμές των βαρών. Η εύρεση των ιδιοδιανυσμάτων όμως δεν είναι εύκολη διεργασία.

Τελικά προτιμάται η χρήση του αλγόριθμου *Iterate* για δύο λόγους. Πρώτον, ο αλγόριθμος αυτός υποδηλώνει την ώθηση σε αυτή την προσέγγιση λόγω της αμοιβαίας ενίσχυσης που προκύπτει από τις συναρτήσεις *I* και *O*. Δεύτερον, δεν χρειάζεται να εκτελεστεί ο αλγόριθμος αυτό μέχρι να συγκλίνει, καθώς αρκεί για να υπολογιστούν τα διανύσματα των βαρών να αρχικοποιηθούν και στη συνέχεια να εφαρμοστεί ένας καθορισμένος μικρός αριθμός διαδοχικών επαναλήψεων των συναρτήσεων *I* και *O*.

Ερωτήματα που ζητούν όμοιες σελίδες

Ο αλγόριθμος που παρουσιάστηκε παραπάνω για την εύρεση καλών σελίδων σχετικών με το ερώτημα που θέτει ένας χρήστης σε μία μηχανή αναζήτησης μπορεί να χρησιμοποιηθεί και σε έναν άλλο τύπο προβλήματος. Η δομή των διασυνδέσεων παρέχει μία διαφορετική αντίληψη για την ομοιότητα ανάμεσα στις σελίδες. Για παράδειγμα ο χρήστης μπορεί να γνωρίζει μία καλή σελίδα, κατά την προσωπική του άποψη, για κάποιο θέμα και να επιθυμεί να βρει και άλλες τέτοιες σελίδες.

Εάν η σελίδα αυτή που δίνεται αρχικά από το χρήστη δείχνεται από πολλές άλλες σελίδες τότε προκύπτει το πρόβλημα της αφθονίας. Πιο συγκεκριμένα η δομή των διασυνδέσεων, που υπάρχουν, αναπαριστάνει έναν τεράστιο αριθμό από διαφορετικές γνώμες για τη σχέση της συγκεκριμένης σελίδας με τις άλλες σελίδες. Χρησιμοποιώντας την ιδέα των hubs και των authorities, δίνεται μία άλλη προσέγγιση για το θέμα της ομοιότητας. Σύμφωνα με αυτή την προσέγγιση αρκεί να αναζητηθεί εάν υπάρχουν άλλες καλές authorities στην περιοχή από τη δομή των διασυνδέσεων κοντά στην αρχική σελίδα.

Ο αλγόριθμος που προτείνεται παρουσιάζει μία μόνο σημαντική διαφορά με αυτόν που προτείνεται όταν δίνεται ένα σύνολο λέξεων για ερώτημα. Και σε αυτή την περίπτωση ζητείται ένα αρχικό σύνολο σελίδων από μία μηχανή αναζήτηση βάση ενός ερωτήματος, αλλάζει όμως το ερώτημα. Το ερώτημα που δίνεται στη μηχανή είναι να βρεθούν οι σελίδες που δείχνουν προς τη συγκεκριμένη σελίδα που δίνεται. Το σύνολο R_σ που προκύπτει αυξάνεται με τον ίδιο τρόπο για να προκύψει το βασικό σύνολο S_σ.

Πολλαπλά σύνολα από σελίδες hubs και authorities

Από τον τρόπο υπολογισμού των σελίδων hubs και authorities προκύπτει το πιο ισχυρά συνδεδεμένο σύνολο hubs και authorities που εμφανίζεται στο γράφο G_σ. Όμως υπάρχουν διάφοροι λόγοι για τους οποίους είναι δυνατή η εμφάνιση στο γράφο αυτό διαφόρων ισχυρά συνδεδεμένων συνόλων σελίδων τα οποία όμως να είναι καλά χωρισμένα μεταξύ τους. Μερικοί τέτοιοι λόγοι είναι και οι ακόλουθοι:

- Το ερώτημα να έχει διάφορες σημασίες όπως η λέξη «jaguar», η οποία μπορεί να σημαίνει είτε το ζώο είτε το αυτοκίνητο.
- Το ερώτημα μπορεί να εμφανίζεται σαν τεχνικός όρος στο περιεχόμενο ποικίλλων τεχνικών κοινωνιών του διαδικτύου, όπως ο όρος «randomized algorithms».
- Το ερώτημα μπορεί να αναφέρεται σε ένα θέμα ιδιαίτερα πολωμένο, το οποίο εμφανίζεται σε διάφορα σύνολα τα οποία πιθανότατα δεν δείχνουν το ένα το άλλο, όπως το θέμα «abortion».

Σε κάθε ένα από τα προαναφερόμενα παραδείγματα, οι σχετικές σελίδες που υπάρχουν μπορεί να είναι φυσικά διαχωρισμένες σε διαφορετικά σύνολα. Το ερώτημα που προκύπτει επομένως είναι πώς μπορούν να διαχωριστούν αυτά τα σύνολα και τελικά στο χρήστη να παρουσιαστούν σελίδες για όλα αυτά τα διαφορετικά σύνολα.

Όπως αναφέρθηκε και παραπάνω, οι σελίδες hubs και authorities που προκύπτουν συσχετίζονται με το πρωτεύον ιδιοδιάνυσμα των πινάκων A^TA και AA^T, όπου A είναι ο πίνακας γειτνίασης του γράφου G_σ. Επομένως είναι αναμενόμενο τα μη πρωτεύοντα ιδιοδιανύσματα των πινάκων αυτών να σχετίζονται με τα μικρότερα ισχυρά συνδεδεμένα σύνολα *hubs* και *authorities* που εμφανίζονται στο βασικό σύνολο S_σ.

Αξίζει να σημειωθεί ότι τα μη πρωτεύον ιδιοδιανύσματα μπορεί να περιέχουν και αρνητικές τιμές σε αντίθεση με το πρωτεύον ιδιοδιάνυσμα που μπορεί να περιέχει μόνο θετικές τιμές. Επομένως έτσι μπορούν να προκύψουν δύο σύνολα ισχυρά συνδεδεμένων σελίδων, καθώς στο ένα θα περιέχονται οι σελίδες με τις μεγαλύτερες θετικές τιμές και στο άλλο οι σελίδες με τις μεγαλύτερες αρνητικές τιμές. Επίσης μπορεί να παρουσιαστεί και το φαινόμενο οι σελίδες με τις μεγαλύτερες τιμές να εμφανίζονται σε αρκετά από τα πρώτα μη πρωτεύοντα ιδιοδιανύσματα, επομένως μπορεί από αρκετά από τα πρώτα ισχυρά μη πρωτεύοντα ιδιοδιανύσματα να προκύψουν τα ίδια σύνολα από σελίδες *hubs* και *authorities*.

Διάχυση και Γενίκευση

Για να δώσει αρκετά καλά αποτελέσματα η τεχνική που παρουσιάστηκε στηρίζεται στο γεγονός ότι ο γράφος G_σ περιέχει αρκετές σχετικές με το θέμα σελίδες. Όμως αυτό δεν συμβαίνει πάντα. Στις περιπτώσεις που το ερώτημα που θέτει ο χρήστης υποδηλώνει ένα θέμα που δεν είναι αρκετά γενικό και ευρύ ο γράφος που προκύπτει δεν περιέχει αρκετές σελίδες σχετικές με το θέμα, έτσι ώστε να είναι δυνατό να προκύψει ένα αρκετά ισχυρό σύνολο σελίδων.

Παράλληλα είναι αρκετά πιθανό να υπάρχουν πολλές σελίδες σχετικές με ένα ή περισσότερα θέματα πιο γενικά, από αυτό που υποδηλώνεται από το ερώτημα. Επομένως ένα από τα σύνολα αυτών των σελίδων θα είναι αρκετά ισχυρό και θα περιέχει πολλές καλές σελίδες με αποτέλεσμα να είναι αυτό το σύνολο που θα επιστραφεί από τον αλγόριθμο. Έτσι

παρατηρείται μία «διάχυση» του βασικού θέματος που παρουσιάζεται από την τεχνική, καθώς δεν έχει ιδιαίτερη σχέση με το ερώτημα του χρήστη.

Παρόλο που η διάχυση αυτή περιορίζει τις δυνατότητες της τεχνικής για την εύρεση καλών σελίδων για ειδικευμένα ερωτήματα, υποδηλώνει μία άλλη της δυνατότητα

5.2.2 Η Μηχανή Αναζήτησης Google

Μία άλλη ιδιαίτερα σημαντική τεχνική αναζήτησης στο διαδίκτυο είναι η τεχνική που χρησιμοποιείται στη μεγάλης-κλίμακας μηχανή αναζήτησης Google [BP98].

Το Google έχει δύο σημαντικά και αξιοπρόσεχτα χαρακτηριστικά, τα οποία επιτυγχάνουν αποτελέσματα μεγάλης ακρίβειας. Το πρώτο είναι ότι το Google χρησιμοποιεί τη δομή του διαδικτύου που προκύπτει από τις διασυνδέσεις για να υπολογίσει ένα βαθμό ποιότητας για κάθε σελίδα του διαδικτύου. Αυτή η μέθοδος βαθμολόγησης ονομάζεται *PageRank*. Το δεύτερο χαρακτηριστικό αυτής της μηχανής αναζήτησης είναι ότι χρησιμοποιεί τις διασυνδέσεις για να βελτιώσει τα αποτελέσματα της αναζήτησης.

Η Μέθοδος Βαθμολόγησης PageRank

Το PageRank ορίζεται ως εξής::

Έστω A μία σελίδα προς την οποία δείχνουν οι σελίδες $T_1...T_n$. Η παράμετρος d είναι παράγοντας απόσβεσης, ο οποίος μπορεί να πάρει μία τιμή μεταξύ του 0 και του 1. Συνήθως τίθεται το d ίσο με το 0.85. Επίσης ορίζεται το C(A) ως ο αριθμός των διασυνδέσεων που περιέχει η σελίδα A. Τότε το PageRank της σελίδας A δίνεται από τον παρακάτω τύπο:

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

Σημειώνεται ότι οι βαθμοί PageRanks αποτελούν μία πιθανοτική κατανομή στις σελίδες του διαδικτύου, με αποτέλεσμα το άθροισμα όλων των PageRanks των σελίδων του διαδικτύου είναι ίσο με το ένα.

Τα *PageRanks* μπορούν να υπολογιστούν με τη χρήση ενός απλού επαναληπτικού αλγορίθμου, και αντιστοιχούν στο πρωτεύον ιδιοδιάνυσμα του κανονικοποιημένου πίνακα γειτνίασης του γράφου του διαδικτύου.

Το *PageRank* μπορεί να θεωρηθεί ως ένα μοντέλο της συμπεριφοράς των χρηστών. Θεωρείται ότι υπάρχει ένας τυχαίος χρήστης που κινείται μέσα στο διαδίκτυο. Ο χρήστης αυτός ξεκινάει από μία τυχαία σελίδα που του δίνεται και συνεχίζει επιλέγοντας διασυνδέσεις, χωρίς όμως ποτέ να επιστρέφει σε προηγούμενη σελίδα επιλέγοντας την «επιστροφή» (back) από το διαφυλλιστή (browser). Κάποια στιγμή μπορεί να βαρεθεί και τότε ξεκινάει πάλι την ίδια διαδικασία από μία άλλη τυχαία σελίδα.

Η πιθανότητα να επισκεφτεί μία σελίδα ο χρήστης αυτός είναι το *PageRank* της σελίδας αυτής. Και ο παράγοντας απόσβεσης *d* είναι η πιθανότητα ο χρήστης να βαρεθεί τη σελίδα που βρίσκεται και να ζητήσει μία άλλη τυχαία. Μία σημαντική παραλλαγή είναι ο παράγοντας απόσβεσης *d* να αντιστοιχεί σε κάθε μία σελίδα χωριστά ή σε ένα σύνολο σελίδων. Αυτό επιτρέπει την εξατομίκευση και μπορεί να κάνει σχεδόν αδύνατο το γεγονός να παρασυρθεί το σύστημα με αποτέλεσμα να δώσει υψηλότερη βαθμολόγηση σε κάποιες σελίδες.

Μία άλλη δικαιολόγηση που προκύπτει από τη διαίσθηση είναι ότι μία σελίδα μπορεί να έχει υψηλή τιμή για το *PageRank* εάν υπάρχουν πολλές σελίδες που δείχνουν σε αυτή ή εάν υπάρχουν μερικές σελίδες που δείχνουν σε αυτή, οι οποίες όμως έχουν αυτές υψηλό *PageRank*. Διαισθητικά παρατηρείται ότι σελίδες που αναφέρονται από πολλές άλλες σελίδες στο διαδίκτυο περιέχουν αξιοπρόσεχτη πληροφορία. Επίσης σελίδες που αναφέρονται ακόμα και από μία μόνο άλλη σελίδα, η οποία όμως είναι ιδιαίτερα ποιοτική, όπως για παράδειγμα η αρχική σελίδα του Yahoo!, γενικά περιέχουν αξιοπρόσεχτη πληροφορία. Επομένως προκύπτει

ότι το PageRank χειρίζεται και τις δύο προαναφερόμενες περιπτώσεις, καθώς αναδρομικά διαδίδει τα βάρη μέσω της δομής του διαδικτύου που προκύπτει από τις διασυνδέσεις.

Το κείμενο των διασυνδέσεων (anchor text)

Στο Google το κείμενο των διασυνδέσεων αντιμετωπίζεται με ένα διαφορετικό τρόπο, από τις άλλες μηχανές αναζήτησης. Συνήθως το κείμενο αυτό σχετίζεται με τη σελίδα στην οποία βρίσκεται η διασύνδεση. Στο Google το κείμενο αυτό σχετίζεται και με τη σελίδα προς την οποία δείχνει η συγκεκριμένη διασύνδεση.

Αυτή η διπλή συσχέτιση του κειμένου παρέχει αρκετά πλεονεκτήματα. Πρώτον, τα κείμενα αυτά περιγράφουν ακριβέστερα τις σελίδες από ότι οι σελίδες οι ίδιες. Δεύτερον, μπορεί να υπάρχουν διασυνδέσεις προς σελίδες οι οποίες δεν έχουν κατηγοριοποιηθεί από μηχανές αναζήτησης που στηρίζονται στο κείμενο, όπως είναι οι εικόνες, τα προγράμματα, οι βάσεις δεδομένων.

Πρέπει να σημειωθεί ότι σελίδες οι οποίες δεν έχουν γίνει crawled από τη μηχανή μπορεί να δημιουργήσουν προβλήματα, καθώς δεν έχουν ελεγχθεί για την εγκυρότητα και την αξιοπιστία τους πριν επιστραφούν στο χρήστη. Δηλαδή σε αυτή την περίπτωση η μηχανή αναζήτησης μπορεί να επιστρέψει μία σελίδα η οποία μπορεί και να μην υπάρχει, αλλά απλά υπάρχει διασύνδεση προς αυτή. Καθώς όμως είναι δυνατό να ταξινομηθούν τα αποτελέσματα, το συγκεκριμένο αυτό πρόβλημα σπάνια εμφανίζεται.

Άλλα χαρακτηριστικά του συστήματος

Το σύστημα του Google αποτελείται και από άλλα χαρακτηριστικά, εκτός από αυτά που περιγράφηκαν παραπάνω. Πρώτον, διατηρεί πληροφορία για τη θέση στην οποία εμφανίζονται σημαντικές λέξεις μέσα στις σελίδες, με αποτέλεσμα να αξιοποιεί εύχρηστα τη γειτνίαση όρων κατά τη διάρκεια της αναζήτησης. Δεύτερον, διατηρεί στοιχεία για την οπτική αναπαράσταση κάποιων λεπτομερειών, όπως το μέγεθος των γραμμάτων των λέξεων. Έτσι σε λέξεις με μεγαλύτερα ή πιο έντονα γράμματα τους δίνεται μεγαλύτερο βάρος από ότι στις άλλες λέξεις. Τρίτον, όλος ο κώδικας HTML των σελίδων είναι διαθέσιμος σε έναν αποθηκευτικό χώρο.

Η αρχιτεκτονική του συστήματος

Στο *Google* η διαδικασία του *crawling* του διαδικτύου, δηλαδή η ανάκτηση σελίδων, γίνεται από αρκετούς κατανεμημένους *crawlers*. Υπάρχει ένας *URLserver*, ο οποίος στέλνει λίστες από διευθύνσεις (*urls*) για να ανακτηθούν από τους *crawlers*. Οι σελίδες που ανακτώνται στέλνονται στον *StoreServer*, ο οποίος τις συμπιέζει και τις αποθηκεύει σε ένα *Repository*. Κάθε σελίδα συσχετίζεται με έναν αριθμό *ID*, ο οποίος λέγεται *docID*, και ανατίθεται όποτε βρεθεί μία καινούρια διεύθυνση.

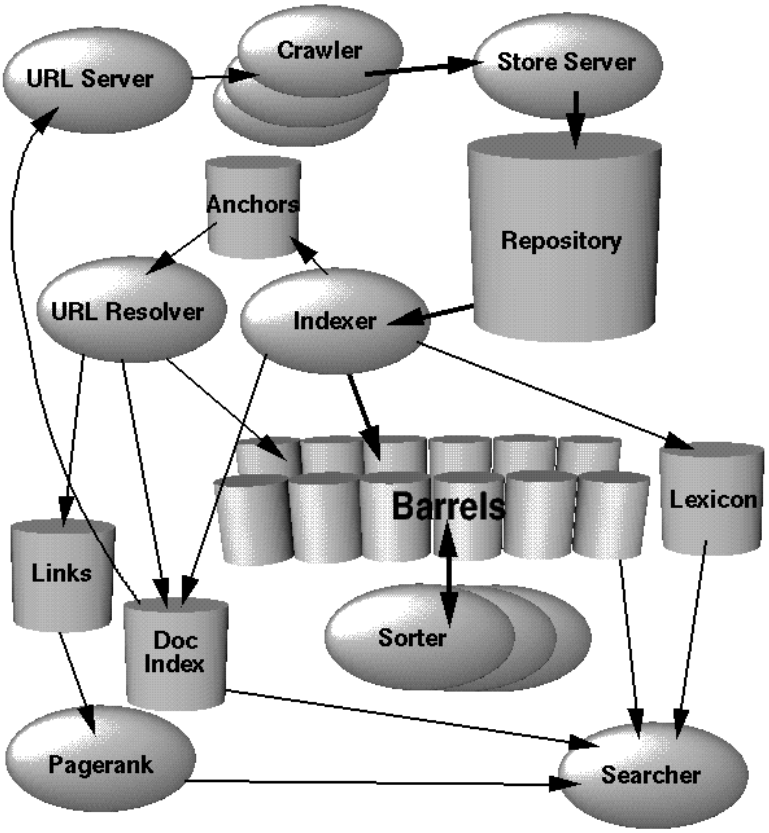
Η διαδικασία της κατηγοριοποίησης γίνεται από τον κατηγοριοποιητή (*indexer*) και τον ταξινομητή (*sorter*). Ο *indexer* προσπελαύνει το *repository*, αποσυμπιέζει τις σελίδες και τις αναλύει. Κάθε σελίδα μετατρέπεται σε ένα σύνολο από εμφανίσεις λέξεων που λέγονται *hits*. Για κάθε *hit* καταγράφεται η λέξη, η θέση της στη σελίδα και μία προσέγγιση του μεγέθους των γραμμάτων. Ο *indexer* κατανέμει αυτά τα *hits* σε ένα σύνολο από «*barrels*», δημιουργώντας με αυτόν τον τρόπο ένα μερικώς ταξινομημένο ευρετήριο (forward index). Ο indexer πραγματοποιεί άλλη μία σημαντική λειτουργία. Βρίσκει όλες τις διασυνδέσεις που περιέχονται σε κάθε σελίδα και αποθηκεύει σημαντικές πληροφορίες για αυτές σε ένα αρχείο (anchors). Αυτό το αρχείο περιέχει αρκετή πληροφορία ώστε να καθορίζεται που περιέχεται μία διασύνδεση, προς τα πού δείχνει και ποιο είναι το κείμενό της.

Ο *URLresolver* διαβάζει το αρχείο *anchors* και μετατρέπει τις σχετικές διευθύνσεις σε απόλυτες και στη συνέχεια σε *docIDs*. Μετά τοποθετεί το κείμενο *anchor* στο ευρετήριο συσχετίζοντάς το με το *docID* προς το οποίο δείχνει η διασύνδεση. Επίσης δημιουργεί μία βάση

δεδομένων από διασυνδέσεις (*links*), με τη μορφή ζευγαριών docIDs. Αυτή η βάση δεδομένων χρησιμοποιείται για τον υπολογισμό του *PageRank* για όλες τις σελίδες.

Ο *sorter* χρησιμοποιεί τα *barrels*, τα οποία είναι ταξινομημένα βάση του *docID*, και κατηγοριοποιεί βάση του *wordID*, με σκοπό να παράγει ένα άλλο ευρετήριο (*inverted index*). Ο *sorter* ακόμα δημιουργεί μία λίστα από *wordIDs* σε αυτό το ευρετήριο. Ένα πρόγραμμα, που λέγεται *DumpLexicon*, παίρνει αυτή τη λίστα μαζί με το λεξικό που έχει δημιουργηθεί από τον *indexer* και παράγει ένα νέο λεξικό το οποίο χρησιμοποιείται από τον αναζητητή (*searcher*). Ο *searcher* χρησιμοποιώντας αυτό το λεξικό, το τελικό ευρετήριο και τα *PageRanks* απαντά τα ερωτήματα του χρήστη.

Μία γραφική αναπαράσταση της αρχιτεκτονικής του συστήματος δίνεται στο Σχήμα 6.



Σχήμα 6 – Η αρχιτεκτονική του συστήματος του Google

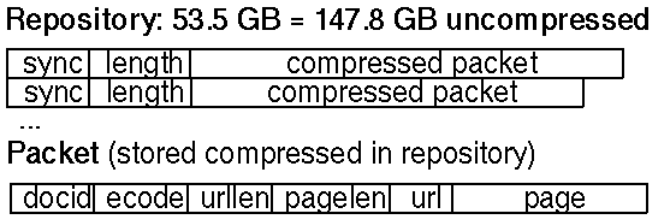
Οι δομές δεδομένων του συστήματος έχουν βελτιστοποιηθεί με τέτοιο τρόπο ώστε μία μεγάλη συλλογή σελίδων να μπορεί να crawled, κατηγοριοποιηθεί και αναζητηθεί με το λιγότερο δυνατό κόστος.

Big Files

Τα Big Files είναι εικονικά αρχεία που προκύπτουν από τη σύνδεση διάφορων συστημάτων αρχείων. Η κατανομή σε αυτά τα συστήματα αρχείων γίνεται αυτόματα. Τα αρχεία αυτά υποστηρίζουν κάποιες στοιχειώδεις επιλογές συμπίεσης.

Το *repository* περιέχει όλο τον κώδικα HTML κάθε σελίδας, σε συμπιεσμένη μορφή. Οι σελίδες είναι αποθηκευμένες η μία μετά την άλλη και προηγούνται το docID, το μέγεθός και η διεύθυνσή toyw. Για να προσπελαστεί το repository δεν απαιτούνται άλλες δομές, με

αποτέλεσμα να διευκολύνεται η συνύπαρξη των δεδομένων και να είναι ευκολότερη η ανάπτυξη. Όλες οι υπόλοιπες δομές του συστήματος μπορούν να σχηματιστούν πάλι απλά χρησιμοποιώντας το repository και ένα αρχείο που περιέχει τα λάθη που έγιναν κατά τη διαδικασία του crawling. Η μορφή του repository δίνεται στο Σχήμα 7.



Σχήμα 7 – Η δομή Repository

Document Index

Το ευρετήριο των σελίδων (document index) διατηρεί πληροφορίες για κάθε σελίδα και είναι διατεταγμένο βάση του *docID*. Η πληροφορία που είναι αποθηκευμένη σε κάθε στοιχείο περιέχει την κατάσταση της συγκεκριμένης σελίδας, ένα δείκτη στο *repository*, ένα άθροισμα ελέγχου της σελίδας (*checksum*) και διάφορες στατιστικές. Εάν το αντικείμενο έχει γίνει *crawled* τότε περιέχεται άλλος ένας δείκτης σε ένα αρχείο που λέγεται *docinfo* και περιέχει τη διεύθυνση και τον τίτλο της σελίδας, στην αντίθετη περίπτωση ο δείκτης αυτό δείχνει στην λίστα των *URLs*, η οποία περιέχει μόνο τις διευθύνσεις.

Επιπρόσθετα υπάρχει ένα αρχείο το οποίο χρησιμοποιείται για την μετατροπή των *URLs* σε *docIDs*. Αυτό το αρχείο είναι μία λίστα *checksums* των διευθύνσεων μαζί με τα αντίστοιχα *docIDs* και είναι ταξινομημένα βάση των αθροισμάτων αυτών. Για να βρεθεί το *docID* ενός συγκεκριμένου *URL*, αρκεί να υπολογιστεί το *checksum* όλων των *URLs* και μετά να εφαρμοστεί δυαδική αναζήτηση στο αρχείο των checksums για να βρεθεί το *docID*. Τα *URLs* μπορούν να μετατραπούν σε *docIDs* μαζικά απλά συγχωνεύοντας το αρχείο αυτό. Αυτή είναι και η τεχνική που χρησιμοποιεί ο URLresolver.

Lexicon

Το λεξικό έχει αρκετές διαφορετικές μορφές. Είναι υλοποιημένο σε δύο κομμάτια. Το πρώτο είναι μία λίστα από λέξεις, οι οποίες είναι συνενωμένες και χωρίζονται μόνο από κενά, και έναν πίνακα hash από δείκτες.

Hit Lists

Μία hit list αντιστοιχεί σε μία λίστα από εμφανίσεις μίας συγκεκριμένης λέξης σε μία συγκεκριμένη σελίδα, όπου συμπεριλαμβάνονται πληροφορίες για τη θέση της εμφάνισης, τη γραμματοσειρά και το εάν τα γράμματα είναι κεφαλαία. Καθώς οι λίστες αυτές χρησιμοποιούν τον περισσότερο χώρο και στα δύο ευρετήρια, είναι σημαντικό η αναπαράστασή τους να γίνεται όσο πιο αποδοτικά είναι εφικτό. Τελικά επιλέχθηκε μία βελτιστοποιημένη συμπιεσμένη κωδικοποίηση.

Η κωδικοποίηση αυτή χρησιμοποιεί δύο bytes για κάθε hit. Υπάρχουν δύο τύποι hits: τα ιδιόμορφα hits και τα απλά. Ιδιόμορφα χαρακτηρίζονται τα hits που εμφανίζονται σε ένα URL, σε έναν τίτλο, στο κείμενο μίας διασύνδεσης ή σε meta tag. Απλά χαρακτηρίζονται όλα τα υπόλοιπα. Ένα απλό hit αποτελείται από ένα bit που υποδηλώνει το εάν είναι κεφαλαίο, το μέγεθος των γραμμάτων και 12 bits που καθορίζουν τη θέση της λέξης στη σελίδα. Το μέγεθος των γραμμάτων αναπαριστάνεται σε σχέση με την υπόλοιπη σελίδα χρησιμοποιώντας τρία bits. Ένα ιδιόμορφο hit αποτελείται από το ένα bit που υποδηλώνει το εάν είναι κεφαλαίο, το

μέγεθος των γραμμάτων το οποίο όμως τίθεται ίσο με το 7 για να υποδηλώνει ότι είναι ένα ιδιόμορφο hit, 4 bits για την κωδικοποίηση του τύπου του hit και 8 bits για τη θέση. Για τα ιδιόμορφα hits που εμφανίζονται στο κείμενο των διασυνδέσεων τα 8 bits της θέσης χωρίζονται σε 4 bits που καθορίζουν τη θέση στο anchor κείμενο και στα υπόλοιπα 4 bits που προσδιορίζουν το docID στο οποίο εμφανίζεται το anchor κείμενο. Δεν αποθηκεύεται το απόλυτο μέγεθος των γραμμάτων, αλλά το σχετικό με αυτό της σελίδας, γιατί δεν πρέπει να αξιολογηθούν διαφορετικά παρόμοιες σελίδες απλά και μόνο γιατί τα γράμματα στη μία σελίδα είναι μεγαλύτερα.

Το μέγεθος της λίστας των hits αποθηκεύεται πριν τα ίδια τα hits. Για την εξοικονόμηση χώρου το μέγεθος της λίστας των hits συνδυάζεται με το wordID στο forward index και το docID στο inverted index.

Forward Index

Το forward index είναι στην πραγματικότητα μερικώς ταξινομημένο, καθώς είναι αποθηκευμένο στα 64 *barrels*. Κάθε barrel περιέχει μία συλλογή από *wordIDs*. Εάν μία σελίδα περιέχει λέξεις που βρίσκονται σε κάποιο *barrel*, τότε το *docID* της σελίδας αποθηκεύεται σε αυτό το *barrel*, ακολουθούμενο από μία λίστα από wordIDs με τις hit lists που αντιστοιχούν στις λέξεις της λίστας. Επίσης δεν αποθηκεύονται τα πραγματικά wordIDs, αλλά κάθε wordID αποθηκεύεται σαν μία σχετική διαφορά από το μικρότερο wordID που υπάρχει στο ίδιο barrel.

Inverted Index

Το *inverted index* αποτελείται από τα ίδια *barrels*, όπως και το *forward index*, με τη μόνη διαφορά ότι έχουν επεξεργαστεί από τον ταξινομητή (*sorter*). Για κάθε έγκυρο *wordID*, το λεξικό περιέχει ένα δείκτη στο *barrel* στο οποίο βρίσκεται αυτό το *wordID*. Ο δείκτης αυτός δείχνει σε μία λίστα από *docIDs* μαζί με τις αντίστοιχες hit lists. Αυτή η λίστα αναπαριστά όλες τις εμφανίσεις μίας λέξης σε όλες τις σελίδες.

Ένα σημαντικό θέμα είναι το με ποια σειρά πρέπει να εμφανίζονται σε αυτή τη λίστα τα docIDs. Μία απλή λύση είναι να αποθηκεύονται ταξινομημένα βάση του docID. Αυτή η τεχνική καθιστά δυνατή την εύκολη συγχώνευση διαφορετικών τέτοιων λιστών για ερωτήματα πολλών λέξεων. Μία άλλη λύση είναι να αποθηκεύονται ταξινομημένα βάση μίας βαθμολόγησης της εμφάνισης της λέξης στην κάθε σελίδα. Αυτό καθιστά την απάντηση ενός ερωτήματος μίας λέξης τετριμμένη και είναι δυνατό οι απαντήσεις σε ερωτήματα πολλών λέξεων να είναι κοντά στην αρχή αυτών των λιστών. Από την άλλη με αυτή τη λύση είναι πολύ δύσκολη η συγχώνευση και η ανάπτυξη επίσης είναι ιδιαίτερα δύσχρηστη, καθώς μία αλλαγή στη συνάρτηση βαθμολόγησης απαιτεί τον επαναϋπολογισμό για όλο το ευρετήριο. Επιλέχθηκε τελικά μία συμβιβαστική λύση, καθώς διατηρούνται δύο σύνολα από inverted barrels, όπου στο ένα σύνολο αποθηκεύονται οι λίστες των hits που περιέχονται σε τίτλους ή σε κείμενα διασυνδέσεων και στο άλλο σύνολο αποθηκεύονται όλες οι άλλες λίστες των hits. Με αυτή την τεχνική ελέγχεται πρώτα το αρχικό σύνολο των barrels και εάν δεν υπάρχουν αρκετά ταίρια σε αυτά τα barrels τότε ελέγχεται και το μεγαλύτερο σύνολο.

Οι Κύριες Λειτουργίες του Συστήματος

Crawling

Για να ανακτηθούν εκατοντάδες εκατομμύρια σελίδων, το Google έχει έναν γρήγορο κατανεμημένο σύστημα crawling. Ένας μόνο URLserver διαχειρίζεται τις λίστες των URLs και τις στέλνει σε έναν αριθμό από crawlers. Ο κάθε crawler διατηρεί περίπου 300 συνδέσεις ανοιχτές ανά πάσα στιγμή, το οποίο είναι αρκετό για να ανακτηθούν οι σελίδες με έναν αρκετά γρήγορο ρυθμό.

Κατηγοριοποιώντας το Διαδίκτυο

- *Parsing* – Κάθε *parser* που είναι σχεδιασμένος για το διαδίκτυο πρέπει να μπορεί να διαχειριστεί μία μεγάλη πληθώρα λαθών, τα οποία οφείλονται είτε σε τυπογραφικά λάθη στα tags του HTML είτε σε ένα μεγάλο αριθμό κενών χαρακτήρων είτε ακόμα και στην ύπαρξη μη ASCII χαρακτήρων.
- Η κατηγοριοποίηση των σελίδων σε barrels – Αφού κάθε σελίδα αναλυθεί από τον parser, κωδικοποιείται σε έναν αριθμό από barrels. Κάθε λέξη μετατρέπεται σε ένα wordID χρησιμοποιώντας έναν πίνακα hash, το λεξικό. Καινούριες προσθέσεις στον πίνακα του λεξικού σημειώνονται σε ένα αρχείο. Στη συνέχεια αφού οι λέξεις μετατραπούν σε wordIDs, οι εμφανίσεις τους στη συγκεκριμένη σελίδα μετατρέπονται σε λίστες hits, οι οποίες αποθηκεύονται στα forward barrels. Η κύρια δυσκολία του παραλληλισμού της φάσης της κατηγοριοποίησης είναι ότι απαιτείται το λεξικό να διαμοιράζεται. Έτσι αντί να γίνεται αυτό επιλέχθηκε να αποθηκεύονται σε ένα αρχείο (log) όλες οι επιπλέον λέξεις που δεν περιλαμβάνονται στο βασικό λεξικό. Με αυτό τον τρόπο πολλοί indexers μπορούν να λειτουργούν παράλληλα και στη συνέχεια το μικρό αρχείο με τις επιπλέον λέξεις να επεξεργαστεί από έναν τελικό indexer.
- Η ταξινόμηση – Για να δημιουργηθεί το inverted index ο ταξινομητής παίρνει κάθε ένα από τα forward barrels και τα ταξινομεί βάση του wordID, με σκοπό να δημιουργήσει ένα inverted barrel για τον τίτλο και τα hits του κειμένου anchor και ένα inverted barrel με όλο το κείμενο. Αυτή η διαδικασία χρησιμοποιεί ένα barrel κάθε φορά και μπορεί να εφαρμοστεί παράλληλα σε διάφορες μηχανές, απλά χρησιμοποιώντας αρκετούς τέτοιους ταξινομητές.

Αναζήτηση

Ο σκοπός της αναζήτησης είναι η επιστροφή ποιοτικών αποτελεσμάτων με έναν αποδοτικό τρόπο. Τα βήματα της διαδικασίας εκτίμησης του ερωτήματος, που πραγματοποιείται στο Google, δίνεται στη συνέχεια:

1. *Ανάλυση του ερωτήματος*
2. *Μετατροπή των λέξεων στα αντίστοιχα wordIDs*
3. *Αναζήτηση της αρχής των λιστών των σελίδων στα μικρά barrels για κάθε λέξη*
4. *Σάρωση των λιστών των σελίδων μέχρι να βρεθεί μία σελίδα που να ταιριάζει με όλους τους όρους της αναζήτησης*
5. *Υπολογισμός του βαθμού της σελίδας για το ερώτημα*
6. *Εάν σε καμία λίστα σελίδων που βρίσκεται στα μικρά barrels δεν βρεθεί σελίδα που να ταιριάζει με όλους τους όρους του ερωτήματος, τότε οδηγείται στην αρχή των λιστών των σελίδων που βρίσκονται στο μεγάλο barrel για κάθε λέξη και συνεχίζει από το βήμα 4*
7. *Για όση ώρα δεν έχει καταλήξει στο τέλος οποιασδήποτε λίστα σελίδων επιστρέφει στο βήμα 4. Ταξινόμηση των σελίδων που ταιριάζουν βάση του βαθμού τους και επιστρέφονται οι καλύτερες k.*

Το Σύστημα Βαθμολόγησης

Το Google διατηρεί πολύ περισσότερη πληροφορία για τις σελίδες του διαδικτύου από ότι οι τυπικές μηχανές αναζήτησης. Κάθε λίστα των hits περιλαμβάνει πληροφορίες για τη θέση, τη γραμματοσειρά και το εάν είναι κεφαλαία τα γράμματα. Επίσης αποθηκεύονται τα hits που εμφανίζονται σε κείμενα anchor, όπως και το PageRank της σελίδας. Συνδυάζοντας όλες αυτές τις πληροφορίες σε ένα βαθμό αξιολόγησης είναι ιδιαίτερα δύσκολο, για αυτό το λόγο

σχεδιάστηκε μία συνάρτηση βαθμολόγησης, η οποία να μην επηρεάζεται ιδιαίτερα από κανένα παράγοντα.

Η πιο απλή περίπτωση είναι αυτή που το ερώτημα αποτελείται από μόνο μία λέξη. Σε αυτή την περίπτωση το Google ψάχνει στις λίστες hits των σελίδων για αυτή τη λέξη. Θεωρείται ότι κάθε hit είναι κάποιου συγκεκριμένου τύπου, όπου ο κάθε ένας από αυτούς έχει δικό του βάρος. Αυτά τα βάρη των τύπων σχηματίζουν ένα διάνυσμα ταξινομημένο βάση του τύπου. Το Google μετράει τον αριθμό των hits κάθε τύπου στη λίστα των hits. Κάθε αριθμός εμφανίσεων, που αντιστοιχεί σε έναν τύπο hit, μετατρέπεται σε ένα βάρος μετρητή, τα οποία αυξάνονται γραμμικά μέχρι μία συγκεκριμένη τιμή. Το εσωτερικό γινόμενο του διανύσματος των βαρών μετρητή και του διανύσματος των βαρών τύπου υπολογίζει ένα IR ποσό για τη σελίδα. Τέλος αυτό το IR ποσό συνδυάζεται με το PageRank για να οδηγήσει στην τελική βαθμολόγηση της σελίδας.

Για πιο πολύπλοκη αναζήτηση, όπου το ερώτημα αποτελείται από περισσότερες από μία λέξεις, η κατάσταση είναι πιο πολύπλοκη. Πρέπει πολλές λίστες από hits να σαρωθούν έτσι ώστε στα hits που εμφανίζονται κοντά σε μία σελίδα να δίνεται μεγαλύτερο βάρος από ότι σε αυτά που εμφανίζονται σε απομακρυσμένα μεταξύ τους σημεία. Τα hits τα οποία εμφανίζονται σε κοντινές θέσεις συνενώνονται κατά κάποιο τρόπο, έτσι ώστε για κάθε τέτοιο σύνολο να υπολογίζεται μία τιμή προσεγγυσιμότητας. Η τιμή αυτή στηρίζεται στο πόσο μακριά εμφανίζονται τα hits στη σελίδα και κατηγοριοποιείται σε 10 διαφορετικούς χαρακτηρισμούς που κυμαίνονται από «φράση» μέχρι «ούτε καν σχετικά». Οι αριθμοί εμφάνισης υπολογίζονται όχι μόνο για κάθε τύπο hits, αλλά για κάθε τύπο και προσεγγυσιμότητα. Κάθε τέτοιο ζευγάρι έχει ένα αντίστοιχο βάρος τύπου και προσέγγισης. Οι αριθμοί εμφάνισης μετατρέπονται και σε αυτή την περίπτωση σε βάρη μετρητών και για να υπολογιστεί το ποσό IR απαιτείται το εσωτερικό γινόμενο του διανύσματος των βαρών των μετρητών και του διανύσματος των βαρών τύπου και προσέγγισης.

Ανάδραση

Παρατηρείται ότι η συνάρτηση βαθμολόγησης περιέχει πολλές παραμέτρους, όπως τα διάφορα βάρη που αναφέρθηκαν. Για τον υπολογισμό των σωστών τιμών αυτών των παραμέτρων, χρησιμοποιείται ένας μηχανισμός ανάδρασης. Αξιόπιστοι χρήστες μπορούν εάν επιθυμούν να εκτιμήσουν τα αποτελέσματα που τους επιστρέφονται. Έτσι ώστε όταν τροποποιείται η συνάρτηση βαθμολόγησης από τους διαχειριστές του συστήματος, λαμβάνονται υπόψη αυτές οι εκτιμήσεις των χρηστών.

Σύγκριση των Συστημάτων Clever και Google

Τα συστήματα αυτά έχουν δύο σημαντικές διαφορές. Πρώτον, το Google αναθέτει κάποιες αρχικές βαθμολογίες και τις διατηρεί ανεξάρτητα από το ερώτημα που θέτει ο χρήστης στη μηχανή, ενώ το Clever αφού δημιουργήσει ένα αρχικό σύνολο για κάθε ερώτημα στη συνέχεια δίνει προτεραιότητες στις σελίδες του συνόλου αυτού με βάση το συγκεκριμένο ερώτημα. Συνεπώς η διαδικασία του Google επιτρέπει ταχύτερη απόκριση.

Η δεύτερη διαφορά είναι ότι η φιλοσοφία του Google είναι τέτοια ώστε να εξετάζεται μόνο η προς τα εμπρός κατεύθυνση, καθώς μόνο οι διασυνδέσεις που ξεκινούν από μία σελίδα χρησιμοποιούνται. Από την άλλη το Clever εξετάζει και την προς τα πίσω κατεύθυνση, καθώς χρησιμοποιούνται και διασυνδέσεις που δείχνουν προς μία authoritative σελίδα για να βρεθούν οι σελίδες που δείχνουν σε αυτή. Συνεπώς το Clever πλεονεκτεί, καθώς εκμεταλλεύεται το κοινωνικό φαινόμενο, όπου οι άνθρωποι έμφυτα υποκινούνται να δημιουργήσουν κείμενο τύπου hub για να εκφράσουν ότι είναι ειδικοί σε κάποιο συγκεκριμένο θέμα.

5.2.2 Ο Αλγόριθμος SALSA

Ο αλγόριθμος SALSA [LM00] παρουσιάζει μία στοχαστική προσέγγιση στην ανάλυση της δομής των διασυνδέσεων και εφαρμόζει τυχαίους περιπάτους σε γράφους που προκύπτουν από σύνολα σελίδων του διαδικτύου. Η στοχαστική αυτή προσέγγιση στηρίζεται στη θεωρία των αλυσίδων Markov, και προτάθηκε ως λύση του προβλήματος που προκαλέι το φαινόμενο TCK, δηλαδή της επίδρασης των πολύ ισχυρά συνδεδεμένων κοινοτήτων (TCK – Tightly-Knit Community) στο διαδίκτυο. Μία ισχυρά συνδεδεμένη κοινότητα είναι ένα μικρό αλλά ισχυρά διασυνδεδεμένο σύνολο σελίδων. Η επίδραση του φαινομένου αυτού εμφανίζεται όταν μία τέτοια κοινότητα αξιολογείται πολύ καλά από έναν αλγόριθμο που εκμεταλλεύεται τη δομή των διασυνδέσεων, ακόμα και αν δεν περιέχει την ποιοτικότερη πληροφορία για ένα θέμα. Ο αλγόριθμος του Kleinberg επηρεάζεται ιδιαίτερα από την ύπαρξη τέτοιων κοινοτήτων. Αυτό αποδεικνύεται καλύτερα στο ακόλουθο παράδειγμα.

> Έστω μία συλλογή σελίδων η οποία περιέχει τις ακόλουθες δύο κοινότητες. Η κοινότητα *y* αποτελείται από ένα μικρό αριθμό σελίδων *hub* και *authorities*, αλλά κάθε hub της δείχνει στα περισσότερα *authority* της. Η κοινότητα *z* αποτελείται από ένα μεγάλο αριθμό σελίδων στην οποία όμως κάθε σελίδα *hub* δείχνει σε λιγότερες σελίδες *authority* από ότι οι αντίστοιχες του συνόλου *y*. Το βασικό θέμα της συλλογής αυτών των σελίδων εμφανίζεται στην κοινότητα *z* και πιθανότατα ενδιαφέρει περισσότερο τους χρήστες. Καθώς όμως υπάρχουν πολλές σελίδες authority στην κοινότητα *z* και οι σελίδες *hub* της κοινότητας αυτής δεν δείχνουν σε πολλές από αυτές, και ενώ η κοινότητα *y* είναι ισχυρά διασυνδεδεμένη τελικά οι σελίδες της κοινότητας *y* θα αξιολογηθούν καλύτερα από ότι αυτές της *z*. Έτσι τελικά δεν θα εμφανιστεί το κεντρικό θέμα της αρχικής συλλογής των σελίδων.

Ο αλγόριθμος SALSA εφαρμόζεται σε ένα βασικό σύνολο σελίδων, σχετικών με ένα θέμα, το οποίο προκύπτει όπως και στον αλγόριθμο του Kleinberg. Διαισθητικά προκύπτει ότι οι authoritative σελίδες, βάση του θέματος, θα δείχνονται από πολλές σελίδες του γράφου, που προκύπτει από το σύνολο αυτό. Επομένως είναι αναμενόμενο ότι ένας τυχαίος περίπατος σε αυτό το γράφο με μεγάλη πιθανότητα θα διατρέξει αυτές τις σελίδες.

Συνδυάζοντας τη θεωρία των τυχαίων περιπάτων και την έννοια των δύο διαφορετικών τύπων σελίδων του διαδικτύου, hub και authority, αναλύονται τελικά δύο αλυσίδες Markov, μία για κάθε τύπο σελίδων. Σε αντίθεση με τους συμβατικούς τυχαίους περιπάτους σε γράφους, οι αλλαγές της κατάστασης σε αυτές τις αλυσίδες δημιουργούνται διασχίζοντας δύο διασυνδέσεις κάθε φορά, είτε πρώτα μία προς τα εμπρός διασύνδεση και μετά μία προς τα πίσω ή το ανάστροφο. Αναλύοντας τις δύο αλυσίδες που προκύπτουν δίνεται η δυνατότητα να δοθούν σε κάθε σελίδα του γράφου δύο διαφορετικά βάρη, ένα *hub* και ένα *authority*.

Περιγραφή των τυχαίων περιπάτων

Από το βασικό σύνολο των σελίδων και τις διασυνδέσεις που εμφανίζονται σε αυτό δημιουργείται ένας διμερής μη κατευθυνόμενος γράφος $G' = (V_h, V_a, E)$. Ο υπογράφος V_h αντιστοιχεί στο σύνολο των hub σελίδων και ο V_a στο σύνολο των authority σελίδων. E είναι το σύνολο των ακμών στο γράφο G'. Κάθε μη απομονωμένη σελίδα s στο βασικό σύνολο αναπαριστάνεται με δύο κόμβους στο γράφο αυτό, τους s_h και s_a. Κάθε διασύνδεση $s \rightarrow r$ στον γράφο αυτό αναπαριστάνεται σαν μία μη κατευθυνόμενη ακμή που συνδέει τους κόμβους s_h στον r_a.

Σε αυτό το διμερή γράφο εφαρμόζονται δύο διαφορετικοί τυχαίοι περίπατοι. Κάθε περίπατος μπορεί να επισκεφτεί κόμβους μόνο από το ένα μέρος του γράφου, ακολουθώντας μονοπάτια, που αποτελούνται από δύο ακμές, σε κάθε βήμα. Καθώς όλες οι ακμές διασχίζουν τα δύο μέρη του γράφου, ο κάθε τυχαίος περίπατος περιορίζεται στο ένα μόνο μέρος του

γράφου και οι δύο περίπατοι ξεκινάνε προφανώς από διαφορετικές πλευρές του γράφου. Αξίζει να σημειωθεί ότι ένα μονοπάτι μήκους 2 στο γράφο αυτό αναπαριστάνει τη διάσχιση μία διασύνδεσης του διαδικτύου με την κανονική φορά (όταν διασχίζει την πλευρά των hubs για να καταλήξει στην πλευρά των authorities). Καθώς οι σελίδες hub και authority θα πρέπει να δείχνονται από πολλές σελίδες του γράφου προκύπτει ότι οι κόμβοι που αντιστοιχούν σε αυτές τις σελίδες θα επισκέπτονται συχνά από τους τυχαίους περιπάτους.

Επομένως ο ένας από τους δύο τυχαίους περιπάτους που εφαρμόζονται στο γράφο εναλλακτικά πρώτα πηγαίνει ομοιόμορφα σε μία από τις σελίδες που δείχνουν στη σελίδα που βρίσκεται και μετά πηγαίνει πάλι ομοιόμορφα σε μία από τις σελίδες που δείχνονται από τη σελίδα που βρίσκεται. Τα βάρη των authorities ορίζονται σαν τη στάσιμη κατανομή των δύο αλυσιδωτών βημάτων όταν πρώτα γίνεται το πρώτο βήμα και μετά το δεύτερο, ενώ τα βάρη των hubs ορίζονται σαν τη στάσιμη κατανομή των δύο αλυσιδωτών βημάτων όταν πρώτα συμβαίνει το δεύτερο και μετά το πρώτο.

Το σύνολο $B(i) = \{k : k \rightarrow i\}$ ορίζεται ως το σύνολο όλων των σελίδων που δείχνουν στη σελίδα i, δηλαδή είναι οι σελίδες που μπορούν να ανακτηθούν από τη σελίδα i, εάν ακολουθηθούν οι προς τα πίσω διασυνδέσεις. Το σύνολο $F(i) = \{k : i \rightarrow k\}$ ορίζεται ως το σύνολο όλων των σελίδων που δείχνονται από τη σελίδα i και μπορούν να ανακτηθούν ακολουθώντας τις διασυνδέσεις της σελίδας αυτής.

Η αλυσίδα *Markov* που προκύπτει για τις τιμές των *authorities* είναι η ακόλουθη:

$$P_a(i,j) = \sum_{k : k \in B(i) \cap B(j)} \frac{1}{|B(i)|} \frac{1}{|F(k)|}.$$

Αποδεικνύεται ότι η στάσιμη κατανομή $\alpha=(\alpha_1, \alpha_2,..., \alpha_N)$ αυτής της αλυσίδας Markov, ικανοποιεί την εξίσωση $\alpha_i = |B(i)|/|B|$, όπου $B=U_i B(i)$ είναι το σύνολο όλων των προς τα πίσω διασυνδέσεων. Αντίστοιχα η αλυσίδα Markov που προκύπτει για τις τιμές των hubs είναι η ακόλουθη:

$$P_h(i,j) = \sum_{k : k \in F(i) \cap F(j)} \frac{1}{|F(i)|} \frac{1}{|B(k)|},$$

Αποδεικνύεται ότι η στάσιμη κατανομή $h=(h_1, h_2,..., h_N)$ αυτής της αλυσίδας Markov, ικανοποιεί την εξίσωση $h_i = |F(i)| / |F|$, όπου $F= U_i F(i)$ είναι το σύνολο όλων των προς τα εμπρός διασυνδέσεων.

Ο αλγόριθμος αυτός δεν παρουσιάζει την ίδια τεχνική αμοιβαίας ενίσχυσης που εμφανίζεται στον αλγόριθμο του Kleinberg. Αυτό συμβαίνει γιατί η σχετική τιμή του authority μίας σελίδας σε μία συνεκτική συνιστώσα, α_i, δεν καθορίζεται από τη δομή της συνιστώσας αυτής, αλλά από τις τοπικές διασυνδέσεις γύρω από αυτή τη σελίδα. Παρατηρείται όμως ότι στην ειδική περίπτωση όπου υπάρχει μόνο μία συνεκτική συνιστώσα, ο αλγόριθμος αυτός μπορεί να θεωρηθεί σαν μία ενός βήματος περιορισμένη έκδοση του αλγορίθμου του Kleinberg. Αυτό ισχύει, καθώς στην πρώτη επανάληψη του αλγορίθμου του Kleinberg εάν πραγματοποιηθεί πρώτα η συνάρτηση I, τα βάρη των authorities θέτονται ως $\alpha = A^T u$, όπου u είναι ένα διάνυσμα που περιέχει παντού άσσους. Εάν κανονικοποιηθεί στη νόρμα L_1. τότε προκύπτει ότι $\alpha_i = |B(i)| / |B|$, η οποία είναι η στάσιμη κατανομή του αλγορίθμου *SALSA*. Αντίστοιχα ισχύει και για τα βάρη *hubs*. Στην περίπτωση που ο γράφος, που προκύπτει από το σύνολο *Base Set*, αποτελείται από περισσότερες από μία συνεκτικές συνιστώσες, τότε ο αλγόριθμος SALSA επιλέγει ένα σημείο εκκίνησης ομοιόμορφα τυχαία και εκτελεί ένα τυχαίο περίπατο στη συνεκτική συνιστώσα στην οποία περιέχεται ο αρχικό κόμβος.

Έστω j η συνεκτική συνιστώσα που περιέχει τον κόμβο i, N_i ο αριθμός των κόμβων της συνεκτικής συνιστώσας και B_i το σύνολο των προς τα πίσω διασυνδέσεων της συνιστώσας j. Τότε η τιμή του authority του κόμβου i ορίζεται ως εξής:

$$a_i = \frac{N_j}{N} \frac{|B(i)|}{|B_j|} \ .$$

Βιβλιογραφία

[Altavista] AltaVista Search Engine http://www.altavista.com

[BH98] K.Bharat and M.R.Henzinger "Improved Algorithms for Topic Distillation in a Hyperlinked Environment" *Proc. ACM Conf. Res. and Developments in Information Retrieval*, 1998

[BP98] S.Brin and L.Page "The Anatomy of a Large-Scale Hypertextual Web Search Engine" Proc. *7th International World Wide Web Conference*, 1998

[BR99] R.Baeza-Yates and B.Ribeiro-Neto "Modern Information Retrieval" *ACM Press 1999, Chapter 13*

[BRRT01] A.Borodin, G.O.Roberts, J.S.Rosenthal and P.Tsaparas, "Finding Authorities and Hubs from Link Structure on the World Wide Web, *Proc. 10th International World Wide Web Conference*, Hong Kong, May 2001

[CDGKRR98] S.Chakrabarti, B.Dom, D.Gibson, J.Kleinberg, P.Raghavan and S.Rajagopalan"Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text" *Proc. 7th International World Wide Web Conference*, 1998

[CDI98] S.Chakrabarti, B.Dom and P.Indyk "Enhanced Hypertext Categorization Using Hyperlinks" Proc. *ACM SIGMOD International Conference on Management of Data*, 1998

[Chakrabarti et al. 99] S.Chakrabarti, B.Dom, D.Gibson, J.Kleinberg, S.R.Kumar, P.Raghavan, S.Rajagopalan and A.Tomkins "Hypersearching the Web" *Scientific American*, June 1999

[Chakrabarti et al. 99b] S.Chakrabarti, B.Dom, D.Gibson, J.Kleinberg, S.R.Kumar, P.Raghavan, S.Rajagopalan and A.Tomkins "Mining the Link Structure of the World Wide Web" *IEEE Computer*, August 1999

[F97] G.N.Frederickson "A Data Structure for Dynamically Maintaining Rooted Tree" *Journal of Algorithms* 24, 1997

[GKR98a] D.Gibson, J.Kleinberg and P.Raghavan "Inferring Web Communities from Link Topology" *Proc. 9th ACM Conference on Hypertext and Hypermedia*, 1998

[GKR98b] D.Gibson, J.Kleinberg and P.Raghavan "Structural Analysis of the World Wide Web" Position paper at the *WWW Consortium Web Characterization Workshop*, November 1998

[Google] Google Search Engine http://www.google.com

[GG01] G.Greco and S.Greco "Topic Distillation on Hyperlinked Data", *Unpublished manuscript*

[H01] M.R.Henzinger "Web Information Retrieval – an Algorithmic Perspective", In *Proceedings of European Symposium on Algorithms* (ESA)

[K97] J.Kleinberg "Autoritative Sources in a Hyperlinked Environment" Proc. *9th ACM-SIAM Symposium on Discrete Algorithms*, 1998 Extended version in *Journal of the ACM* 46(1999) Also appears as IBM Research Report RJ 10076, May 1997

[K99] J.Kleinberg "The Small-World Phenomenon: an Algorithmic Perspective" *Cornell Computer Science* Technical Report 99-1778, October 1999

[Kleinberg et al. 99] J.Kleinberg, S.R.Kumar, P.Raghavan, S.Rajagopalan and A.Tomkins "The Web as a Graph: Measurements, Models and Methods" Invited survey at the International Conference on Combinatorics and Computing, 1999

[KT99] J.Kleinberg and A.Tomkins "Applications of Linear Algebra to Information Retrieval and Hypertext Analysis" Tutorial survey at the *ACM Symposium on Principles of Database Systems, 1999*

[LM00] R.Lempel and S.Moran "The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect" Proc. 9th International World Wide Web Conference, Amsterdam, May 2000

[LS01] R.Lempel and A.Saffer "PicASHOW: Pictorial Authority Search by Hyperlinks on the Web" Proc. 10th International World Wide Web Conference, Hong Kong, May 2001

[PRTV98] C.H.Papadimitriou, P.Raghavan, H.Tamaki and S.Vempala "Latent Semantic Indexing: a Probabilistic Analysis" Proc ACM Symposium on Principles of Database systems, 1998

[Search Engine Watch] Search Engine Watch http://searchenginewatch.com/reports/sizes.html

[Telcordia] Telcordia Technologies – NetSizer http://www.netsizer.com/