ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Προχωρημένων Μηχανισμών Εντοπισμού Ερευνητικών Τάσεων σε Βιοϊατρική Βιβλιογραφία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Α. Λάμπρης

Επιβλέπων : Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιούλιος 2013

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Προχωρημένων Μηχανισμών Εντοπισμού Ερευνητικών Τάσεων σε Βιοϊατρική Βιβλιογραφία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Α. Λάμπρης

Επιβλέπων : Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Ιουλίου 2013.

.....	.....	.....
Θεοδώρα Βαρβαρίγου Καθηγήτρια Ε.Μ.Π.	Δημήτριος Κουτσούρης Καθηγητής Ε.Μ.Π.	Βασίλειος Λούμος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2013

.....
Λάμπρης Α. Κωνσταντίνος

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

Περιεχόμενα

Περιεχόμενα Εικόνων

Περίληψη

Σκοπός της διπλωματικής αυτής είναι η μελέτη και ανάπτυξη τεχνικών εντοπισμού τάσεων στο χώρο της ιατρικής μέσω της ανάλυσης άρθρων και δημοσιεύσεων.

Ένας μεγάλος αριθμός επιστημονικών πληροφοριών δημοσιεύεται καθημερινά σε διάφορα μέσα, με τη δυσκολία εύρεσης, πρόσβασης, κατανόησης, αξιολόγησης αλλά και σύνδεσης αυτών να είναι σημαντική. Ειδικότερα στο χώρο της κλινικής και μη κλινικής έρευνας, η συγκέντρωση και ανάλυση υφιστάμενων δεδομένων που δημοσιοποιούνται ανοικτά και μη μέσα από τη βιβλιογραφία, τις κλινικές δοκιμές και τα ερευνητικά πειράματα, μεταξύ άλλων, μπορεί να αποκαλύψει νέα ευρήματα που θα μπορούσαν να έχουν μεγάλη αξία.

Αν και σημαντικές ερευνητικές προσπάθειες γίνονται για τον εντοπισμό και την ανάλυση τάσεων στην έρευνα, τα αποτελέσματά τους έχουν ακόμη μόνο περιορισμένη χρήση σε σχετικές δραστηριότητες της επιστημονικής και βιομηχανικής έρευνας. Επομένως, είναι αναγκαία η ανάπτυξη μηχανισμών για την αυτόματη ανάλυση επιστημονικών άρθρων οι οποίοι θα μπορούν να τα κατηγοριοποιήσουν ποικιλοτρόπως αλλά και να εντοπίσουν νέες και ανερχόμενες τάσεις στην έρευνα.

Στα πλαίσια αυτά, μελετήθηκε το πεδίο της εύρεσης τάσεων με μια ανασκόπηση στην τρέχουσα τεχνολογική κατάσταση και τις ήδη υπάρχουσες εφαρμογές αλλά και το πεδίο της εξόρυξης δεδομένων, δεδομένης της ανάγκης εξαγωγής των δεδομένων ενδιαφέροντος από ελεύθερα κείμενα δημοσιεύσεων. Ακολούθως, αναπτύχθηκε η προτεινόμενη μεθοδολογία για τον εντοπισμό των τάσεων και εστιάσαμε στην ανάπτυξη της τεχνικής των άμεσων βιβλιογραφικών αναφορών. Χρησιμοποιώντας σύγχρονες τεχνολογίες, δημιουργήσαμε μια υλοποίηση της εν λόγω μεθοδολογίας.

Abstract

The purpose of this thesis is to study and develop techniques to identify trends in the field of medicine through analysis of articles and publications.

There is an excessive amount of dispersed scientific information published daily, at a variety of data sources, which is difficult to find, access, comprehend, evaluate and link. Notably in the area of clinical and non-clinical research, the collection and analysis of existing data, openly published or not, through the literature, clinical trials and research experiments, among others, can reveal new findings that could be of great value.

Although significant research efforts are made to identify and analyze trends in research, their results still have only limited use in related activities of scientific and industrial research. It is therefore necessary to develop mechanisms for the automatic analysis of scientific articles which they will categorize this articles accordingly and to identify new and emerging trends in research.

In this context, we studied the scope of identifying trends by looking at the current technological situation and existing applications but also the scope of the field of data mining, given the need to export data of interest from the texts of free publications. Subsequently we developed the proposed methodology to identify trends and focused on the development of the technique of direct citations. Using modern technology, we have created an implementation of this methodology.

Ευχαριστίες

Ολοκληρώνοντας τη διπλωματική μου εργασία, θα ήθελα να ευχαριστήσω την Καθηγήτρια Ε.Μ.Π. κ. Θεοδώρα Βαρβαρίγου για τη δυνατότητα που μου έδωσε να δουλέψω υπό την επίβλεψη και την πολύτιμη καθοδήγησή της. Από τη συνεργασία αυτή αποκόμισα ιδιαίτερη γνώση και εμπειρία.

Επίσης, ιδιαίτερα θα ήθελα να ευχαριστήσω τη Δρ. Βασιλική Ανδρόνικου για τη συνεχή καθοδήγηση και ενθάρρυνση που μου έδωσε κατά την εκπόνηση της εργασίας αυτής και για τις πολύτιμες συμβουλές της τόσο επί της ουσίας της έρευνάς μου όσο και σε γενικότερα ακαδημαϊκά θέματα καθώς και για το χρόνο που μου αφιέρωσε.

Κεφάλαιο 1: Εισαγωγή

1.1 Το Γενικό Πλαίσιο

Καθημερινά δημοσιεύεται σε πολλά και διάφορα μέσα μια πλειάδα από επιστημονικά άρθρα και δεδομένα τα οποία αφορούν διάφορους τομείς γνώσης. Ο τεράστιος όγκος αυτής της πληροφορίας δυσχεραίνει την ανάλυση, διασύνδεση, κατανόηση και εξαγωγή γνώσης από αυτήν.

Φυσικά είναι αδύνατον όλα τα επιστημονικά άρθρα που δημοσιεύονται να διαβαστούν και ταξινομηθούν πλήρως μη αυτόματα από άλλους επιστήμονες λόγω του ότι θα ήταν απίστευτα χρονοβόρο αλλά και με μεγάλο κόστος. Γι' αυτό δημιουργήθηκε η ανάγκη έτσι ώστε να υπάρχει ένας τρόπος τα νέα άρθρα να αναλύονται με αυτόματο ή έστω ημι-αυτόματο τρόπο από υπολογιστές οι οποίοι θα μπορούν να τα κατατάξουν σε κατηγορίες αλλά ταυτόχρονα να εντοπίσουν νέες και ανερχόμενες τάσεις στην έρευνα.

Η δυνατότητα να γνωρίζουμε στην αρχή της έρευνας ότι κάτι αποτελεί ανερχόμενη τάση είναι πολύ σημαντικό και χρήσιμο. Μέχρι σήμερα η ανίχνευση τάσεων είχε μικρή εφαρμοσιμότητα στην ερευνητική και βιομηχανική έρευνα κυρίως λόγω των πτωχών αποτελεσμάτων των εφαρμοζόμενων μοντέλων και μηχανισμών ή/και περιορισμού τους σε εξαιρετικά συγκεκριμένα πεδία. Παράλληλα όμως παρατηρείται διαρκής ανάπτυξη της τεχνολογίας και συνεχιζόμενη αύξηση του όγκου των διαθέσιμων πληροφοριών. Έτσι, καθίσταται ιδιαιτέρως δύσκολος ο εντοπισμός των τάσεων στην έρευνα τόσο για ενδιαφερόμενες εταιρείες όσο και για ερευνητές, με αποτέλεσμα να μην έχουν τη δυνατότητα να προγραμματίσουν αποτελεσματικά τη διαθεσιμότητα των κεφαλαίων τους και να αξιολογήσουν με άφθονες πληροφορίες τη μελλοντική στρατηγική τους.

Επομένως, είναι αναγκαίο να δημιουργηθούν κάποιοι μηχανισμοί που να εξάγουν και να αναλύουν όλες τις διαθέσιμες πληροφορίες, είτε αυτές είναι ανοικτές στο κοινό είτε όχι, από τη βιβλιογραφία, τα ερευνητικά πειράματα και άλλες πηγές και να μπορούν να εντοπίσουν τάσεις σε διάφορους τομείς.

1.2 Στόχος Διπλωματικής

Ο στόχος της παρούσας διπλωματικής είναι να γίνει μια ανάλυση και παρουσίαση των προηγούμενων και υφιστάμενων τεχνολογιών στον τομέα του εντοπισμού τάσεων και να γίνει μια καλύτερη κατανόηση των περιορισμών αλλά και των δυνατοτήτων που μας προσφέρει η σημερινή τεχνολογία στη δυνατότητά μας να εντοπίζουμε εύκολα και έγκαιρα τις διάφορες τάσεις. Επίσης θα παρουσιαστεί η υλοποίηση μιας εφαρμογής για τον εντοπισμό των τάσεων αυτών.

1.3 Διάρθρωση Κειμένου

Το κείμενο της διπλωματικής εργασίας διαρθρώνεται ως εξής:

Στο κεφάλαιο 2 γίνεται μια σύντομη εισαγωγή στον εντοπισμό τάσεων όπου παρουσιάζεται τι είναι ο εντοπισμός τάσεων αλλά και η τρέχουσα τεχνολογική κατάσταση με τη θεωρητική παρουσίαση διαφόρων μεθοδολογιών για την έυρεση διαφόρων τάσεων. Ακολούθως παρουσιάζονται διάφορες υλοποιημένες εφαρμογές, που κάποιες από αυτές βρίσκονται στο εμπόριο και αφορούν τον εντοπισμό τάσεων.

Στο κεφάλαιο 3 γίνεται μια ανασκόπηση στην εξόρυξη δεδομένων. Η εξόρυξη δεδομένων είναι ένα αναγκαίο και άρρηκτο κομμάτι του εντοπισμού τάσεων αφού για την ανάλυση των διαθέσιμων πληροφοριών απαιτούνται μηχανισμοί που να διαβάζουν τα υπάρχων δεδομένα και να εξάγουν τις πιο σημαντικές πληροφορίες. Και πάλι γίνεται μια γενική εισαγωγή στο τι είναι εξόρυξη δεδομένων αλλά και στα προβλήματα που οδήγησαν στη δημιουργία του πεδίου αυτού και ακολούθως παρουσιάζονται η τρέχουσα τεχνολογική κατάσταση με διάφορες μεθοδολογίες αλλά και το πού βρίσκουν χρησιμότητα οι εφαρμογές εξόρυξης δεδομένων.

Στο κεφάλαιο 4 παρουσιάζεται η αναπτυχθείσα εφαρμογή για τον εντοπισμό τάσεων. Αρχικά παρουσιάζονται οι εργασίες που έχουν ήδη γίνει από προηγούμενους ερευνητές και ακολούθως δίνεται μια θεωρητική περιγραφή του μοντέλου που έχει ακολουθηθεί. Στη σύνεχεια γίνεται μια πιο αναλυτική περιγραφή του μοντέλου αλλά και παρουσίαση των μηχανισμών που έχουν χρησιμοποιηθεί για την υλοποίησης. Τέλος παρατίθενται κάποιες λεπτομέρειες της υλοποίησης.

Στο κεφάλαιο 5 γίνεται μια αποτίμηση των μηχνανισμών που έχουν χρησιμοποιηθεί. Αρχικά δίνονται παραδείγματα εφαρμογής των μηχανισμών μέσω στιγμιότυπων από την έξοδο του προγράμματος και αναφέρονται σχόλια για τη ροή εκτέλεσης του προγράμματος.

Στο κεφάλαιο 6 παρουσιάζονται τα αποτελέσματα αλλά και τα συμπεράσματά μας σε μορφή πλεονεκτημάτων και μειονεκτημάτων της μεθοδολογίας που ακολουθήθηκε. Στη συνέχεια γίνεται μια σύνοψη της διπλωματικής εργασίας όπου περιγράφεται ο στόχος της διπλωματικής, η εργασία μας και το τι έχουμε πετύχει κατά την εκπόνησή της. Ακολούθως παρατίθενται ιδέες και μεθοδολογίες για μελλοντική έρευνα για τη βελτίωση των μηχανισμών που έχουν αναπτυχθεί.

Κεφάλαιο 2: Εντοπισμός Τάσεων

Ο εντοπισμός τάσεων είναι ένας σχετικά νέος τομέας του οποίου στόχος είναι από ένα σύνολο από διάφορες πηγές (άρθρα, κείμενα), με αυτόματους ή ημι-αυτόματους τρόπους, να εξαχθούν διάφορες τάσεις σε τομείς που ενδιαφέρουν το χρήστη.

Στο χώρο της έρευνας, καθημερινά εκδίδεται ένας μεγάλος αριθμός από επιστημονικά άρθρα σε διάφορα μέσα (επιστημονικά περιοδικά, εφημερίδες, ιστολόγια), που αφορούν σε νέες έρευνες και καινοτομίες. Η ανάγκη για εντοπισμό των τάσεων στην έρευνα είναι μεγάλη μιας και οι μεγάλες εταιρείες που ασχολούνται με την έρευνα πρέπει να είναι σε θέση να εντοπίζουν τις κατευθύνσεις της τεχνολογίας και της επιστήμης, έτσι ώστε να διαθέσουν τους ανάλογους πόρους και να καθορίσουν τη στρατηγική τους στην έρευνα. Επομένως, υπάρχει ανάγκη για την ανάπτυξη μηχανισμών οι οποίοι να μπορούν να αναλύουν αυτά τα άρθρα και να μπορούν να εντοπίζουν διάφορες τάσεις στην έρευνα.

2.1 Τι είναι ο εντοπισμός τάσεων

Ο εντοπισμός τάσεων (emerging trend detection - ETD) αφορά στη συλλογή και ανάλυση πληροφοριών με σκοπό τον εντοπισμό ενός μοτίβου, ή μιας τάσης, στις πληροφορίες και στη μετέπειτα επεξεργασία τους. Μια ανερχόμενη τάση είναι ένα θεματικό πεδίο μιας επιστήμης το οποίο συνεχώς αυξάνεται σε επιστημονικό ενδιαφέρον αλλά και χρησιμότητα όσο περνά ο χρόνος.

Ο εντοπισμός τάσεων της έρευνας – έστω και σε ένα μικρό και συγκεκριμένο ερευνητικό πεδίο - μέσω της μελέτης και συσχέτισης της επιστημονικής βιβλιογραφίας αποτελεί ένα εξαιρετικά χρονοβόρο εγχείρημα του οποίου η συνδυαστική πολυπλοκότητα πολύ συχνά αποτρεπτική και καθιστά ανέφικτη την προσπάθεια. Γι’ αυτό και δημιουργήθηκε η ανάγκη για δημιουργία αυτόματων ή ημι-αυτόματων μηχανισμών που θα μπορούν σε λίγο χρόνο να αναλύσουν τα άρθρα αυτά και να προτείνουν διάφορες τάσεις στους τομείς που ενδιαφέρουν τους ερευνητές.

Η συνήθης διαδικασία η οποία ακολουθείται κατά τον εντοπισμό τάσεων περιλαμβάνει τρεις φάσεις: την αναπαράσταση του θέματος, την ταυτοποίηση και τον έλεγχο (Minh-Hoang Le, 2005). Κατά την αναπαράσταση του θέματος, το κάθε θέμα που μας ενδιαφέρει αναπαρίσταται από κάποια χαρακτηριστικά όπως λέξεις-κλειδιά και ακολούθως κατά την ταυτοποίηση αυτά τα χαρακτηριστικά εξάγονται από τα άρθρα με διάφορες τεχνικές εξόρυξης δεδομένων. Τέλος, κατά τη φάση του ελέγχου, τα χαρακτηριστικά παρατηρούνται σε βάθος χρόνου και αξιολογούνται με βάση το *ενδιαφέρον* που παρουσιάζουν και τη *χρησιμότητά* τους. Αν το ενδιαφέρον και η χρησιμότητά τους αυξάνονται με το χρόνο, τότε είναι πιθανόν το θέμα που αναλύθηκε να αποτελεί ανερχόμενη τάση.

Επομένως, οι εφαρμογές για εντοπισμό τάσεων λαμβάνουν ως είσοδο μια μεγάλη συλλογή από δεδομένα κειμένου και βρίσκουν θέματα τα οποία δεν έχουν ξανά αναδειχθεί ή ερευνηθεί και των οποίων η σημασία αυξάνεται ανάλογα με την αύξηση του πλήθος των άρθρων όσο περνά ο χρόνος. Δηλαδή αν οι δημοσιεύσεις για ένα θέμα αυξάνονται όσο περνά ο χρόνος αυτό σημαίνει πως αυξάνεται και η σημαντικότητα του θέματος αυτού. Οι εφαρμογές αυτές χωρίζονται σε δύο κατηγορίες: τα πλήρως αυτόματα συστήματα και τα ημι-αυτόματα συστήματα (Kontostathis, 2002). Τα πλήρως αυτόματα συστήματα παίρνουν μια συλλογή από δεδομένα και βγάζουν στην έξοδό τους μια λίστα με τις πιθανές τάσεις. Ακολούθως ένας ειδικός αξιολογεί τα αποτελέσματα και αποφαίνεται αν όντως τα αποτελέσματα αυτά αποτελούν τάσεις. Συνήθως τα συστήματα αυτά παρουσιάζουν τα αποτελέσματά τους σε γραφικό περιβάλλον για να βοηθήσουν τους χρήστες τους στην κατανόηση των αποτελεσμάτων. Τα ημι-αυτόματα συστήματα ως είσοδο - εκτός από τη συλλογή των δεδομένων - περιμένουν από το χρήστη να δώσει την κατευθυντήρια γραμμή για την έρευνα. Έτσι, τα συστήματα αυτά θα παρουσιάσουν αποδείξεις και θα αποφανθούν αν το θέμα που έδωσε ο χρήστης είναι όντως μια ανερχόμενη τάση.

Παρακάτω παρουσιάζεται η τρέχουσα τεχνολογική κατάσταση στον τομέα του εντοπισμών τάσεων και διάφορες τεχνικές που χρησιμοποιούνται για την εύρεση αυτών των τάσεων.

2.2 Τρέχουσα Τεχνολογική Κατάσταση

Ο εντοπισμός τάσεων είναι ένα ανερχόμενο πεδίο έρευνας. Ακόμα δεν υπάρχουν πλήρως ικανοποιητικές λύσεις για το πρόβλημα του άμεσου εντοπισμού των τάσεων καθώς αυτές αναπτύσσονται. Τα διάφορα συστήματα και τεχνικές που υπάρχουν δίνουν λύσεις σε συγκεκριμένα προβλήματα αφήνοντας όμως ανεπίλυτα άλλα. Παρακάτω θα παρουσιαστούν κάποιες βασικές τεχνικές που χρησιμοποιούνται για τον εντοπισμό τάσεων.

2.2.1 Μετρικές χρησιμότητας και ενδιαφέροντος για εντοπισμό τάσεων

Στην τεχνική αυτή χρησιμοποιούνται δύο κυρίως μετρικές για την αξιολόγηση των διάφορων άρθρων, η *χρησιμότητα* (*utility*) και το *ενδιαφέρον* (*interest*). Για την εύρεση ανερχόμενων τάσεων το κάθε θέμα (topic) συνδέεται με διάφορα χαρακτηριστικά που προέρχονται από τα επιστημονικά άρθρα και γίνεται ο υπολογισμός δύο μετρικών για κάθε θέμα για την κατάταξή του ανάλογα με το ενδιαφέρον και τη χρησιμότητά του. Η μέθοδος αυτή μπορεί να προσαρμοστεί εύκολα σε διάφορα είδη επιστημονικών άρθρων και μπορεί να τροποποιηθεί εύκολα για να προσαρμοστεί στις ανάγκες του κάθε χρήστη (Minh-Hoang Le, 2005).

Το μοντέλο που ακολουθείται είναι το εξής:

M = {T,D,f,g,CE}

όπου: $D=\{d_j\}$: Ένα σύνολο από επιστημονικά άρθρα
$T=\{t_i\}$: Ένα σύνολο από θέματα (topics)
f(.): Το μέτρο της αύξησης του ενδιαφέροντος
g(.): Το μέτρο της αύξησης της χρησιμότητας
C: Ο αξιολογητής
E: Το σύνολο των ανερχόμενων τάσεων

Το μοντέλο έχει ένα σύνολο T που αποτελείται από τα θέματα που πρέπει να αξιολογηθούν. Όλα τα θέματα στο T είναι οργανωμένα σε ένα ιεραρχικό λεξικό όπου κάθε κόμβος είναι ένα θέμα και οι σχέσεις μεταξύ των θεμάτων θεωρούνται σχέσεις παιδιού-γονέα στο ιεραρχικό λεξικό. Το μοντέλο αυτό λαμβάνει μια σειρά επιστημονικών άρθρων D ως είσοδο και συσχετίζει το κάθε θέμα με κάποια χαρακτηριστικά που προέρχονται από το D. Μετά από αυτό, ο αξιολογητής C αναλύει τα χαρακτηριστικά στο D σε διάφορες χρονικές περιόδους για την αξιολόγηση του μέτρου αύξησης του ενδιαφέροντος και της χρησιμότητας με δύο συναρτήσεις f και g και επαληθεύει κατά πόσον ή όχι το κάθε θέμα είναι μια αναδυόμενη τάση. Η έξοδος είναι το σύνολο των αναδυόμενων τάσεων που επιλέγονται από το μοντέλο.

Στο μοντέλο αυτό, το κάθε θέμα της ιεραρχίας T, είναι ένα σύνολο από συνώνυμες λέξεις. Η ιεραρχική δομή βασίζεται στις σχέσεις μεταξύ των θεμάτων.

Η τεχνική για να υπολογιστεί η σχέση ενός θέματος $t_i \in T$ με ένα άρθρο $d_j \in D$ είναι η εξής: Πρώτον, χρησιμοποιούμε το tf * idf μέτρο για να εξαγάγουμε λέξεις-κλειδιά από το d_j. Κάθε λέξη-κλειδί στη συνέχεια αντιστοιχίζεται με τα θέματα της ιεραρχίας T. Στη συνέχεια, σαρώνεται ολόκληρο το έγγραφο, για να μετρηθεί πόσες φορές ένα θέμα αναφέρεται στο d_j. Κάθε φορά που υπολογίζεται ένα θέμα, συνυπολογίζονται τα θέματα γονείς στην ιεραρχία T. Η σχέση του άρθρου d_j στο θέμα t_i υπολογίζεται ως εξής:

$$r(i,j) = \frac{Count(t_i)}{\sum_{t_j \in T} Count(t_j)}$$

όπου Count(t_i) είναι ο αριθμός των φορών που το θέμα t_i υπολογίζεται.

Για να προσδιορίσουμε πόσο συχνά το θέμα t_i αναφέρεται στο k-ιοστό έτος, προσθέτουμε όλες τις σχέσεις μεταξύ των άρθρων που δημοσιεύονται στο έτος k στο t_i:

$$t_i^k(1) = \sum_{year(d_j)=k} r(i,j)$$

2.2.2 Αυτόματη πρόβλεψη των όρων της οντολογίας MeSH

Στο σύνολο αυτών των τεχνικών γίνεται αυτόματη πρόβλεψη των επιστημονικών όρων της οντολογίας MeSH για τις περιλήψεις (abstracts) νέων άρθρων (Fabian Mörchen, 2008). Η MeSH είναι μια οντολογία η οποία ως σκοπό έχει την κατάταξη επιστημονικών άρθρων και βιβλίων, με τη χρήση κάποιων παραμέτρων/επιστημονικών όρων οι οποίοι σχετίζονται με το κάθε άρθρο. Στόχος των παρουσιαζόμενων τεχνικών είναι η πρόβλεψη όρων οι οποίοι δεν περιέχονται στη MeSH και θα μπορούσαν να την εμπλουτίσουν. *Οι όροι αυτοί πολλές φορές αντιπροσωπεύουν και τις διάφορες τάσεις στην έρευνα*. Διάφορες τεχνικές χρησιμοποιούνται για την πρόβλεψη των όρων της οντολογίας MeSH (Dmitriy Fradkin, 2008):

Αναγνώριση όρων-οντοτήτων: Οι όροι-οντότητες, όπως τα γονίδια, οι ασθένειες ή οι δραστικές ουσίες ανιχνεύονται με έναν συνδυασμό αναγνώρισης με βάση ένα λεξικό και τεχνικές επεξεργασίας φυσικής γλώσσας (natural language processing).

Ταξινόμηση: Όροι της οντολογίας MeSH εναποτίθενται στα νέα άρθρα αυτόματα χρησιμοποιώντας ένα μοντέλο πιθανοτήτων.

Αναλύση τάσεων: Η συχνότητα όλων των όρων παρακολουθείται διαχρονικά. Χαρακτηριστικά εξάγονται από τις τάσεις ώστε να εντοπίζονται αυτόματα οι αναδυόμενες τάσεις.

Ομαδοποίηση: Τεχνικές ομαδοποίησης χρησιμοποιούνται για να ανακαλυφθούν τάσεις στο σύνολο των άρθρων και σε ομάδες άρθρων.

Κατάταξη: Τα έγγραφα, οι συστάδες εγγράφων, καθώς καi οι τάσεις κατατάσσονται ανάλογα με τα χαρακτηριστικά που εξάγονται και τα βάρη που έχουν ανατεθεί σε αυτά τα χαρακτηριστικά.

2.2.3 Δίκτυα Αναφορών

Η βασική ιδέα είναι ότι τα έγγραφα τα οποία ασχολούνται με ένα παρόμοιο θέμα συχνά αναφέρουν το ένα το άλλο κι έτσι συνδέονται στενά μεταξύ τους, ενώ για τα έγγραφα που ασχολούνται με διαφορετικά θέματα είναι πολύ μικρή η πιθανότητα να αναφέρουν το ένα το άλλο και είναι ασθενώς συνδεδεμένα (Naoki Shibata, Comparative Study on Methods of Detecting Research Fronts Using Different Types of Citation, 2008). Ως εκ τούτου, μπορούμε να διαχωρίσουμε τα διάφορα άρθρα σε διάφορες συνδεδεμένες ομάδες με την ανάλυση των παραπομπών τους και έτσι να εντοπίσουμε ποια είναι τα είδη των θεμάτων που συζητήθηκαν σε ένα συγκεκριμένο τομέα της έρευνας. Συνήθως για την ομαδοποίηση χρησιμοποιείται η τοπολογική μέθοδος ομαδοποίησης (Ichiro Sakata, 2012).

Υπάρχουν τρεις τύποι δικτύου βιβλιογραφικών αναφορών:

- Συν-αναφορά (co-citation): ορίζεται ως η ακμή μεταξύ δύο άρθρων-κόμβων που αναφέρονται από τα ίδια άρθρα.
- Βιβλιογραφική σύζευξη: ορίζεται ως η ακμή μεταξύ δύο άρθρων για τα οποία υπάρχει αναφορά στα ίδια άρθρα.
- Άμεση αναφορά: ορίζεται ως η ακμή μεταξύ δύο άρθρων που το ένα αναφέρει το άλλο.

Παράδειγμα: Αν τα άρθρα Α και Β έχουν αναφερθεί από ένα άρθρο Γ, τότε μεταξύ του Α και Β υπάρχει συν-αναφορά. Αν το Δ και το Ε αναφέρονται στο Ζ, τότε υπάρχει βιβλιογραφική σύζευξη μεταξύ του Δ και του Ε.

Οι τρεις μεθόδοι αυτοί δοκιμάστηκαν για την ανίχνευση των αναδυόμενων τάσεων σε τρεις διαφορετικούς τομείς έρευνας. Τρεις τύποι δικτύου βιβλιογραφικών αναφορών κατασκευάστηκαν για κάθε τομέα της έρευνας και οι εργασίες σε αυτούς τους τομείς χωρίστηκαν σε ομάδες, για να ανιχνευθούν τα μέτωπα της έρευνας.

Η απόδοση κάθε τύπου δικτύου αναφορών στην ανίχνευση νέων τάσεων γίνεται, χρησιμοποιώντας τις ακόλουθες μετρικές:

Ορατότητα: μετριέται με το κανονικοποιημένο μέγεθος της συστάδας

Ταχύτητα: μετράται από το μέσο χρόνο έκδοσης των άρθρων της συστάδας

Τοπολογική σημασία: μετριέται από την πυκνότητα της συστάδας

Τα αποτελέσματα της έρευνας δείχνουν πως η άμεση αναφορά έχει την καλύτερη επίδοση, δεδομένου ότι μπορεί να ανιχνεύσει νέες και αναδυόμενες συστάδες νωρίτερα. Η συν-αναφορά παρουσιάζει τα χειρότερα αποτελέσματα. Επιπλέον, στα δίκτυα άμεσης αναφοράς, ο συντελεστής ομαδοποίησης ήταν μεγαλύτερος, γεγονός που υποδηλώνει ότι η ομοιότητα του περιεχομένου των εγγράφων που συνδέονται με τις άμεσες αναφορές είναι μεγαλύτερη και ότι τα δίκτυα άμεσης αναφοράς έχουν τον ελάχιστο κίνδυνο να μην εμφανίσουν ένα νέο ανερχόμενο τομέα.

2.2.4 Φράσεις που απαντώνται συχνά

Ακολούθως περιγράφεται η μέθοδος ανακάλυψης νέων τάσεων με φράσεις που συν-εμφανίζονται πιο συχνά (Saurabh Goorha, 2010).

Ως είσοδο ο χρήστης καθορίζει ένα σταθερό σύνολο από όρους τους οποίους ενδιαφέρεται να παρακολούθησει, σε ένα σύνολο από άρθρα και έγγραφα το οποίο ενημερώνεται συνεχώς.

Στη συνέχεια ακολουθούνται τα εξής βήματα:

1. Προσδιορισμός των φράσεων, των ομάδων των λέξεων που εμφανίζονται μαζί πιο συχνά από ότι θα αναμενόταν αν ήταν τυχαία η συνεμφάνισή τους.

2. Εύρεση φράσεων που βρίσκονται κοντά στους όρους που μας ενδιαφέρουν.

3. Καθορισμός των φράσεων που συναντώνται πιο συχνά (αυξάνεται η χρήση τους)

4. Προσδιορισμός των φράσεων που έχουν περισσότερο ενδιαφέρον με βάση το
    - πόσο συχνά αναφέρονται
    - πόσο πιο συχνά αναφέρονται από πριν
    - πόσο συγκεκριμένες είναι στο θέμα που μας ενδιαφέρει

5. Παρουσίαση των αποτελεσμάτων στο χρήστη

Μια φράση θεωρείται ότι εμφανίζεται πιο συχνά σε ένα δεδομένο σύνολο από άρθρα αν η φράση έχει:

1) Αναφερθεί περισσότερο από ένα συγκεκριμένο ελάχιστο αριθμό φορών ανά ημέρα.

2) Αναφερθεί πρόσφατα περισσότερο από ένα συγκεκριμένο αριθμό φορών.

3) Η εμφάνισή της αυξήθηκε κατά περισσότερο από ένα συγκεκριμένο ποσοστό σε σχέση με το τελευταίο ποσοστό εμφάνισής της.

2.3 Υλοποιημένες Εφαρμογές

Παρακάτω παρουσιάζονται διάφορες εφαρμογές για εντοπισμό ανερχόμενων τάσεων που έχουν υλοποιηθεί και τα διάφορα πλεονεκτήματα και μειονεκτήματά τους.

2.3.1 Technology Opportunities Analysis TOA (Ανάλυση Ευκαιριών Τεχνολογίας)

Το TOA είναι ένα ημι-αυτόματο σύστημα ανίχνευσης νεών τάσεων που αναπτύχθηκε για την ανάλυση των τεχνολογικών ευκαιριών (Kontostathis, 2002). Το σύστημα εξάγει έγγραφα τα οποία είναι σχετικά με κάποιους όρους που δίνει ο χρήστης και παρέχει ανάλυση των δεδομένων χρησιμοποιώντας πληροφορίες όπως πλήθος λέξεων, πληροφορίες για τα δεδομένα και τις παραπομπές αλλα και πληροφορίες για τη δημοσίευση του κάθε άρθρου, παρακολουθώντας έτσι τη δραστηριότητα ενός γνωστικού αντικειμένου. Το TOA διευκολύνει την ανάλυση των διαθέσιμων στοιχείων στα διάφορα άρθρα και έγγραφα παρουσιάζοντας καταλόγους με τις συχνά εμφανιζόμενες λέξεις-κλειδιά ή καταλόγους με τις συνεργασίες μεταξύ συγγραφέων, πόλεων ή και χωρών.

Είσοδος:

Χρησιμοποιήθηκε η βάση INSPEC (μπορούν να χρησιμοποιηθούν και άλλες βάσεις δεδομένων όπως η USPTO-United States Patent Database). Ακολούθως καθορίζεται μια λίστα με λέξεις-κλειδιά και οι πιθανοί συνδυασμοί τους με κανόνες Boolean. Εν συνεχεία υπολογίζονται οι εμφανίσεις και οι συνεμφανίσεις των διαφόρων λέξεων-κλειδιών για κάθε χρόνο αλλά και όλα τα χρόνια συνολικά. Με ένα δεύτερο πέρασμα επιλέγονται όλες οι φράσεις από ένα συγκεκριμένο πεδίο και υπολογίζεται το πλήθος των εμφανίσεων των φράσεων αυτών.

Αλγόριθμοι Μάθησης:

Στο TOA δεν υπάρχουν αλγόριθμοι μαθήσεως μιας και ο χρήστης είναι υπεύθυνος να αναλύσει τα δεδομένα που θα εξαγάγει το σύστημα και να αποφασίσει ποιες είναι οι ανερχόμενες τάσεις.

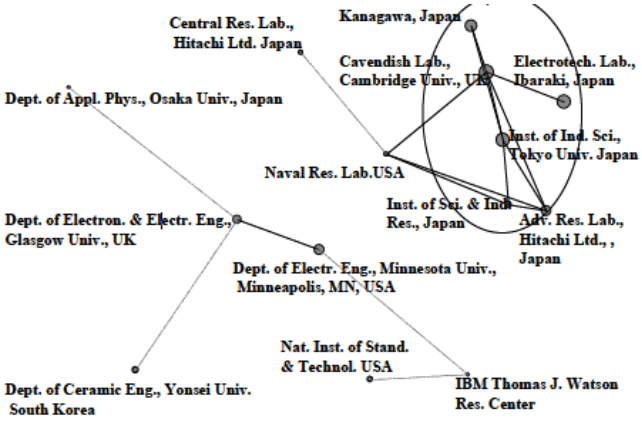
Παρουσίαση:

Η παρουσίαση των αποτελεσμάτων στο TOA γίνεται με πολλούς τρόπους. Χρησιμοποιούνται πίνακες συχνότητας, ιστογράμματα, σταθμισμένοι δείκτες κ.α. Αυτοί οι τρόποι παρουσίασης αποσκοπούν στο να βοηθήσουν το χρήστη να κατανοήσει καλύτερα τα αποτελέσματα και να τα παρουσιάσουν με πιο συνοπτικό τρόπο. Για την παρουσίαση αυτή χρησιμοποιούνται τεχνικές ομαδοποίησης και κλιμάκωσης σε πολλές διαστάσεις.

Αξιολόγηση:

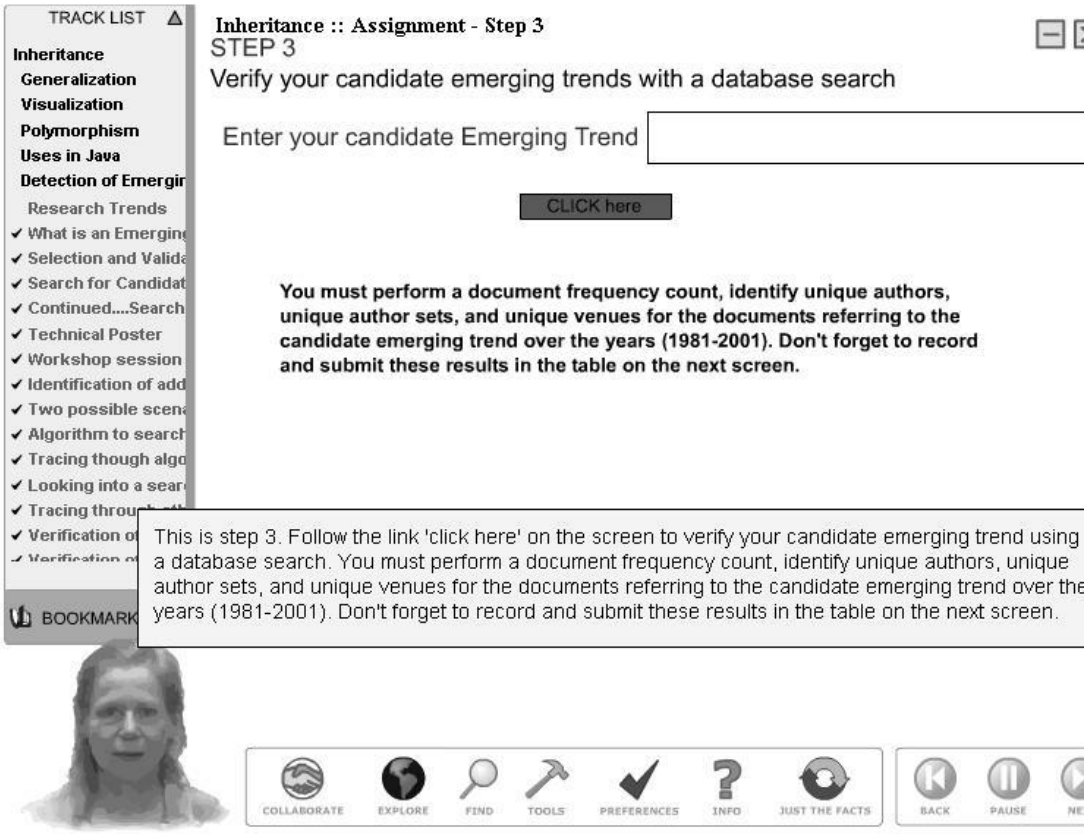
Το TOA μπορεί να αξιολογηθεί ως προς το πόσο καλή και χρήσιμη πληροφορία παρουσιάζει στο χρήστη ο οποίος πρέπει να αποφασίσει ποιες είναι οι πιθανές αναδυόμενες τάσεις. Για

την αξιολόγηση χρησιμοποιούνται ανεξάρτητες πηγές και ομάδες ατόμων σχετικές με το αντικείμενο οι οποίοι μπορούν να επιβεβαιώσουν πως τα δεδομένα που παρουσιάζονται με τις διάφορες τεχνικές απεικόνισης είναι όντως χρήσιμα.



Εικόνα 1: Οι ερευνητικοί οργανισμοί που έχουν σχέση με την έρευνα στον τομέα της νανοτεχνολογίας

2.3.2 CIMEL



Εικόνα 2: CIMEL αρχική σελίδα βοήθειας προς χρήστη (tutorial)

Το CIMEL είναι ένα multi-media πλαίσιο το οποίο χρησιμοποιεί μια ημι-αυτόματη μέθοδο ανίχνευσης νέων τάσεων για την ενίσχυση της εκπαίδευσης στην επιστήμη των υπολογιστών. Οι μαθητές εισάγουν στο CIMEL το θέμα που τους ενδιαφέρει και βλέπουν την έρευνα που γίνεται πάνω σε αυτό το πεδίο αλλά και τα νέα συνέδρια και ημερίδες σε αυτό το πεδίο. Η βάση δεδομένων του συστήματος αυτού μπορεί να είναι οποιοσδήποτε δικτυακός πόρος (Leon M. Galitsky, 2003).

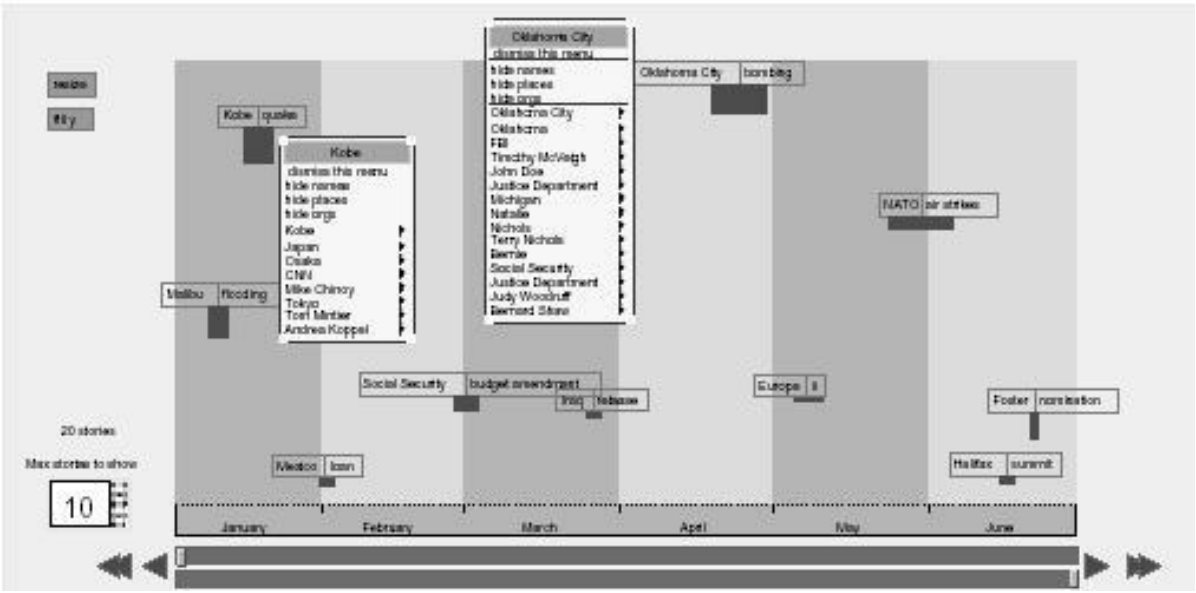
Είσοδος:

Για την εφαρμογή αυτή τα δεδομένα μπορούν να είναι οποιαδήποτε δεδομένα υπάρχουν ελεύθερα στο διαδίκτυο. Ακολούθως επιλέγεται μια περιγραφή του θέματος και γίνεται μια αρχική αναζήτηση συνεδρίων και ημερίδων, για να εντοπιστούν πιθανές τάσεις. Διάφορες μετρικές χρησιμοποιούνται για την αξιολόγηση των δεδομένων όπως ο πλήθος των φορών που εμφανίζεται το κύριο θέμα, η χρονολογία του άρθρου, ο αριθμός των παρόμοιων φράσεων στο κείμενο κ.α. Ακολούθως, υπολογίζονται αυτόματα τέσσερις συχνότητες στο χρόνο: ο αριθμός των εγγράφων, των εκδοτών, των χώρων συνεδρίων και ομάδες εκδοτών. Αυτές οι συχνότητες βοηθούν το χρήστη στην τελική του απόφαση.

Αλγόριθμοι Μάθησης:

Όπως και στο TOA έτσι και στο CIMEL δεν υπάρχουν αλγόριθμοι μαθήσεως αλλά ο χρήστης πρέπει να αποφασίσει, αφού αξιολογήσει τα δεδομένα που θα του παρουσιαστούν, για το ποιες όντως είναι ανερχόμενες τάσεις.

2.2.3 TimeMines



Εικόνα 3: Παράδειγμα εξόδου του TimeMines

Το TimeMines είναι ένα πλήρως αυτοματοποιημένο σύστημα, το οποίο παρουσιάζει στους χρήστες έναν ιεραρχημένο χρονολογικό κατάλογο των θεμάτων που βρίσκονται στο σύνολο των άρθρων προς ανάλυση (Kontostathis, 2002). Οι χρήστες ακολούθως πρέπει να καταλάβουν κατά πόσο ένα θέμα είναι σχετικό με την περιοχή που τους ενδιαφέρει και αν όντως αποτελεί ανερχόμενη τάση. Το σύστημα ξεκινά την επεξεργασία με ένα προεπιλεγμένο μοντέλο το οποίο υποθέτει πως η εμφάνιση ενός θέματος βασίζεται σε μια βασική τιμή εμφάνισης που δε μεταβάλλεται με το χρόνο. Ακολούθως συγκρίνει κάθε θέμα με το μοντέλο αυτό και με στατιστικές μεθόδους βρίσκει αν το θέμα αποκλίνει σημαντικά ή όχι από το μοντέλο. Αν το θέμα δεν αποκλίνει αρκετά παραλείπεται. Αν όμως αποκλίνει τότε το θέμα περνά στη δεύτερη φάση επεξεργασίας κατά την οποία ομαδοποιούνται τα θέματα που σχετίζονται μεταξύ τους. Ακολούθως ένα εμπειρικό κατώφλι καθορίζει πόσα θέματα θα εμφανιστούν στους χρήστες.

Αλγόριθμοι Μάθησης:

Υπάρχουν δύο διαφορετικοί αλγόριθμοι μαθήσεως στο TimeMines. Αρχικά το TimeMines πρέπει να διαλέξει τα πιο σημαντικά θέματα και να τα παρουσιάσει στο χρήστη. Για να το κάνει αυτό το TimeMines πρέπει να εξάγει τα πιο σημαντικά χαρακτηριστικά από τα έγγραφα εισόδου και να τα αναλύσει.

Το TimeMines χρησιμοποεί ένα στατιστικό μοντέλο βασισμένο σε έλεγχο υποθέσεων για να μπορέσει να βρει τα πιο σημαντικά χαρακτηριστικά. Βασίζεται σε ένα προεπιλεγμένο μοντέλο όπου κάθε χαρακτηριστικό θεωρείται πως έχει μια συγκεκριμένη συχνότητα εμφάνισης που δεν αλλάζει με το χρόνο. Μετά κάθε χαρακτηριστικό αναλύεται και αν η πραγματική κατανομή του χαρακτηριστικού ταυτίζεται με το μοντέλο, τότε θεωρείται πως το χαρακτηριστικό αυτό δεν έχει νέα πληροφορία να δώσει και αγνοείται. Αν όχι, τότε τα χαρακτηριστικά αυτά διατηρούνται για περαιτέρω επεξεργασία.

Αφού επιλέξει όλα τα χαρακτηριστικά ενδιαφέροντος, το TimeMines χρησιμοποιεί ακόμα έναν αλγόριθμο μάθησης βασισμένο σε έλεγχο υποθέσεων. Το TimeMines πλέον κοιτά για χαρακτηριστικά τα οποία έχουν παρόμοια κατανομή σε συγκεκριμένα χρονικά διαστήματα. Αυτά τα χαρακτηριστικά στη συνέχεια ομαδοποιούνται σε ένα συγκεκριμένο θέμα και παρουσιάζονται στον χρήστη ως το θέμα αυτό. Έτσι, κάθε χρονική περίοδος μπορεί να συσχετιστεί με ένα μικρό αριθμό θεμάτων.

Παρουσίαση:

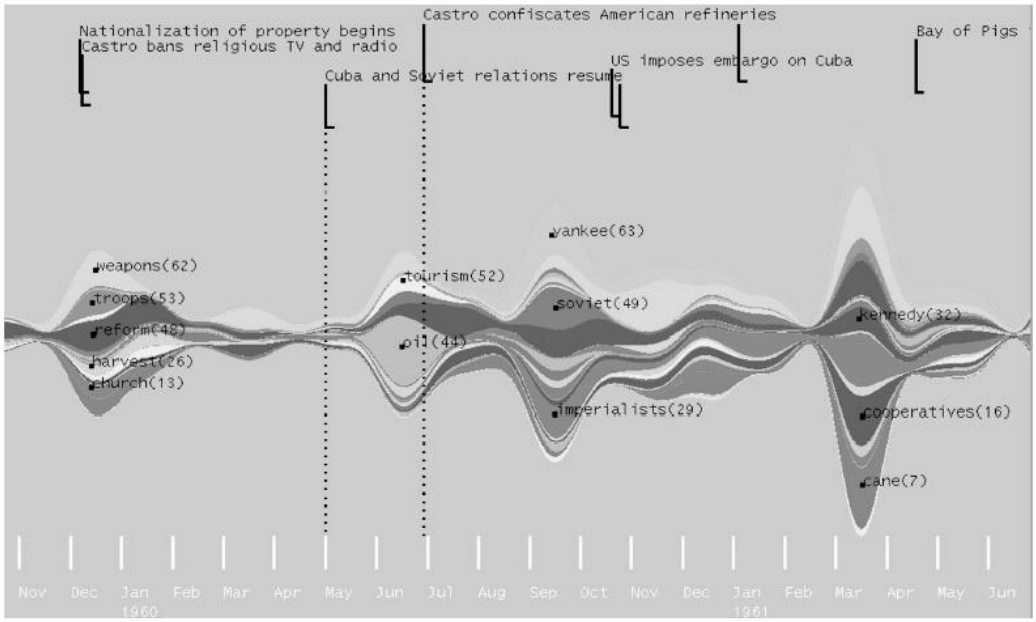
Το TimeMines παρουσιάζει γραφικά τα αποτελέσματά του έτσι ώστε να είναι πιο εύκολα κατανοητά στους χρήστες. Στη γραφική παρουσίαση ο x-άξονας αντιπροσωπεύει το χρόνο ενώ ο y-άξονας αντιπροσωπεύει τη σχετική σημασία του κάθε θέματος. Τα πιο στατιστικά σημαντικά θέματα παρουσιάζονται στην κορυφή. Με το πάτημα του κάθε θέματος παρουσιάζονται τα διάφορα χαρακτηριστικά του και ο χρήστης μπορεί να επιλέξει ένα χαρακτηριστικό για να δει περισσότερες πληροφορίες.

Αξιολόγηση:

Στο TimeMines πρέπει να αξιολογηθούν οι υποθέσεις οι οποίες γίνονται. Για την αξιολόγηση του κατά πόσο εξάγονται τα σωστά χαρακτηριστικά τα οποία έχουν πραγματικά νόημα σε ένα άνθρωπο ένα τεστ τυχαιοποίησης (randomization test) ακολουθήθηκε. Τα έγγραφα τοποθετήθηκαν σε τυχαία σειρά και μεταβλήθηκε η ημερομηνία έκδοσής τους. Από άποψη ανάκτησης πληροφοριών τα αποτελέσματα φαίνονταν τα ίδια και επομένως επήλθε το συμπέρασμα πως η επιλογή των χαρακτηριστικών είναι λογική και όχι τυχαία.

Όσον αφορά στη δεύτερη υπόθεση της ομαδοποίησης των χαρακτηριστικών, χρησιμοποιήθηκαν δύο τεχνικές οι οποίες όμως δεν μπόρεσαν να δώσουν ακριβή αποτελέσματα. Η πρώτη χρησιμοποιούσε τεχνικές ακρίβειας και ανάκλησης από την εξαγωγή δεδομένων και η δεύτερη προσπαθούσε να βρει το κατάλληλο κατώφλι για την παρουσίαση των αποτελεσμάτων.

2.3.4 ThemeRiver



Εικόνα 4: Παράδειγμα εξόδου του ThmeRiver

Το ThemeRiver είναι μια πλήρως αυτόματη μέθοδος εντοπισμού τάσεων, η οποία συνοψίζει τη θεματική περιοχή που ενδιαφέρει το χρήστη και την αναπαριστά σαν ένα «ποτάμι» πληροφοριών παρουσιάζοντας και τις αλλαγές σε αυτές με βάση το χρόνο. Αυτό το σύστημα δημιουργεί αυτόματα μια λίστα των πιθανών θεμάτων που ίσως να αποτελούν τάσεις, που ονομάζονται λέξεις-θέματα, εκ των οποίων ένα υποσύνολο, το οποίο θεωρείται πιο πιθανόν να αποτελεί ανερχόμενη τάση, επιλέγεται χειροκίνητα από τους χρήστες (Leon M. Galitsky, 2003).

Το «ποτάμι» αποτελείται από πολλαπλές ροές. Κάθε ρεύμα αντιπροσωπεύει ένα θέμα και κάθε θέμα αντιπροσωπεύεται από ένα χρώμα και διατηρεί τη θέση του στο ποτάμι, σε σχέση με άλλα θέματα.

Είσοδος:

Το ThemeRiver αυτόματα δημιουργεί μια λίστα από πιθανά θεμάτα, που ονομάζονται "λέξεις-θέματα", εκ των οποίων ένα υποσύνολο επιλέγεται χειροκίνητα για τα χαρακτηριστικά του. Είσοδος για το σύστημα αποτελεί η μέτρηση των πόσων εγγράφων περιέχουν μια συγεκριμένη λέξη-θέμα για κάθε χρονική περίοδο.

Αλγόριθμοι Μάθησης:

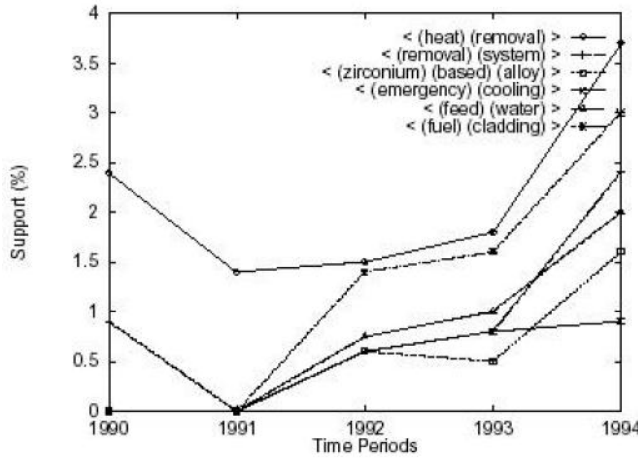
Το ThemeRiver δεν υλοποιεί κάποιο αλγόριθμο μάθησης αλλά όπως το TOA παρουσιάζει μια συνοπτική και με γραφικό τρόπο εικόνα των δεδομένων, και ακολούθως ένας έμπειρος χρήστης μπορεί να αποφασίσει για το ποιες είναι οι ανερχόμενες τάσεις.

Το ThemeRiver ξεκινάει χωρίζοντας τα διάφορα άρθα σε διάφορα χρονικά διαστήματα. Ένα σύνολο από θέματα επιλέγεται και δημιουργείται το «ποτάμι» βασισμένο στη σημαντικότητα του κάθε θέματος. Τα θέματα διαλέγονται χειροκίνητα από τους χρήστες αφού τους παρουσιαστεί μια λίστα από λέξεις που αντιπροσωπεύουν τα δεδομένα. Ο αριθμός των αντικειμένων που περιέχουν τις λέξεις που επιλέγονται αντιπροσωπεύει και το πόσο σημαντικό είναι ένα θέμα στο ποτάμι.

Παρουσίαση:

Το ThemeRiver χρησιμοποεί μια παρουσίαση ως ποτάμι, για να αναπαραστήσει τη ροή των δεδομένων στο χρόνο. Κάθε κάθετο κομμάτι του ποταμού αντιπροσωπεύει και ένα χρονικό διάστημα και κάθε χρωματισμένη ροή μέσα στο ποτάμι αντιπροσωπεύει ένα θέμα. Επίσης θέματα τα οποία σχετίζονται μεταξύ τους χρωματίζονται με σκιές του ίδιου χρώματος έτσι ώστε ο χρήστης να μπορεί εύκολα να αναγνωρίσει ότι συνδέονται.

2.3.5 PatentMiner



Εικόνα 5: Παράδειγμα εξόδου του PatentMiner

Το σύστημα PatentMiner αναπτύχθηκε για να ανακαλύπτει τάσεις στα δεδομένα διπλωμάτων ευρεσιτεχνίας (πατέντες) χρησιμοποιώντας δυναμικά δημιουργημένα SQL ερωτήματα με βάση τα κριτήρια επιλογής που δίνονται από το χρήστη. Το σύστημα είναι συνδεδεμένο σε μια βάση δεδομένων IBM DB2 που περιέχει όλα τα χορηγημένα διπλώματα ευρεσιτεχνίας των ΗΠΑ [USPTO]. Υπάρχουν δύο κύρια συστατικά του συστήματος, η ταυτοποίηση φράσεων χρησιμοποιώντας μοτίβο εξόρυξης ακολουθιών, καθώς και ανίχνευση τάσεων με τη χρήση ερωτημάτων (Kontostathis, 2002).

Είσοδος:

Ως είσοδος στο σύστημα χρησιμοποιήθηκε η βάση δεδομένων IBM DB2 που περιέχει όλα τα διπλώματα ευρεσιτεχνίας των ΗΠΑ [USPTO]. Αρκετές διαδικασίες εφαρμόστηκαν για την προετοιμασία των δεδομένων προκειμένου να γίνει εξαγωγή των σημαντικότερων χαρακτηριστικών: οι τελείες αφαιρέθηκαν, σε κάθε λέξη έχει ανατεθεί ένας ακέραιος αριθμός που υποδεικνύει τη θέση της λέξης στο έγγραφο και τις εμφανίσεις σε προτάσεις, παραγράφους και τμήματα. Ακολούθως, αφού δημιουργηθεί ένα υποσύνολο των διπλωμάτων ευρεσιτεχνίας χωρισμένο σε κατηγορίες και χρονικές περιόδους, χρησιμοποιείται ο αλγόριθμος Generalized Sequential Patterns (GSP) ο οποίος διαλέγει τα σημαντικότερα χαρακτηριστικά τα οποία ορίζονται από το χρήστη και ονομάζονται φράσεις. Μια φράση μπορεί να είναι οποιαδήποτε ακολουθία λέξεων, με ελάχιστο και μέγιστο "χάσμα" μεταξύ των λέξεων. Τα κενά μπορούν να είναι λέξεις, προτάσεις, παραγράφοι ή ενότητες.

Αλγόριθμοι Μάθησης:

Το PatentMiner χρησιμοποιεί μια τεχνική ταιριάσματος διαδοχικών μοτίβων που χρησιμοποιείται συχνά σε συστήματα εξόρυξης δεδομένων. Το σύστημα κοιτάζει για μοτίβα λέξεων που εμφανίζονται συχνά. Αυτή η τεχνική επιτρέπει στο σύστημα να εντοπίζει λέξεις που εμφανίζονται συχνά μαζί και να τις θεωρεί ως ένα θέμα. Τα δεδομένα εισόδου χωρίζονται σε διάφορες ομάδες ανάλογα με την ημερομηνία δημοσίευσής τους. Ακολούθως, η παραπάνω τεχνική εφαρμόζεται σε κάθε ομάδα και εξάγονται φράσεις και υπολογίζεται η συχνότητα εμφάνισης της κάθε φράσης. Στη συνέχεια δημιουργείται ένα ερώτημα (query) το οποίο εξάγει τις πιο σημαντικές φράσεις με βάση τα δεδομένα που δίνει ο χρήστης. Ακολούθως είναι δουλειά του χρήστη να βρει ποιες φράσεις όντως αποτελούν τάσεις στα διπλώματα ευρεσιτεχνίας με βάση τα αποτελέσματα.

Παρουσίαση:

Το σύστημα είναι διαδραστικό. Ένα ιστόγραμμα δείχνει τις εμφανίσεις των διπλωμάτων ευρεσιτεχνίας ανά έτος με βάση τα κριτήρια επιλογής του χρήστη. Ο χρήστης έχει τη δυνατότητα αργότερα να επικεντρωθεί σε ένα συγκεκριμένο χρονικό διάστημα και να επιλέξει διάφορα ερωτήματα (queries) για να διερευνήσει τις πιθανές τάσεις.

2.3.6 HDDI

Το HDDI (Hierarchical Distributed Dynamic Indexing) έχει ως στόχο να προσδιορίσει τα χαρακτηριστικά και τις μεθόδους, για να βελτιωθεί η αυτόματη ανίχνευση των αναδυόμενων τάσεων με τη δημιουργία συστάδων που θα βασίζονται στη σημασιολογική ομοιότητα των άρθρων που μας ενδιαφέρουν (Leon M. Galitsky, 2003). Χρησιμοποιεί νευρωνικά δίκτυα για την ταξινόμηση των θεμάτων σε αναδυόμενες ή μη αναδυόμενες τάσεις.

Το σύστημα HDDI περιέχει επεξεργαστή κειμένου ο οποίος μπορεί να εξάγει σημαντικές πληροφορίες και χαρακτηριστικά από τα κείμενα αλλά και αλγορίθμους εξόρυξης κειμένου και μηχανικής μάθησης που το βοηθούν αν βρίσκει τις ανερχόμενες τάσεις.

Το HDDI χρησιμοποιείται για την εξαγωγή γλωσσικών στοιχείων από μια συλλογή από άρθρα και ακολούθως δημιουργούνται συστάδες με βάση την σημασιολογική ομοιότητα των χαρακτηριστικών που εξάγονται από τα άρθρα. Ο αλγόριθμος παίρνει ένα στιγμιότυπο της στατιστικής κατάστασης μιας συλλογής από άρθρων σε διάφορα χρονικά σημεία και παρατηρεί το ρυθμό μεταβολής στο μέγεθος των συστάδων αλλά και τη συχνότητα που εμφανίζονται τα διάφορα χαρακτηριστικά των άρθρων. Ακολούθως αυτές οι μεταβολές στα χαρακτηριστικά και στο μέγεθος των συστάδων χρησιμοποιούνται σαν είσοδος σε ένα νευρωνικό δίκτυο το οποίο ταξινομεί τα διάφορα θέματα ως ανερχόμενες ή οχι τάσεις.

Είσοδος:

Ως είσοδος στο HDDI χρησιμοποιούνται τέσσερις βάσεις δεδομένων: η βάση δεδομένων διπλωμάτων ευρεσιτεχνίας των ΗΠΑ (USPTO - http://www.uspto.gov/), η βάση δεδεομένων διπλωμάτων ευριστεχνίας της IBM (Delphion - http://www.delphion.com/), η βάση δεδομένων INSPEC - http://www.theiet.org/resources/inspec/, και η βάση δεδομένων COMPENDEX - http://www.engineeringvillage.com/controller/servlet/Controller. Τα δεδομένα αυτά αρχικά χρειάζονται ανάλυση από ένα διαχωριστή (parser) ο οποίος κρατά μόνο τα κομμάτια των κειμένων που έχουν ενδιαφέρον. Ακολούθως μια μηχανή πεπερασμένων καταστάσεων εξάγει περίπλοκες φράσεις (ονομάζονται ενοιολογικά ζεύγη) χρησιμοποιώντας κανωνικές εκφράσεις (regular expressions). Στη συνέχεια προσδιορίζεται η ομοιότητα αυτών των ζευγών κοιτάζοντας το κατά πόσο εμφανίζονται συχνά μαζί. Ακολούθως διάφορες μετρικές εξάγονται έτσι ώστε το σύστημα να μπορεί να χωρίσει τα δεδομένα σε συστάδες και μετά ένα νευρωνικό δίκτυο χρησιμοποιείται με είσοδο τις συστάδες και διάφορες παραμέτρους όπως το πόσες φορές εμφανίζονται αυτά τα ζεύγη σε διάφορα χρονικά διαστήματα, για να καθορίσει αν αποτελούν ανερχόμενες τάσεις.

Αλγόριθμοι μαθήσεως:

Το HDDI χρησιμοποιεί την παραδοχή ότι οι αναδυόμενες τάσεις μπορούν να ανιχνευτούν αυτόματα μόνο με αλγόριθμους, ανιχνεύοντας τις αλλαγές στη συχνότητα με την οποία εμφανίζονται και συσχετίζονται διάφορες εκφράσεις στην πάροδο ενός χρονικού διαστήματος. Η προσέγγιση αυτή περιλαμβάνει διαχωρισμό των δεδομένων σε συστάδες

με βάση το χρόνο (όπως το PatentMiner και το TimeMines) και ακολούθως λαμβάνει στιγμιότυπα των σημασιολογικών σχέσεων μεταξύ των όρων. Δύο ιδιαίτερα χαρακτηριστικά ελήφθησαν υπόψη για την κατασκευή του μοντέλου. Το πρώτο είναι ότι η συχνότητα εμφάνισης ενός όρου ή μιας φράσης πρέπει να αυξάνεται όσο περνά ο χρόνος αν αυτός ο όρος ή η φράση είναι ανερχόμενη και ταυτόχρονα ο όρος/φράση θα πρέπει να συνυπάρχει με ένα αυξανόμενο αριθμό άλλων όρων/φράσεων.

Το μοντέλο μάθησης που χρησιμοποιήθηκε είναι ένα νευρωνικό δίκτυο με είσοδο διάφορες παραμέτρους που εξάχθηκαν με διάφορους αλγόριθμους εξαγωγής δεδομένων όπως δέντρα αποφάσεων και μηχανές διανυσμάτων. Ακόμα και εδώ όμως την τελική απόφαση για το αν ένας όρος ή φράση αποτελούν ανερχόμενη τάση την έχει ένας ειδικός.

2.4 Ανοιχτά Ζητήματα

Παραπάνω είδαμε συστήματα εντοπισμού ανερχόμενων τάσεων που είναι είτε ημιαυτόματα είτε πλήρως αυτόματα και παρέχουν στους χρήστες μια σύντομη περίληψη του περιεχομένου αλλά και των χαρακτηριστικών ενός μεγάλου αριθμού από επιστημονικά άρθρα. Η παρουσιάση αυτή γίνεται είτε με τη μορφή κειμένου, είτε με την οπτικοποίησή του με τη χρήση γραφικών των διαθέσιμων πληροφοριών.

Παρ’ όλα αυτά από την ανάλυση των διαφόρων συστημάτων προκύπτει πως στα περισσότερα συστήματα, αφού παρουσιαστεί η έξοδος στο χρήστη, απαιτείται η γνώση ενός ειδικού για να αποφασίσει αν όντως τα παρουσιαζόμενα αποτελέσματα αποτελούν ανερχόμενες τάσεις ή όχι. Έτσι, παρόλη την πρόοδο που έχει γίνει προς την πλήρη αυτοματοποίηση της διαδικασίας ανίχνευσης ανερχόμενων τάσεων, υπάρχει ακόμα αρκετό περιθώριο για βελτίωση στον τομέα αυτό. Επομένως η έρευνα πάνω στη δημιουργία αξιόπιστων και αποτελεσματικών συστημάτων που να είναι σε θέση να εντοπίσουν τις αναδυόμενες τάσεις είναι αναγκαίο να συνεχίσει και να εμπλουτιστεί.

Επίσης περισσότερη έρευνα απαιτείται και στο κομμάτι της παρουσίασης των αποτελεσμάτων ενός συστήματος ανίχνευσης τάσεων. Ο τρόπος με τον οποίο θα παρουσιαστούν στο χρήστη τα αποτελέσματα παίζει σημαντικό ρόλο στην κατανόησή τους. Ένα γραφικό περιβάλλον, το οποίο θα ξεχωρίζει τις διάφορες ανερχόμενες τάσεις με τρόπο εύκολα κατανοητό από τους χρήστες (π.χ. με διαφορετικά χρώματα) και θα τους δίνει τη δυνατότητα να αναλύσουν με ευκολία τα διάφορα χαρακτηριστικά του κάθε θέματος, διευκολύνει πολύ τους χρήστες και αυξάνει το ποσοστό κατανόησης των αποτελεσμάτων.

Επιπλέον αρκετή έρευνα απαιτείται ακόμη για τη βελτιστοποίηση των διαφόρων αλγορίθμων μάθησης. Στα παραπάνω συστήματα που έχουμε δει, χρησιμοποιούνται διάφοροι αλγόριθμοι μαθήσεως οι οποίοι όμως πολλές φορές παράγουν διαφορετικά αποτελέσματα και ακόμα δεν υπάρχει κάποιος αλγόριθμος ο οποίος να υπερτερεί έναντι των άλλων. Για ένα σύστημα εντοπισμού ανερχόμενων τάσεων οι αλγόριθμοι που θα χρησιμοποιηθούν απαιτείται να είναι γρήγοροι, αξιόποιστοι αλλά και ευέλικτοι έτσι ώστε να μπορούν να ανταποκριθούν σε διάφορα επιστημονικά άρθρα και βάσεις δεδομένων

χωρίς να χρειάζεται αλλαγή του αλγορίθμου για διαφορετικές εισόδους, για παράδειγμα ως προς το επιστημονικό πεδίο.

Επιπρόσθετα παρατηρούμε μέσα από τη βιβλιογραφία πως δεν έχει δοθεί μεγάλη έμφαση από τους ερευνητές στο κομμάτι της αξιολόγησης των διαφόρων συστημάτων και τεχνικών και δεν έχουν αναπτυχθεί επιστημονικές μεθοδολογίες για το σκοπό αυτό. Η ανάπτυξη τέτοων μεθολογιών κρίνεται απαραίτητη καθώς δε γνωρίζουμε ακριβώς την αποτελεσματικότητά τους και το βαθμό αξιοπιστίας των υπαρχόντων συστημάτων, ιδιαιτέρως σε συνθήκες πραγματικού κόσμου. Επίσης λόγω του ότι το κάθε σύστημα χρησιμοποεί διαφορετικές τεχνικές για την ανίχνευση ανερχόμενων τάσεων, ένα γενικό σύστημα αξιολόγησης των αποτελεσμάτων θα επέτρεπε την σύγκριση μεταξύ των διαφόρων μεθόδων και επομένως την καλύτερη κατανόηση των δυνατοτήτων και των αδυναμιών του κάθε συστήματος. Αυτό μπορεί να επιτευχθεί με τη δημιουργία ειδικά σχεδιασμένων αρχείων εισόδου, για τα οποία να γνωρίζουμε εκ των προτέρων τα αποτελέσματα και τα οποία να μπορούν να εφαρμοστούν σε διάφορα συστήματα. Επιπλέον χρειάζεται να γίνουν και μελέτες οι οποίες θα δείχνουν το βαθμό ευκολίας και χρησιμοποίησης της κάθε εφαρμογής από τους χρήστες, γεγονός που θα οδηγήσει στην περαιτέρω βελτίωση των εφαρμογών αυτών.

Τέλος στα παραπάνω συστήματα παρατηρούμε πως δινόταν ιδιαίτερη έμφαση είτε στην ανάπτυξη αποτελεσματικών αλγορίθμων μάθησης είτε στην παρουσίαση των αποτελεσμάτων με ένα εύχρηστο γραφικό περιβάλλον. Ένας συνδυασμός των δύο θα αποτελούσε μια ακόμα καλύτερη λύση και θα οδηγούσε σε πιο ολοκληρωμένα συστήματα εντοπισμού ανερχόμενων τάσεων.

Κεφάλαιο 3: Εξόρυξη Δεδομένων

Η εξόρυξη δεδομένων είναι ένας τομέας πολλά υποσχόμενος και ανερχόμενος στην επιστήμη των υπολογιστών με πολλές και χρήσιμες εφαρμογές. Πολλοί ερευνητές έχουν ασχοληθεί με τον τομέα αυτόν με την αντίστοιχη βιβλιογραφία να είναι πλούσια σε δημοσιευμένα άρθρα και έρευνες που προσπαθούν να δώσουν λύσεις σε προβλήματα που αφορούν στην εξόρυξη δεδομένων. Παρακάτω θα ασχοληθούμε με τα βασικά προβλήματα που αντιμετώπισαν και αντιμετωπίζουν οι ερευνητές και που τους ώθησαν να ασχοληθούν με την εξόρυξη δεδομένων αλλά και μια γενική παρουσίαση του πεδίουκαι της τρέχουσας τεχνολογικής κατάστασης σε αυτό. Ακουλούθως θα παρουσιαστούν κάποιες υπάρχουσες εφαρμογές και αλγόριθμοι εξόρυξης δεδομένων και θα γίνει μια αναφορά για το πώς συνδέεται η εξόρυξη δεδομένων με τον εντοπισμό τάσεων.

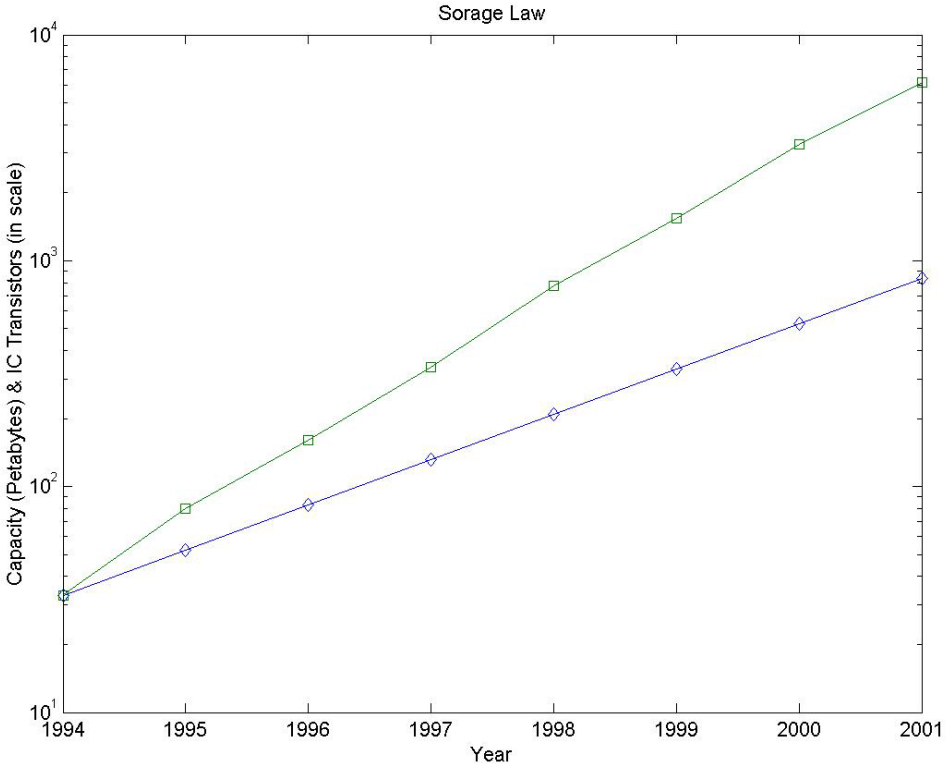
3.1 Προβλήματα που οδήγησαν στη δημιουργία του πεδίου εξόρυξης δεδομένων

3.1.1 Ραγδαία αύξηση στον όγκο των δεδομένων

Η εξαγωγή πληροφορίας και δεδομένων από μεγάλες συλλογές πληροφοριών αποτελεί μια επίπονη και χρονοβόρα δραστηριότητα διαχρονικά. Παρ' όλα αυτά τα τελευταία χρόνια με την αύξηση της δύναμης της τεχνολογίας των υπολογιστών αλλά και την πανταχού παρουσία τους στις ζωές των ανθρώπων, ο όγκος των διαθέσιμων δεδομένων αυξήθηκε ραγδαία με αποτέλεσμα το οποιοδήποτε χειροκίνητο πέρασμα από τα δεδομένα να καταστεί υπολογιστικά αδύνατον ή έστω αναποτελεσματικό. Καθώς τα δεδομένα αυξάνονταν σε μέγεθος και πολυπλοκότητα νέες τεχνικές έπρεπε να εφευρεθούν έτσι ώστε να γίνει δυνατή η προσπέλασή τους αποτελεσματικά και σε αποδεκτούς χρόνους. Έτσι η παραδοσιακή χειροκίνητη προσπέλαση των δεδομένων εξελίχθηκε πολύ από άλλες ανακαλύψεις στην επιστήμη των υπολογιστών όπως τα νευρωνικά δίκτυα, την ανάλυση συστάδων και τα δέντρα αποφάσεων.

3.1.2 Το κενό μεταξύ της αύξησης της επεξεργαστικής δύναμης και της αύξησης των δεδομένων

Ο νόμος του Moore δηλώνει πως η επεξεργαστική δύναμη των υπολογιστών διπλασιάζεται περίπου κάθε 18 μήνες. Επίσης όμως ισχύει πως ο συνολικός όγκος που πρέπει να επεξεργαστούν οι υπολογιστές διπλασιάζεται κάθε 12 μήνες (Porter, 1998). Το γεγονός αυτό δημιουργεί ένα κενό μεταξύ της αύξησης της επεξεργαστικής δύναμης και της αύξησης του όγκου των δεδομένων (U. Fayyad R. U., 2002). Τα δεδομένα αυξάνονται πολύ πιο γρήγορα από τις υπολογιστικές δυνατότητες των μηχανών. Η διαφορά αυτή αυξάνεται εκθετικά και ονομάζεται το κενό των δεδομένων όπως φαίνεται και στο παρακάτω διάγραμμα. Έτσι οι συμβατικοί αλγόριθμοι που είναι σχεδιασμένοι για μικρότερα μεγέθη δεδομένων παρουσιάζουν προβλήματα, καθώς ο όγκος των δεδομένων αυξάνεται και το υλικό δεν μπορεί να ανταπεξέλθει στην αύξηση αυτή. Το μεγαλύτερο πρόβλημα των αλγορίθμων αυτών είναι πως με την αύξηση των δεδομένων αυξάνεται κατά πολύ ο χρόνος εκτέλεσης του αλγορίθμου. Έτσι, παραδείγματος χάριν, αν έχουμε ένα αλγόριθμο ο οποίος τρέχει με χρονική πολυπλοκότητα $O(n^2)$ η οποία παραδοσιακά για μικρά μεγέθη δεδομένων είναι αποδεκτή, αν αυξήσουμε το μέγεθος των δεδομένων κατά δισεκατομμύρια τάξης μεγέθους, τότε ο χρόνος εκτέλεσης θα αυξηθεί δραματικά κάνοντας τον αλγόριθμό μας ανεπαρκή.



Εικόνα 6: Γράφημα που παρουσιάζει το κενό μεταξύ της αύξησης της επεξεργαστικής δύναμης και της αύξησης των δεδομένων

Επομένως η αύξηση στον όγκο των δεδομένων οδήγησε τους επιστήμονες στο να ψάξουν να βρουν άλλες τεχνικές, για να μπορέσουν να αντλήσουν χρήσιμες πληροφορίες από τον ολοένα και αυξανόμενο όγκο δεδομένων.

Έτσι δημιουργήθηκε η ανάγκη για την ύπαρξη ενός συστήματος που θα πληρούσε τους εξής δύο στόχους:

1. Να μπορεί να διαχειριστεί μεγάλο όγκο δεδομένων
2. Να έχει την απαραίτητη " ευφυΐα " έτσι ώστε να μπορεί αυτόματα να αναγνωρίσει και να εξάγει σημαντικές πληροφορίες από τα δεδομένα.

Έτσι η εξόρυξη δεδομένων είναι η διαδικασία της εφαρμογής διάφορων μεθόδων και τεχνικών με σκοπό την ανακάλυψη κρυμμένων μοτίβων και χρήσιμων πληροφοριών σε μεγάλα σύνολα δεδομένων.

3.2 Το πεδίο της εξόρυξης δεδομένων

Εξόρυξη δεδομένων είναι γενικά η διαδικασία ανάλυσης δεδομένων από διαφορετικές σκοπιές, συνοψίζοντας τα δεδομένα αυτά σε χρήσιμες πληροφορίες που μπορούν να αξιοποιηθούν για διάφορους σκοπούς. Το λογισμικό εξόρυξης δεδομένων είναι μια σειρά από αναλυτικά εργαλεία για την ανάλυση των δεδομένων αυτών. Έτσι οι χρήστες μπορούν να αναλύσουν τα δεδομένα από διαφορετικές οπτικές γωνίες και να συνοψίσουν ή και να κατηγοριοποιήσουν τις σχέσεις μεταξύ των δεδομένων. Έπομένως τεχνικά η εξόρυξη δεδομένων είναι η διαδικασία της εύρεσης συσχετισμών ή μοτίβων σε μεγάλες συλλογές από δεδομένα.

Με τις διάφορες τεχνικές εξόρυξης δεδομένων μπορούμε να αναλύσουμε αυτόματα ή ημιαυτόματα μεγάλες ποσότητες δεδομένων και να εξάγουμε πληροφορία η οποία προηγουμένως ήταν άγνωστη, όπως ενδιαφέροντα μοτίβα στα δεδομένα, ομάδες δεδομένων και ανάλυση κατά συστάδες αλλά και ανίχνευση ανωμαλιών ή και συσχετίσεων ανάμεσα στα δεδομένα. Για την επίτευξη αυτού του σκοπού συνήθως χρησιμοποιούνται τεχνικές βάσεων δεδομένων.

Ακολούθως αυτά τα διάφορα μοτίβα μπορούν να χρησιμοποιηθούν σαν ένα είδος περίληψης των αρχικών δεδομένων, να αναλυθούν περισσότερο και να χρησιμοποιηθούν είτε για μηχανική μάθηση, είτε για εύρεση τάσεων είτε για άλλους σκοπούς. Παρ΄όλα αυτά η συλλογή των δεδομένων, η προετοιμασία τους αλλά και η ερμηνεία των αποτελεσμάτων δεν αποτελούν μέρος της εξόρυξης δεδομένων αλλά ανήκουν γενικότερα στη διαδικασία εύρεσης γνώσης σε πηγές δεδομένων.

Η εξόρυξη δεδομένων είναι ένα διεπιστημονικό πεδίο της επιστήμης των υπολογιστών και χρησιμοποιεί διάφορες μεθόδους που σχετίζονται με την τεχνητή νοημοσύνη, τη μηχανική μάθηση (machine learning), τη στατιστική και τα συστήματα βάσεων δεδομένων όπως φαίνεται παρακάτω.

Η εξόρυξη δεδομένων σχετίζεται άμεσα με τη στατιστική μιας και των δύο σκοπός είναι η εύρεση χρήσιμης πληροφορίας και μοτίβων σε δεδομένα. Πολλές από τις τεχνικές που χρησιμοποιούνται για την εξόρυξη δεδομένων είναι ουσιαστικά στατιστικές τεχνικές προσαρμοσμένες σε υπολογιστικές και αλγοριθμικές απαιτήσεις. Μερικές στατιστικές τεχνικές που χρησιμοποιούνται στην εξόρυξη δεδομένων είναι η ανάλυση συστάδων, η ανάλυση παλινδρόμησης και τα διαστήματα εμπιστοσύνης.

Άλλα δύο πεδία της επιστήμης των υπολογιστών που σχετίζονται άμεσα με την εξόρυξη δεδομένων είναι η τεχνητή νοημοσύνη και η μηχανική μάθηση. Στόχος της τεχνητής νοημοσύνης αλλά και της μηχανικής μάθησης είναι η ανάλυση ακατέργαστων δεδομένων από μηχανές και η κατανόησή τους, κάτι πολύ κοντινό στην ιδέα της εξόρυξης δεδομένων. Πολλές από τις τεχνικές τόσο της τεχνητής νοημοσύνης όσο και της μηχανικής μάθησης χρησιμοποιούνται στην εξόρυξη δεδομένων όπως τα νευρωνικά δίκτυα, τα δέντρα αποφάσεων και οι ευριστικοί αλγόριθμοι.

Τέλος, η εξόρυξη δεδομένων είναι άμεσα συσχετιζόμενη με τα συστήματα βάσεων δεδομένων. Για να μπορέσουμε να χρησιμοποιήσουμε τις τεχνικές της εξόρυξης δεδομένων και να εξάγουμε χρήσιμη πληροφορία, σε μεγάλο όγκο δεδομένων, τα δεδομένα αυτά πρέπει να είναι αποθηκευμένα κάπου με δομημένο και εύκολα προσβάσιμο τρόπο. Επομένως μια καλή γνώση των συστημάτων βάσεων δεδομένων είναι χρήσιμη για την ενασχόληση με την εξόρυξη δεδομένων.

3.3 Τρέχουσα τεχνολογική κατάσταση

Στο σημείο αυτό θα παρουσιάσουμε μια σύντομη περιγραφή της τρέχουσας τεχνολογικής κατάστασης όσον αφορά στο πεδίο της εξόρυξης δεδομένων. Θα παρουσιαστούν οι κυριότεροι αλγόριθμοι οι οποίοι χρησιμοποιούνται αυτή τη στιγμή στον τομέα αυτό.

Η εξόρυξη δεδομένων περιλαμβάνει τις εξής κατηγορίες εργασιών (Fayyad Usama, 1996):

- Ανίχνευση Ανωμαλιών: η ανίχνευση διαφόρων ανωμαλιών ή λαθών στα δεδομένα η οποία μπορεί να μας φανεί χρήσιμη.
- Μάθηση εξάρτησης/Μάθηση με κανόνες συσχέτισης (Association rule learning, Dependency modeling): ψάχνει για τις σχέσεις μεταξύ των διάφορων μεταβλητών.
- Δημιουργία συστάδων: ομαδοποίηση ή κατηγοριοποίηση των δεδομένων σε ομάδες που μπορεί να έχουν κάποιο κοινό χαρακτηριστικό χωρίς όμως να χρησιμοποιούμε ήδη γνωστές δομές στα δεδομένα.
- Ταξινόμηση: κατάτασει τα νέα δεδομένα σε ήδη γνωστές δομές.
- Παλινδρόμηση (Regression): προσπαθεί να βρει μια λειτουργία η οποία μοντελοποιεί τα δεδομένα με το μικρότερο δυνατό σφάλμα.
- Ανακεφαλαιοποίηση: παρουσιάζονται τα δεδομένα με έναν πιο συμπαγή τρόπο χρησιμοποιώντας διάφορες τεχνικές παρουσίασης.
- Εξόρυξη διαδοχικών μοτίβων: βρίσκει σύνολα δεδομένων που εμφανίζονται μαζί σε ορισμένες αλληλουχίες. Έτσι εξάγει συχνές υποακολουθίες από βάσεις δεδομένων ακολουθιών και αποτελεί τη βάση για πολλές σύγχρονες εφαρμογές.

3.3.1 Ανίχνευση Ανωμαλιών

Η ανίχνευση ανωμαλιών ασχολείται με την εύρεση μοτίβων σε ένα σύνολο από δεδομένα τα οποία δεν αντοποκρίνονται σε μια αναμενόμενη συμπεριφορά. Τα μοτίβα αυτά ονομάζονται ανωμαλίες και χρησιμοποιούνται ως σημαντική πληροφορία σε διάφορες εφαρμογές (Hans-Peter Kriegel, 2009).

Υπάρχουν τρεις γενικές κατηγορίες τεχνικών ανίχνευσης ανωμαλιών:

- Τεχνικές ανίχνευσης ανωμαλιών χωρίς επίβλεψη: ανιχνεύουν ανωμαλίες σε δοκιμαστικά σετ δεδομένων υποθέτοντας πως η πλειοψηφία του συνόλου των

δεδομένων είναι φυσιολογικά και ανιχνεύοντας τις περιπτώσεις όπου τα δεδομένα δεν ταιριάζουν με το υπόλοιπο σύνολο των δεδομένων.

- Τεχνικές ανίχνευσης ανωμαλιών με επίβλεψη: απαιτεί ένα σύνολο από δεδομένα τα οποία θεωρούνται φυσιολογικά και ένα σύνολο δεδομένων που θεωρούνται μη φυσιολογικά και εφαρμόζεται ένας ταξινομητής ο οποίος εκπαιδεύεται στα δεδομένα.
- Τεχνικές ανίχνευσης ανωμαλιών με ημι-επίβλεψη: κατασκευάζεται ένα μοντέλο που αντιπροσωπεύει τη φυσιολογική συμπεριφορά με βάση κάποια δοκιμαστικά δεδομένα και ακολούθως κατατάσει τα νέα δεδομένα με βάση την πιθανότητα να μπορούν να παραχθούν από το μοντέλο αυτό.

Βασικές τεχνικές που χρησιμοποιούνται είναι:

- Τεχνικές βασισμένες στην απόσταση (k-nearest neighbors)
- Ανάλυση συστάδων με βάση την ανίχνευση ακραίων τιμών
- Βρίσκοντας δεδομένα που αποκλίνουν από τους κανόνες συσχέτισης μάθησης

3.3.2 Μάθηση με κανόνες Συσχέτισης

Με τους κανόνες συσχέτισης (association rule learning) ανιχνεύονται σχέσεις μεταξύ διαφόρων μεταβλητών σε μεγάλα συστήματα βάσεων δεδομένων (Piatetsky-Shapiro, 1991). Παραδείγματος χάριν στα ράφια ενός σούπερ μάρκετ μπορεί να υπάρξει συσχέτιση της αγοράς δύο προϊόντων από τον καταναλωτή με την άμεση αγορά ενός τρίτου. Αν αυτό το γνωρίζει εκ των προτέρων ο καταστηματάρχης, τότε μπορεί να προβεί σε καλύτερη προώθηση των προϊόντων του.

Για να ορίσουμε επακριβώς τη μάθηση με κανόνες συσχέτισης ας υποθέσουμε ότι έχουμε ένα σύνολο από αντικείμενα και ένα σύνολο από συναλλαγές με κάθε συναλλαγή να περιέχει κάποια αντικείμενα από το σύνολο των αντικειμένων. Στόχος είναι να βρούμε κάποιους κανόνες που θα μας βοηθήσουν να υπολογίσουμε την εμφάνιση ενός αντικειμένου στη συναλλαγή δεδομένης της ύπαρξης άλλων αντικειμένων.

Για παράδειγμα ας υποθέσουμε ότι βρισκόμαστε σε ένα σούπερ μάρκετ και ότι X = {βούτυρο, ψωμί, τυρί, σοκολάτα} και ότι Y είναι ένα σύνολο από διάφορες συναλλαγές που έχουν γίνει με αυτά τα προϊόντα. Ένα παράδειγμα για έναν κανόνα που μπορεί να εξαχθεί είναι {ψωμί, τυρί} ⇒ {βούτυρο}, πράγμα που σημαίνει πως αν ένας καταναλωτής αγοράσει ψωμί και τυρί, τότε είναι πολύ πιθανόν να αγοράσει και βούτυρο.

Ένας κανόνας για να μπορεί να θεωρηθεί στατιστικά σημαντικός πρέπει να επιβεβαιώνεται από αρκετές εκατοντάδες συναλλαγές και συνήθως οι βάσεις δεδομένων περιέχουν αρκετά εκατομμύρια συναλλαγές.

Η μάθηση με κανόνες συσχέτισης έχει πάρα πολλές εφαρμογές με κυριότερη την εφαρμογή στην προώθηση προϊόντων (marketing) όπου χρειάζεται να βρεθεί ποια προϊόντα αγοράζονται μαζί από τους καταναλωτές.

3.3.3 Δημιουργία Συστάδων

Δημιουργία συστάδων είναι η διαδικασία ομαδοποίησης αντικειμένων κατά την οποία αντικείμενα που μοιάζουν μεταξύ τους σύμφωνα με κάποιο κριτήριο ομαδοποιούνται μαζί σε μία συστάδα σε σχέση με άλλα αντικείμενα που ομαδοποιούνται σε διαφορετικές συστάδες (Berkhin, 2002).

Η δημιουργία συστάδων μπορεί να επιτευχθεί με διάφορους αλγόριθμους που διαφέρουν στο τι συνιστά μια συστάδα και στο κριτήριο κατηγοριοποίησης των συστάδων. Η συνηθέστερη τεχνική είναι αυτή της κατηγοριοποίησης με βάση την ευκλείδια απόσταση μεταξύ δύο σημείων σε ένα δισδιάστατο χώρο. Η τεχνική αυτή μπορεί να γραφεί ως εξής:

Χωρίζουμε τα σημεία σε διαφορετικές συστάδες οι οποίες έχουν τα εξής χαρακτηριστικά:

- Για κάθε δύο σημεία που ανήκουν στην ίδια συστάδα η μεταξύ τους απόσταση είναι πολύ μικρή
- Για κάθε δύο σημεία που ανήκουν σε διαφορετική συστάδα η μεταξύ τους απόσταση είναι πολύ μεγαλή

Παρ’ όλα αυτά υπάρχουν διάφορες άλλες τεχνικές, όπως δημιουργία συστάδων με βάση την πυκνότητα, τον κεντροειδή ή τη μέση κατανομή των σημείων.

Η δημιουργία συστάδων βρίσκει πάρα πολλές εφαρμογές σε διάφορους τομείς, όπως τη βιολογία με την ανάλυση ακολουθιών, την ιατρική με την κατάτμηση εικόνων, το διαδίκτυο με την ανάλυση των κοινωνικών γράφων ή την ομαδοποίηση των αποτελεσμάτων αναζήτησης αλλά και σκοπούς μάρκετινγ με κατηγοριοποίηση των αναγκών των πελατών και το διαχωρισμό τους σε ομάδες καταναλωτών.

3.3.4 Ταξινόμηση

Ταξινόμηση είναι η διαδικασία κατά την οποία προσπαθούμε να κατηγοριοποιήσουμε ένα νέο δεδομένο σε ήδη υπάρχουσες κατηγορίες με βάση ένα σύνολο από δεδομένα των οποίων γνωρίζουμε την κατηγοριοποίηση από πριν (Fisher, 1936). Ένα παράδειγμα ταξινόμησης είναι ο διαχωρισμός της ηλεκτρονικής αλληλογραφίας σε ανεπιθύμητη ή μη από το ίδιο το πρόγραμμα ηλεκτρονικού ταχυδρομείου. Όταν ένα νεό μήνυμα λαμβάνεται, αυτό κατηγοριοποιείται αυτόματα σε ανεπιθύμητο ή μη με βάση την προηγούμενη κατάταξη των μηνυμάτων.

Επομένως, στόχος της ταξινόμησης είναι η κατασκευή ενός μοντέλου έτσι ώστε κάθε νέο δεδομένο που δεν ανήκει στα προϋπάρχοντα δεδομένα να μπορεί να κατηγοριοποιηθεί σε μια κατηγορία/κλάση όσο πιο σωστά γίνεται. Αφού φτιαχτεί το μοντέλο, χρησιμοποιούνται δεδομένα τα οποία έχουν ήδη κατηγοριοποιηθεί, για να δούμε κατά πόσο το μοντέλο που υλοποιήθηκε επιτυχάνει το σκοπό του.

Η ταξινόμηση γενικώς χρησιμοποιείται εκτενώς στην εξόρυξη δεδομένων, αφού μετά την εύρεση χρήσιμων δεδομένων συνήθως θέλουμε να τα κατηγοριοποιήσουμε έτσι ώστε να είναι πιο εύκολο να τα επεξεργαστούμε.

3.3.5 Παλινδρόμηση (Regression)

Η ανάλυση παλινδρόμησης είναι μια στατιστική τεχνική κατά την οποία παρατηρείται η συμπεριφορά μιας εξαρτημένης μεταβλητής σε σχέση με άλλες ανεξάρτητες μεταβλητές. Κάθε φορά μια ανεξάρτητη μεταβλητή μεταβάλλεται, ενω οι άλλες παραμένουν σταθερές και παρατηρούνται οι αλλαγές στη σχετική τιμή της εξαρτημένης μεταβλητής (L. Breiman, 1984).

Η ανάλυση παλινδρόμησης χρησιμοποιείται πάρα πολύ για προβλέψεις και προγνώσεις σε συνεχείς μεταβλητές χρησιμοποιώντας τις προηγούμενες τιμές των μεταβλητών αυτών. Έτσι η παλινδρόμηση αποτελεί σημαντικό μέρος της εξόρυξης δεδομένων που ασχολείται με τις προβλέψεις.

Διάφορες τεχνικές χρησιμοποιούνται στην παλινδρόμηση όπως η γραμμική παλινδρόμηση και η παλινδρόμηση ελαχίστων τετραγώνων.

3.3.6 Παραγωγή Σύνοψης (Summarization)

Η αυτόματη ανακεφαλαιοποίηση είναι η διαδικασία της δημιουργίας της περίληψης ενός κειμένου, η οποία διατηρεί όλες τις σημαντικές πληροφορίες του κειμένου αυτού και έχει πολύ μικρότερη έκταση από το αρχικό (Rada Mihalcea, 2004). Η ανακεφαλαιοποίηση παίζει πολύ σημαντικό ρόλο σήμερα όπου ο όγκος των δεδομένων έχει αυξηθεί σημαντικά αφού δεν είναι δυνατή πλέον η προσπέλαση όλων των δεδομένων σε ικανοποιητικό χρόνο. Έτσι χρειαζόμαστε μια περίληψη των δεδομένων διατηρώντας όμως όλες τις σημαντικές πληροφορίες.

Υπάρχουν δύο σημαντικές τεχνικές για την ανακεφαλαιοποίηση: η εξόρυξη και η άντληση. Κατά την εξόρυξη επιλέγεται ένα μέρος από τις λέξεις, φράσεις ή και προτάσεις του αρχικού κειμένου και σχηματίζεται η περίληψη, ενώ αντίθετα κατά την άντληση δημιουργείται μια εσωτερική σημασιολογική αναπαράσταση του κειμένου και μετά χρησιμοποιούνται τεχνικές παραγωγής φυσικής γλώσσας έτσι ώστε να δημιουργηθεί η περίληψη η οποία είναι πιο κοντά στην περίληψη που θα δημιουργούσε ένας άνθρωπος. Κατά την τεχνική αυτή μπορεί να χρησιμοποιηθούν λέξεις και φράσεις οι οποίες δεν αναφέρονται στο αρχικό κείμενο. Η τεχνική αυτή όμως είναι σχετικά νέα.

3.3.7 Εξόρυξη διαδοχικών μοτίβων

Η εξόρυξη διαδοχικών μοτίβων ασχολείται με την εύρεση ενός συνόλου δεδομένων που εμφανίζονται μαζί σε ορισμένες αλληλουχίες. Η τεχνική αυτή εφαρμόζεται σε βάσεις ακολουθιών και σε βάσεις δεδομένων συναλλαγών έτσι ώστε να βρεθούν αυτές οι υποακολουθίες (Fayyad Usama, 1996).

Οι τεχνικές που χρησιμοποιούνται συνήθως συνδυάζουν αποτελεσματικές τεχνικές κλαδέματος (pruning) και ιεράρχισης (indexing) ακόμα και για πολύ μεγάλα μοτίβα ακολουθιών. Επίσης χρησιμοποιείται και η τεχνική LVP (Load Value Prediction) έτσι ώστε να βρεθούν συχνά μοτίβα στις προσβάσεις στη μνήμη ενός προγράμματος.

Επίσης σε ορισμένους αλγόριθμους αρχικά χρησιμοποιείται η τεχνική κατά πλάτος έτσι ώστε να ανιχνευθούν πιθανές ακολουθίες και αργότερα χρησιμοποιούνται οι τεχνικές κλαδέματος για να περιορίσουν τις ακολουθίες.

3.4 Εφαρμογές Εξόρυξης Δεδομένων

Στο σημείο αυτό θα γίνει μια σύντομη παρουσίαση των διαφόρων τομέων στους οποίους μπορεί να εφαρμοστεί η εξόρυξη δεδομένων. Οι τομείς αυτοί διαφέρουν αρκετά μεταξύ τους και μπορούν να επεκταθούν σχεδόν σε ολόκληρο το επιστημονικό φάσμα.

3.4.1 Παγκόσμιος Ιστός

Με τον ολοένα αυξανόμενο όγκο διαθέσιμης πληροφορίας στον παγκόσμιο ιστό, δημιουργήθηκε η ανάγκη για γρήγορη και αποτελεσματική αναζήτηση στην πλειάδα των διαθέσιμων πληροφοριών. Τη λύση στο πρόβλημα αυτό ήρθαν να δώσουν οι διάφορες μηχανές αναζήτησης, οι οποίες χρησιμοποιώντας τεχνικές εξόρυξης δεδομένων αναλύουν σε χρόνο λιγότερο του ενός δευτερολέπτου, δισεκατομμύρια από σελίδες και παραδίδουν σε πραγματικό χρόνο τα καλύτερα δυνατά αποτελέσματα στην αναζήτηση του χρήστη. Μηχανές αναζήτησης όπως η Google, το Bing και η Yahoo αναλύουν καθημερινά τεράστιους όγκους δεδομένων και για να το επιτύχουν αυτό χρησιμοποιούν τεχνικές εξόρυξης δεδομένων αφού δεν είναι δυνατόν να προσπελαστεί κάθε νέα πληροφορία που μπαίνει στον παγκόσμιο ιστό. Έτσι η ανάγκη για γρήγορη αλλά και αποτελεσματική αναζήτηση στον παγκόσμιο ιστό έδωσε μεγάλη ώθηση στην ανάπτυξη των τεχνικών εξόρυξης δεδομένων, μια ανάπτυξη η οποία συνεχίζεται με ραγδαίους ρυθμούς μέχρι και σήμερα μιας και η αναζήτηση πληροφορίας δεν περιορίζεται μόνο στις ιστοσελίδες αλλά και σε βίντεο, εικόνες, ειδησεογραφικά άρθρα αλλά και πιο πρόσφατα στους κοινωνικούς γράφους και στα κοινωνικά δίκτυα. Σήμερα η έρευνα γύρω από την αναζήτηση στον παγκόσμιο ιστό με τεχνικές εξύρυξης δεδομένων επικεντρώνεται κυριώς στην εύρεση

πληροφορίας πραγματικού χρόνου, δηλαδή πληροφορίας που μόλις έχει παραχθεί όπως έκτακτα γεγονότα, από πολλές και διαφορετικές μεταξύ τους πηγές.

3.4.2 Επιστήμες και μηχανική

Η εξόρυξη δεδομένων τα τελευταία χρόνια βρίσκει πολλές εφαρμογές στις διάφορες επιστήμες και τη μηχανική, όπως τη βιοπληροφορική και τη βιοϊατρική, τη γενετική, την ιατρική, την εκπαίδευση αλλά και τη μηχανική ηλεκτρικής ισχύος.

Στον τομέα της μηχανικής ηλεκτρικής ισχύος, τεχνικές εξόρυξης δεδομένων έχουν χρησιμοποιηθεί για την παρακολούθηση εξοπλισμού υψηλής ηλεκτρικής ισχύος. Σκοπός είναι να εξαχθούν χρήσιμες πληροφορίες για το δίκτυο αλλά και για την κατάσταση διαφόρων παραμέτρων του δικτύου και του εξοπλισμού που παίζουν σημαντικό ρόλο στην ασφάλεια των διαφόρων εγκαταστάσεων.

Στη βιοπληροφορική και τη γενετική τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται για να αναλυθούν και να κατανοηθούν οι ακολουθίες του ανθρώπινου DNA (Zhu Xingquan, 2007). Μελετάται πώς οι διάφορες αλλαγές στις ακολουθίες του ανθρώπινου DNA, συνδέονται με την εμφάνιση διαφόρων ασθενειών όπως ο καρκίνος, πράγμα που βοηθάει στη μελέτη και τη διάγνωση των ασθενειών αυτών. Η κυριότερη τεχνική εξόρυξης δεδομένων που χρησιμοποιείται στον τομέα αυτόν είναι η πολυπαραγοντική μείωση διάστασης (multifactor dimensionality reduction).

Όσον αφορά στην εκπαίδευση τεχνικές εξόρυξης δεδομένων έχουν χρησιμοποιηθεί στα σχολεία έτσι ώστε να κατανοηθούν οι διάφοροι παράγοντες που αποσπούν τους μαθητές από τα μαθήματά τους αλλά και στα πανεπιστήμια για το ποιοι παράγοντες ωθούν έναν φοιτητή να συνεχίσει ή να διακόψει τις σπουδές του (Ryan, 2007).

Στον τομεά της ιατρικής τεχνικές εξόρυξης δεδομένων έχουν χρησιμοποιηθεί για την εξόρυξη χρήσιμων πληροφοριών σε ηλεκτρονικά αρχεία υγείας για διάφορους σκοπούς όπως τη σύνδεση της συνταγογράφησης διαφόρων φαρμάκων με ορισμένες ασθένειες. Επίσης αντίστοιχες τεχνικές έχουν χρησιμοποιηθεί από τον παγκόσμιο οργανισμό υγείας (WHO) για την εύρεση μοτίβων σε αναδυόμενα θεμάτα ασφάλειας στη χρήση φαρμάκων από μια βάση δεδομένων με εικαζόμενα περιστατικά παρενεργειών από φάρμακα.

Επίσης τεχνικές εξόρυξης δεδομένων έχουν χρησιμοποιηθεί και στην τεχνολογία λογισμικού (software engineering). Η τεχνική MSR (mining software repositories) αναλύει τα πλούσια δεδομένα που υπάρχουν στα αποθετήρια λογισμικού, προκειμένου να εξαχθούν ενδιαφέρουσες και χρήσιμες πληροφορίες οι οποίες αφορούν στην εξέλιξη αλλά και στη συντήρηση αυτών των συστημάτων λογισμικού.

3.4.3 Επιχειρήσεις

Τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται συνεχώς από μεγάλες κυρίως επιχειρήσεις για την ανάλυση και την εξαγωγή χρήσιμων πληροφοριών από τον ολοένα και αυξανόμενο όγκο δεδομένων που συλλέγουν για τους πελάτες τους. Μέσα από τη σωστή ανάλυση αυτών των δεδομένων οι επιχειρήσεις μπορούν να έχουν σημαντικά ωφέλη αφού μπορούν να αναλύσουν τη συμπεριφορά των πελατών τους και να προσαρμόσουν ανάλογα τις καμπάνιες τους και τον τρόπο που επικοινωνούν με τους πελάτες τους. Παραδείγματος χάριν μια μεγάλη αλυσίδα υπεραγορών συλλέγει καθημερινά εκατομμύρια συναλλαγές στα ταμεία της. Αυτά τα δεδομένα φυλάσσονται σε μια κεντρική βάση δεδομένων. Λόγω του όγκου τους δεν μπορούν να προσπελαστούν όλα τα δεδομένα ένα προς ένα, όμως με τη χρήση κατάλληλων τεχνικών εξόρυξης μπορούν να βρεθούν μοτίβα στις αγορές των πελατών, γεγονός που μπορεί να βοηθήσει την εταιρεία να εισάγει καλύτερες προσφορές, να συνδυάσει προϊόντα και γενικότερα να προωθήσει καλύτερα τις επιχειρηματικές της δραστηριότητες (O'Brien J. A., 2011). Ένα άλλο παράδειγμα είναι οι συναλλαγές που γίνονται μέσω πιστωτικών καρτών. Οι συναλλαγές αυτές αποθηκεύονται σε βάσεις δεδομένων και η ανάλυσή τους μπορεί να εξάγει χρήσιμες πληροφορίες για την καταναλωτική συμπεριφορά των πολιτών. Φυσικά σε όλες αυτές τις περιπτώσεις υπάρχει το ζήτημα του απόρρητου των προσωπικών δεδομένων και το τι μπορεί να γίνει αν όλα αυτά τα δεδομένα καταλήξουν σε λάθος χέρια. Γι’ αυτό και στις περισσότερες περιπτώσεις τα δεδομένα που φυλάσσονται είναι ανώνυμα και πολλές φορές κρυπτογραφημένα.

Τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται επίσης ευρέως στη διαχείριση των πελατειακών σχέσεων. Με την ανάλυση των συναλλαγών και των προτιμήσεων των πελατών οι εταιρείες μπορούν να κατατάξουν τους πελάτες τους σε ομάδες σύμφωνα με διάφορα κριτήρια (χρησιμοποιώντας και τεχνικές ομαδοποίησης συστάδων) αξιοποιώντας καλύτερα τους διαθέσιμους πόρους για διαφήμιση και προώθηση αφού οι διαφημίσεις θα είναι πιο προσωπικές και ειδικευμένες για κάθε ομάδα πελατών. Παραδείγματος χάριν αφού οι πελάτες κατηγοριοποιηθούν σε ομάδες διαφημιστικά emails με συγκεκριμένες προσφορές θα σταλούν μόνο στις ομάδες των πελάτων που είναι πιθανότερο να ανταποκριθούν, γεγονός που μπορεί να αυξήσει τα έσοδα ενώ ταυτόχρονα η εταιρεία δεν ενοχλεί τους υπόλοιπους πελάτες της με ανεπιθύμητη ηλεκτρονική αλληλογραφία. Επίσης οι εταιρείες με τεχνικές εξόρυξης δεδομένων μπορούν να βρουν τους πιο πιστούς τους πελάτες, οι οποίοι είναι πιθανότερο να κάνουν και μελλοντικές αγορές και να στείλουν μόνο σε αυτούς προσφορές, ενισχύοντας την πίστη τους στην εταιρεία και εξασφαλίζοντας την προτίμησή τους.

Επίσης η εξόρυξη δεδομένων είναι χρήσιμη στις μεγάλες εταιρείες στη διαχείριση του προσωπικού. Με διάφορες τεχνικές μια εταιρεία μπορεί να βρει τους καλύτερούς της υπαλλήλους, οι οποίοι είναι πιο παραγωγικοί από τους άλλους και να τους προσφέρει προνόμια ή και αυξήσεις στους μισθούς. Επίσης μπορεί να βρει τα συνηθέστερα πανεπιστήμια στα οποία έχουν φοιτήσει οι καλύτεροί της υπαλλήλοι και να επικεντρώσει τις προσπάθειες πρόσληψης στους αποφοίτους των συγκεκριμένων πανεπιστημίων (Monk Ellen, 2006).

3.4.4 Επενδύσεις

Τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται και στον κόσμο των επενδύσεων βοηθώντας τους επενδυτές αλλά και τους αναλυτές να βρουν τις επιχειρήσεις που φέρουν το μεγαλύτερο κέρδος. Αλγόριθμοι εξόρυξης δεδομένων εξάγουν πληροφορίες από διάφορα κείμενα (ειδησεογραφικά άρθρα, οικονομικές και επιχειρηματικές αναλύσεις) έτσι ώστε να προβλέψουν καθημερινώς αν η μετοχή μιας εταιρείας θα ανέβει ή θα κατέβει. Επίσης ένα μεγάλο μέρος της έρευνας διεξάγεται στον τομέα του temporal data mining στις χρηματοπιστωτικές εφαρμογές (B. Wuthrich, 1998).

Κεφάλαιο 4: Αναπτυχθείσα μεθοδολογία Άμεσης Αναφοράς για τον εντοπισμό τάσεων

Ο εντοπισμός τάσεων είναι όπως περιγράφηκε παραπάνω ένας σχετικά καινούργιος τομέας που ασχολείται με τον αυτόματο τρόπο εντοπισμού τάσεων μέσα από συλλογές κειμένων και επιστημονικών άρθρων. Λόγω όμως του γεγονότος ότι ακόμα είναι πρόσφατος τομέας ενδιαφέροντος για τους ερευνητές δεν έχουν ακόμη αναπτυχθεί ολοκληρωμένα συστήματα εντοπισμού τάσεων που να ανταποκρίνονται στις σημερινές απαιτήσεις. Πολλές και διάφορες μεθοδολογίες έχουν προταθεί για το πρόβλημα του εντοπισμού τάσεων - καμία όμως δε λύνει το πρόβλημα συνολικά. Στο πλαίσιο αυτής της διπλωματικής εργασίας θα ασχοληθούμε με τη μεθοδολογία της άμεσης αναφοράς για τον εντοπισμό τάσεων. Μια μεθοδολογία που ασχολείται με τις σχέσεις μεταξύ των βιβλιογραφικών αναφορών μεταξύ των διαφόρων άρθρων που αναφέρονται στο ίδιο θέμα.

4.1 Περιγραφή εργασιών που έχουν ήδη γίνει

Έχει παρατηρηθεί πως συνήθως τα άρθρα τα οποία ασχολούνται με ένα παρόμοιο θέμα συχνά αναφέρουν το ένα το άλλο και επομένως υπάρχει μια συσχέτιση μεταξύ τους, ενώ τα άρθρα που ασχολούνται με διαφορετικά θέματα συνήθως δεν αναφέρουν το ένα το άλλο και επομένως δε συσχετίζονται άμεσα. Έτσι παρατηρώντας τις βιβλιογραφικές αναφορές των άρθρων προς μελέτη μπορούμε να εντοπίσουμε τα άρθρα τα οποία συσχετίζονται μεταξύ τους και μέσω της ανάλυσης αυτών των συσχετισμένων άρθρων να εντοπίσουμε διάφορες τάσεις.

Η πρώτη αναφορά σε ένα μέτωπο έρευνας, δηλαδή ένα πεδίο έρευνας όπου βρίσκεται στα πρώτα στάδια ανάπτυξής του και όπου τα διάφορα άρθα αναφέρουν το ένα το άλλο, έγινε από τον de Solla Price το 1965 στο κλασσικό του άρθρο. Σύμφωνα με τον Price υπάρχει μια τάση από τους επιστήμονες και ερευνητές να αναφέρουν συνήθως τα πιο πρόσφατα δημοσιευμένα άρθρα. Επομένως ένα μέτωπο έρευνας βασίζεται κυρίως σε πρόσφατα δημοσιοποιημένη εργασία και έτσι δημιουργείται ένα αρκετά πυκνό δίκτυο από άρθρα τα οποία σχετίζονται και αναφέρουν το ένα το άλλο. Επομένως το μέτωπο έρευνας αποτελείται από τα άρθρα τα οποία οι ίδιοι οι επιστήμονες θεωρούν σημαντικά και συμπεριλαμβάνουν στις βιβλιογραφικές αναφορές των δικών τους άρθρων.

Οι ερευνητές έχουν μελετήσει και προτείνει διάφορες ποσοτικές μεθόδους που μπορούν να χρησιμοποιηθούν για την παρακολούθηση ενός μετώπου έρευνας καθώς εξελίσσεται στον χρόνο και επομένως και για τον εντοπισμό ανερχόμενων τάσεων στην έρευνα. Στην βιβλιογραφία αναφέρονται κυρίως τρεις μεθόδοι που ακολουθούνται (Naoki Shibata, Comparative Study on Methods of Detecting Research, 2009): η άμεση αναφορά (με την οποία θα ασχοληθούμε) που ορίζεται ως η ακμή μεταξύ δύο άρθρων που το ένα αναφέρει το άλλο, η συν-αναφορά που ορίζεται ως η ακμή μεταξύ δύο άρθρων-κόμβων που αναφέρονται από τα ίδια άρθρα και η βιβλιογραφική σύζευξη που ορίζεται ως η ακμή μεταξύ δύο άρθρων που αναφέρονται στα ίδια άρθρα.

Οι Small και Griffith το 1974 εντοπίζουν ένα μέτωπο έρευνας σε επιστημονικές ειδικότητες ως συστάδες από διάφορα άρθρα τα οποία συν-αναφέρονται. Οι Braam, Moed και van Raan το 1991 ανέλυσαν τα θέματα των διάφορων συστάδων από άρθρα τα οποία συν-αναφέρονται με βάση τη συχνότητα των όρων που χρησιμοποιούνται για την ταξινόμηση των άρθρων (tags). Ο Small το 2006 παρουσίασε μια μέθοδο εντοπισμού και πρόβλεψης ανερχόμενων πεδίων έρευνας αναλύοντας τις συν-αναφορές σε δίκτυα άρθρων που προέρχονται από το το 1% των άρθρων με τις περισσότερες αναφορές (Small, 2006). Ο Schiminovich το 1971 κατέταξε τις ακαδημαϊκές εκδόσεις αυτόματα χρησιμοποιώντας αναδρομική βιβλιογραφική σύζευξη (Schiminovich, 1971). Ο Rousseau το 2002 εξήγαγε σημαντικές υποδομές σε δίκτυα αναφορών που κατασκευάστηκαν με συν-αναφορά και βιβλιογραφική σύζευξη (Fang, 2001) (Egghe, 2002). Ο Garfield το 2004 δημιούργησε ένα χάρτη ιστογραμμάτων της γνώσης στα δίκτυα άμεσων αναφορών (Garfield, 2004). Οι Shibata Kajikawa, Takeda, και Matsushima το 2008 πρότειναν μια μέθοδο ανίχνευσης ανερχόμενων τάσεων με ανάλυση δικτύων άμεσων αναφορών (Shibata, 2008)

Παρ΄όλα αυτά υπάρχει εμφανώς λιγότερη έρευνα στο κομμάτι της αξιολόγησης και των τριών τρόπων αναφορών και στην αποτελεσματικότητά τους στο να βρίσκουν τάσεις. Οι Klavans και Boyack το 2006 σύγκριναν την απόδοση των συστάδων σε δίκτυα αναφορών που δημιουργήθηκαν με άμεση αναφορά και συν-αναφορά. Τα αποτελέσματά τους έδειξαν πως οι συστάδες στο δίκτυο που δημιουργήθηκε με άμεση αναφορά παρουσίαζαν μεγαλύτερη ομοιότητα (Klavans, 2006). Επίσης οι Naoki, Yuya, Yoshiyuki, Katsumori το 2008 σύγκριναν και τις τρεις μεθοδολογίες χρησιμοποιώντας διάφορες μετρικές όπως το μέγεθος συστάδων και τη μέση χρονολογία δημοσίευσης (Naoki Shibata, Comparative Study on Methods of Detecting Research, 2009).

Επομένως από τις διάφορες μελέτες που έχουν γίνει κατά τις οποίες συγκρίνονται οι τρεις τρόποι με τους οποίους μπορούν να κατασκευαστούν τα δίκτυα αναφορών συμπεραίνουμε πως την καλύτερη απόδοση όσον αφορά τον εντοπισμό ανερχόμενων τάσεων παρουσιάζει η άμεση αναφορά αφού μπορεί να εντοπίσει γρηγορότερα και με μεγαλύτερη ακρίβεια τις διάφορες τάσεις.

4.2 Γενική περιγραφή ακολουθηθείσας μεθοδολογίας

Για την εφαρμογή μας η οποία αφορά στον εντοπισμό των τάσεων σε επιστημονικά άρθρα και δημοσιεύσεις, μετά από μια ανασκόπηση των διαθέσιμων μεθοδολογιών και τεχνολογιών, επιλέξαμε οι νέες τάσεις να εντοπίζονται χρησιμοποιώντας δίκτυα αναφορών. Με μια σύντομη έρευνα στη βιβλιογραφία βλέπουμε πως ο καλύτερος τρόπος για τη δημιουργία ενός τέτοιου δικτύου είναι μέσω της άμεσης αναφοράς αφού αποδίδει καλύτερα και με περισσότερη ακρίβεια.

4.2.1 Γενική ιδέα

Η βάση της αναπτυχθείσας μεθοδολογίας αφορά στην κατασκευή ενός γράφου ο οποίος περιέχει όλα τα διαθέσιμα επιστημονικά άρθρα ενός ερευνητικού τομέα λαμβάνοντας υπ’όψιν τις αναφορές μεταξύ αυτών και στην ανεύρεση σε αυτόν ομάδων ισχυρά συσχετιζόμενων δημοσιεύσεων. Συγκεκριμένα, κάθε άρθρο αντιπροσωπεύει και έναν κόμβο και κάθε αναφορά σε ένα άλλο άρθρο μια ακμή προς τον άρθρο-κόμβο αυτό. Στο γράφο αυτό βρίσκουμε τις ισχυρά συνεκτικές συνιστώσες (strongly connected components). Η βασική ιδέα είναι ότι τα άρθρα που ασχολούνται με παρόμοια θέματα αναφέρονται το ένα στο άλλο στη βιβλιογραφία και επομένως μπορούν να είναι μέρος μιας ισχυρά συνεκτικής συνιστώσας, ενώ τα άρθρα που δε σχετίζονται μεταξύ τους δεν είναι ισχυρά συνδεδεμένα αφού δεν αναφέρουν το ένα το άλλο στη βιβλιογραφία. Κάθε ισχυρά συνεκτική συνιστώσα μπορεί να θεωρηθεί πως αποτελεί ένα γενικό πεδίο έρευνας το οποίο συνιστά μια τάση – είτε παλαιότερη είτε σύγχρονη. Αυτό μπορεί να επιβεβαιβεωθεί κοιτάζοντας τα άρθρα-κόμβους που αποτελούν την κάθε ισχυρά συνεκτική συνιστώσα από κάποιον ειδικό. Αν θέλουμε ακόμα μεγαλύτερη ειδίκευση στα ανερχόμενα θέματα μέσα

στο γενικό ανερχόμενο πεδίο έρευνας παίρνουμε την κάθε ισχυρά συνεκτική συνιστώσα ξεχωριστά και χωρίζουμε τα άρθρα-κόμβους της κάθε συνιστώσας σε συστάδες (clusters). Κάθε συστάδα θα αντιπροσωπεύει και ένα πιθανό ανερχόμενο θέμα.

4.2.2 Γενική Μεθοδολογία

Βήμα 1^{ο}:

Αναζητούμε στα επιστημονικά άρθρα με βάση κάποια λέξη κλεδί (keyword), όπως ”breast cancer”(καρκίνος του μαστού) και λαμβάνουμε έναν αριθμό άρθρων σχετικών με τη λέξη-κλειδί που επιλέξαμε.

Βήμα 2^{ο}:

Κατασκευάζουμε ένα δίκτυο αναφορών (citation network) με βάση την αναφερόμενη βιβλιογραφία που στο τέλος κάθε επιστημονικού άρθρου. Θεωρούμε κάθε άρθρο ως ένα κόμβο και τη μεταξύ τους αναφορά ως μια ακμή. Αναλύοντας τις βιβλιογραφικές αναφορές του κάθε άρθρου θα δημιουργείται ένας κόμβος για το άρθρο αυτό και αντίστοιχα άλλοι τόσοι κόμβοι για τα άρθρα τα οποία αναφέρει στη βιβλιογραφία. Ακολούθως ο αρχικός κόμβος θα ενώνεται με ακμές που ξεκινούν από τον κόμβο αυτόν και καταλήγουν στους κόμβους-άρθρα της βιβλιογραφίας. Αν ένας κόμβος-άρθρο έχει προκύψει προηγουμένως από την ανάλυση κάποιου άλλου άρθρου, ο κόμβος αυτός δεν ξαναδημιουργείται αλλά χρησιμοποιείται ο ίδιος.

Βήμα 3^{ο}:

Στο δίκτυο αναφορών που σχηματίσαμε βρίσκουμε τις ισχυρά συνεκτικές συνιστώσες. Τις μικρότερες ισχυρά συνεκτικές συνιστώσες τις απορρίπτουμε αφού δεν περιέχουν επαρκή αριθμό άρθρων, για να μπορέσουμε να βρούμε κάποια ανερχόμενη τάση.

Βήμα 4^{ο}:

Αν θέλουμε περισσότερη λεπτομέρεια χωρίζουμε τις μεγαλύτερες ισχυρά συνεκτικές συνιστώσες σε συστάδες.

Βήμα 5^{ο}:

Τέλος, αξιολογούμε τα άρθρα κάθε συστάδας ή ισχυρά συνεκτικής συνιστώσας σύμφωνα με κάποιες μετρικές όπως το visibility που ορίζεται ως το κανονικοποιημένο μέγεθος της συστάδας, την ταχύτητα που ορίζεται ως η μέση χρονολογία δημοσίευσης των άρθρων της συστάδας και την τοπολογική σημασία που ορίζεται ως η πυκνότητα της κάθε συστάδας. Επίσης μπορούμε να ορίσουμε και άλλες μετρικές όπως τον αριθμό των συγγραφέων που έχουν άρθρα σε κάθε συστάδα αλλά και το κατά πόσο είναι σημαντικοί οι συγγραφείς αυτοί στον τομέα τους.

4.3 Περιγραφή μεθοδολογίας που ακολουθήθηκε

Για την ανάπτυξη των μηχανισμών εύρεσης τάσεων ακολουθήθηκε η παραπάνω μεθοδολογία χωρίς όμως να δοθεί έμφαση στα βήματα 4 και 5. Ο περαιτέρω διαχωρισμός των άρθρων σε συστάδες αλλά και η αξιολόγηση της μεθόδου με τη χρήση διαφόρων μετρικών αφέθηκαν για μελλοντική έρευνα.

4.3.1 Επιλογή και Χρήση Συνόλου Δεδομένων - Αναζήτηση σε επιστημονικές βάσεις

Για να πάρουμε τα άρθρα που μας ενδιαφέρουν χρησιμοποιήθηκε η βάση δεδομένων για επιστημονικά άρθρα PLOS ONE (PLOS ONE: accelerating the publication of peer-reviewed science). Η PLOS ONE είναι μια βάση δεδομένων επιστημονικών άρθρων που είναι ανοικτή στο κοινό και δημοσιεύεται από το Public Library of Science, ένα μη κερδοφόρο οργανισμό που έχει ως σκοπό του τη δημιουργία μιας βιβλιοθήκης με επιστημονικά άρθρα τα οποία θα είναι διαθέσιμα σε όλους δωρεάν (PLOS). Η PLOS ONE εκδίδεται από το 2006 και καλύπτει την έρευνα από κάθε τομέα των επιστημών και της ιατρικής. Πριν δημοσιευθούν τα άρθρα, περνούν μέσα από μια εσωτερική και εξωτερική αξιολόγηση από ομότιμους επιστήμονες και την κοινότητα του PLOS ONE. Τα άρθρα προς αξιολόγηση δεν αποκλείονται βάσει του περιεχόμενού τους ή της σημαντικότητάς τους και της συνεισφοράς τους σε ένα επιστημονικό πεδίο. Η online έκδοση της PLOS ONE χρησιμοποιεί την πολιτική του πρώτα δημοσίευσε και μετά αξιολόγησε, δηλαδή τα άρθρα πρώτα δημοσιεύονται και μετά αξιολογούνται από την κοινότητα του PLOS ONE με διάφορα εργαλεία αξιολόγησης (PLOS ONE).

Εικόνα 7: Το λογότυπο της PLOS ONE

Η PLOS ONE δίνει τη δυνατότητα στο χρήστη για αναζήτηση στη βάση δεδομένων της με βάση κάποιες λέξεις-κλειδιά. Αφού γίνει η αναζήτηση με κάποια λέξη-κλειδί, ακολούθως η PLOS ONE παρουσιάζει όλα τα άρθρα τα οποία έχουν κάποια σχέση με τη λέξη κλειδί αυτή. Το κάθε άρθρο είναι διαθέσιμο δωρεάν μέσω της ιστοσελίδας για όλους τους χρήστες. Επίσης το κάθε άρθρο μπορεί να αποκτηθεί σε τρεις διαφορετικές μορφές: PDF για εύκολη ανάγνωση σε υπολογιστή, Citation για ανάλυση από προγράμματα αναφορών και XML για εύκολη προσπέλαση των περιεχομένων του από υπολογιστή. Εμείς κατεβάζαμε το κάθε αρχείο σε μορφή XML για να μπορούμε εύκολα και με αποδοτικό τρόπο να βρούμε τις βιβλιογραφικές αναφορές του κάθε αρχείου.

4.3.2 Ανάκτηση δεδομένων από τα αρχεία XML

Αφού κατεβάσαμε τα αρχεία που μας ενδιαφέρουν σε μορφή XML, ακολούθως πρέπει για το κάθε αρχείο να βρούμε ένα τρόπο να εξάγουμε τις βιβλιογραφικές του αναφορές. Για τη διάτρεξη του κάθε αρχείου XML χρησιμοποιήσαμε τον SAX Parser (Simple API for XML) που προσφέρει απλούς μηχανισμούς, για να διαβάζει δεδομένα από ένα XML αρχείο. Έτσι διατρέχουμε το κάθε αρχείο μέχρι να βρούμε τις βιβλιογραφικές αναφορές του. Ακολούθως διαβάζουμε με τον SAX Parser μια προς μια τις αναφορές αυτές και ανακτούμε τον τίτλο της κάθε αναφοράς. Αυτό γίνεται για όλα τα αρχεία XML.

4.3.3 Εύρεση PubMed ID για κάθε βιβλιογραφική αναφορά

Αφού εξάγουμε από το κάθε αρχείο XML τους τίτλους των άρθρων τα οποία αναφέρει στη βιβλιογραφία, αναζητούμε στην PubMed το ID (αναγνωριστικό) της κάθε αναφοράς ώστε να καθορίζουμε μονοσήμαντα το κάθε άρθρο και να μπορούμε να βρούμε περισσότερες πληροφορίες γι' αυτό.

Εικόνα 8: Το λογότυπο της PubMed

Η PubMed είναι μια δωρεάν βάση δεδομένων η οποία έχει πρόσβαση στη βάση δεδομένων MEDLINE που περιέχει τις αναφορές και τις περιλήψεις (abstracts) άρθρων για τις βιοεπιστήμες και τη βιοϊατρική (PubMed - NCBI). Η PubMed διαχειρίζεται από την Εθνική Βιβλιοθήκη της Ιατρικής των Ηνωμένων Πολιτειών Αμερικής (US National Library of Medicine – NLM) στα Εθνικά Ινστιτούτα Υγείας των ΗΠΑ (National Institutes of Health) ως μέρος του συστήματος Entrez για εξαγωγή δεδομένων (Lindeberg, 2000).

Κάθε άρθρο στην PubMed αντιστοιχίζεται σε ένα μοναδικό για το άρθρο αυτό αναγνωριστικό (ID) που αποτελείται από οχτώ αριθμούς. Έτσι έχοντας το PubMed ID (PMID) του κάθε άρθρου μπορούμε να ορίσουμε μονοσήμαντα το κάθε άρθρο.

Για την εύρεση του PMID της κάθε βιβλιογραφικής αναφοράς αλλά και του ίδιου του άρθρου χρησιμοποιούμε το ESearch που είναι το API της PubMed για αναζήτηση στη βάση δεδομένων. Η αναζήτηση γίνεται με τον τίτλο του κάθε άρθρου-αναφορά και παίρνουμε πίσω ένα άλλο XML αρχείο που περιέχει το PMID του αρχείου που ζητήσαμε. Ακολούθως χρησιμοποιούμε και πάλι τον SAX Parser έτσι ώστε να ανακτήσουμε τον αναγνωριστικό αριθμό PMID από το XML αρχείο.

Η διαδικασία αυτή γίνεται για όλες τις αναφορές κάθε αρχείου. Ακολούθως το κάθε PMID αποθηκεύεται σε ένα text αρχείο για περαιτέρω επεξεργασία αργότερα, σε ένα φάκελο. Έτσι ο φάκελος αυτός περιέχει ένα text αρχείο που περιλαμβάνει όλες τις αναφορές του κάθε άρθρου (αποθηκεύεονται μόνο τα PMIDs των αναφορών).

4.3.4 Δημιουργία Δικτύου Αναφορών

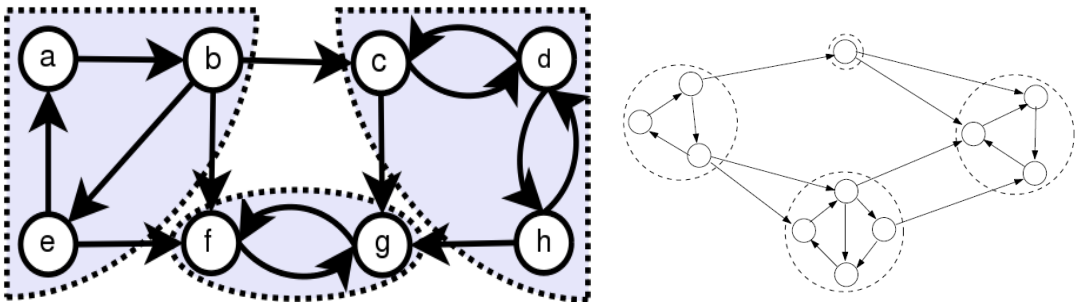
Αφού πλέον έχουμε για κάθε άρθρο και κάθε βιβλιογραφική αναφορά ενα μοναδικό αναγνωριστικό (το PMID) μπορούμε να κατασκευάσουμε το δίκτυο αναφορών. Εμείς θα ασχολήθουμε όπως έχουμε αναφέρει με τα δίκτυα άμεσων αναφορών. Επομένως κάθε άρθρο θα αποτελεί και ένα κόμβο και θα συνδέεται με ακμές που ξεκινούν στο άρθρο-κόμβο αυτό και καταλήγουν στους κόμβους-άρθρα όλων των αναφορών του. Το όνομα του κάθε κόμβου θα είναι το PMID του κάθε άρθρου-κόμβου. Εφαρμόζοντας την ανωτέρω λογική σε όλα τα αρχεία τα οποία παίρνουμε από την αρχική μας αναζήτηση δημιουργούμε ένα κατευθυνόμενο γράφο που αποτελεί και το δίκτυο αναφορών μας. Για τη δημιουργία του γράφου διαβάζουμε ένα ένα τα αρχεία text που περιέχουν τα PMIDs και δημιουργούμε το γράφο χρησιμοποιώντας βασικές τεχνικές δημιουργίας κατευθυνόμενων γράφων.

4.3.5 Εύρεση ισχυρά συνεκτικών συνιστωσών

Το επόμενο βήμα αφού κατασκευάσουμε το δίκτυο αναφορών είναι να βρούμε τις μεγαλύτερες ισχυρά συνεκτικές συνιστώσες. Κάθε ισχυρά συνεκτική συνιστώσα μπορεί να θεωρηθεί πως αποτελεί ένα γενικό πεδίο έρευνας το οποίο είναι δυνατόν να είναι ανερχόμενο. Τις μικρότερες συνεκτικές συνιστώσες τις αγνοούμε.

Ένας κατευθυνόμενος γράφος λέγεται ισχυρά συνδεδεμένος αν υπάρχει ένα μονοπάτι από κάθε κορυφή του γράφου προς κάθε άλλη κορυφή. Αυτό σημαίνει πως υπάρχουν μονοπάτια προς κάθε κατεύθυνση του γράφου, δηλαδή υπάρχει ένα μονοπάτι από το A στο B, αλλά και ένα μονοπάτι από το B στο A.

Οι ισχυρά συνεκτικές συνιστώσες ενός κατευθυνόμενου γράφου είναι οι μέγιστοι ισχυρά συνδεδεμένοι υπογράφοι του (Thomas H. Cormen, 2001).



Εικόνα 9: Κατευθυνόμενοι γράφοι με σημειωμένες τις ισχυρά συνεκτικές συνιστώσες τους

Για την εύρεση των ισχυρά συνεκτικών συνιστωσών στο δίκτυο αναφορών χρησιμοποιούμε τον αλγόριθμο του Kosaraju (Roughgarden, 2012).

4.4 Υλοποίηση Μηχανισμών

Για την υλοποίηση του προγράμματος για τον εντοπισμό των τάσεων ακολουθήθηκε η παραπάνω μεθοδολογία. Παρακάτω περιγράφονται και αναλύονται οι διάφοροι αλγόριθμοι που χρησιμοποιήθηκαν, οι εφαρμοσθείσες τεχνολογίες καθώς και λεπτομέρειες της υλοποίησης.

4.4.1.Αλγόριθμος Kosaraju για την εύρεση ισχυρά συνεκτικών συνιστωσών

Για την εύρεση των ισχυρά συνεκτικών συνιστωσών στο δίκτυο αναφορών χρησιμοποιήθηκε ο αλγόριθμος του Kosaraju (Thomas H. Cormen, 2001). Ο αλγόριθμος αποτελείται από δύο κυρίως ρουτίνες: την DFS Loop και την DFS (DFS – Depth First Search – Αναζήτηση πρώτα κατά βάθος) και μπορεί να υπολογίσει τις ισχυρά συνεκτικές συνιστώσες ενός γραφήματος σε χρόνο O(m+n) όπου m ο αριθμός των ακμών και n ο αριθμός των κόμβων. Δηλαδή ο αλγόριθμος είναι γραμμικού χρόνου (Alfred V. Aho, 1983). Όση ώρα χρειάζεται για να διαβάσει την είσοδο τόση ώρα επίσης χρειάζεται για την έξοδο (Roughgarden, 2012).

Αλγόριθμος:

Είσοδος: Ένα κατευθυνόμενο γραφήμα $G(V,E)$ με αναπαράσταση λίστας γειτνίασης. Θεωρούμε πως οι κόμβοι είναι αριθμημένοι από το 1 έως το n.

1. Ορίζουμε ως G^{rev} = G με όλες τις ακμές ανεστραμμένες.

2. Τρέχουμε το DFS-Loop στον G^{rev}, επεξεργαζόμενοι τις κορυφές με συγκεκριμένη σειρά για να πάρουμε χρόνο τελειώματος $f(v)$ για κάθε $v \in V$

3. Τρέχουμε το DFS-Loop στον G, επεξεργαζόμενοι τις κορυφές σε φθίνουσα τιμή του $f(v)$, για να αναθέσουμε ένα αρχηγό για κάθε $v \in V$.

4. Οι ισχυρά συνεκτικές συνιστώσες του G, αντιστοιχούν στις κορυφές του G που μοιράζονται τον ίδιο αρχηγό.

Έξοδος: οι ισχυρά συνεκτικές συνιστώσες του γράφου G (οι κορυφές κάθε ισχυρά συνεκτικής συνιστώσας έχουν τον ίδιο αρχηγό).

Η ρουτίνα DFS-Loop:

Είσοδος: ένα κατευθυνόμενο γράφημα $G(V, E)$ με αναπαράσταση λίστας γειτνίασης.

1. Αρχικοποίηση μιας global μεταβλητής t = 0 (κρατάει τον αριθμό των κόμβων που έχουν προσπελαστεί).

2. Αρχικοποίηση μιας global μεταβλητής s = NULL (κρατάει τον κόμβο από τον οποίο έχει καλεστεί τελευταία φορά η DFS.

3. For i=n έως το 1:

   (στην πρώτη κλήση οι κόμβοι αριθμούνται από το 1 έως το n τυχαία. Στη δέυτερη κλήση οι κόμβοι αριθμούνται με βάση τις τιμές $f(v)$ από την πρώτη κλήση).

   a) Αν το i δεν έχει ακόμα προσπελαστεί
      i) Θέσε s = i
      ii) DFS(G, i)

Έξοδος: Για κάθε κόμβο έχουμε το χρόνο τελειώματός του στο πρώτο πέρασμα και στο δέυτερο πέρασμα ανατίθεται ένας αρχηγός σε κάθε κόμβο.

Η ρουτίνα DFS:

Είσοδος: ένα κατευθυνόμενο γράφημα $G(V, E)$ με αναπαράσταση λίστας γειτνίασης και ένας κόμβος-αρχή $i \in V$.

1. Σημείωσε τον κόμβο i ως εξερευνημένο.
   (παραμένει εξερευνημένος για ολόκληρη τη διάρκεια του DFS-Loop.)

2. Θέσε $leader(i) \coloneqq s$

3. Για κάθε ακμή $arc(i, j) \in G$:

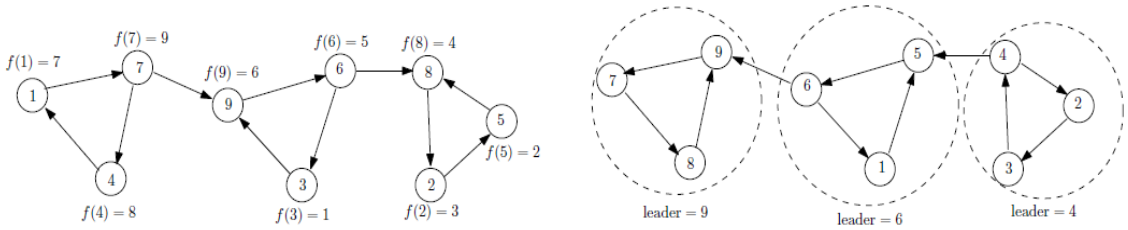
   a) Αν το j δεν έχει εξερευνηθεί ακόμα:

      i) $DFS(G, j)$

4. $t++$

5. Θέσε $f(i) \coloneqq t$

Έξοδος: οι κόμβοι που μπορούν να προσπελαστούν με αναζήτηση κατά βάθος από τον κόμβο-αρχή $i \in V$.



Εικόνα 10: Ενα παράδειγμα εκτέλεσης του αλγορίθμου για τις ισχυρά συνεκτικές συνιστώσες. Στην πρώτη εικόνα οι κόμβοι αριθμούνται τυχαία και φαίνονται οι χρόνοι τελειώματός τους. Στη δέυτερη εικόνα οι κόμβοι αριθμούνται από τους χρόνους τελειώματος και φαίνονται οι αρχηγοί κάθε συνεκτικής συνιστώσας.

4.4.2 SAX Parser

Τα αρχεία εισόδου του προγράμματος εύρεσης τάσεων όπως προέρχονται από την ιστοσελίδα του PLOS ONE είναι σε μορφή XML (Extensible Markup Language). Η XML είναι μια γλώσσα σήμανσης (markup language) που καθορίζει ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων σε μορφή που είναι αναγνώσιμη, τόσο από τους ανθρώπους, όσο και από τους υπολογιστές (XML Introduction - What is XML?). Έτσι τα επιστημονικά άρθρα κωδικοποιούνται με βάση τη γλώσσα XML για να μπορέσει το πρόγραμμά μας να τα διαβάσει.

Η XML κωδικοποιεί τα κείμενα χρησιμοποιώντας κυρίως tags και elements. Τα tags χρησιμοποιούνται κυρίως σαν «τίτλος» για τα δεδομένα που θα ακολουθήσουν στα elements. Τα tags ξεκινούν με < και τελειώνουν με > και χωρίζονται σε δύο είδη: τα start-tags: <example> και τα end-tags: </example>, οπου ενδιάμεσά τους βρίσκονται τα elements που περιέχουν το κείμενο που μας ενδιαφέρει.

Η δουλειά του SAX Parser (Megginson, 2004) επομένως είναι να αναγνωρίζει αυτά τα tags και να μπορεί να επιστρέφει στο χρήστη διάφορες πληροφορίες για τα tags αλλά και τα elements που περιλαμβάνονται μεταξύ των tags.

Ο SAX Parser περνά ένα ένα τα αντικείμενα ενός XML αρχείου σειριακά, ξεκινώντας από την αρχή και επιδρά σε αυτά καθώς τα βρίσκει. Για το λόγο αυτό χρησιμοποιεί κυρίως τρεις ρουτίνες: μία που καλείται όταν διαβάζει ένα start-tag, μία που καλείται όταν διαβάζει ένα element αμέσως μετά το start-tag και μία όταν διαβάζει το end-tag στο τέλος του element (What is SAX?). Έτσι ο χρήστης με αυτές τις τρεις ρουτίνες μπορεί να καθορίσει ποια tags τον ενδιαφέρουν και όταν ο SAX Parser διαβάσει αυτά τα tags να κάνει τις απαραίτητες ενέργειες.

Όσον αφορά στο πρόγραμμά μας θέλουμε να ανακτήσουμε από τα XML αρχεία τις βιβλιογραφικές αναφορές και ιδιαίτερα τον τίτλο της κάθε αναφοράς. Έτσι με τον SAX Parser διαβάζεται το XML αρχείο και όταν αναγνωρίσει ένα tag που αναφέρεται σε βιβλιογραφική αναφορά το επεξεργάζεται και μας επιστρέφει τον τίτλο. Επίσης ο SAX Parser χρησιμοποιείται και για την ανάκτηση των PMID μετά την αναζήτηση στην PubMed με τους τίτλους της κάθε αναφοράς, αφού η αναζήτηση επιστρέφει και αυτή ένα αρχείο XML.

4.4.3 E-Utilities της PubMed

Η PubMed προσφέρει μια σειρά από APIs τα οποία βοηθούν τους χρήστες να εκτελέσουν εύκολα και γρήγορα μια σειρά από λειτουργίες στη βάση δεδομένων MEDLINE. Το Entrez Programming Utilities (E-Utilities) της PubMed είναι ένα σύνολο από οκτώ server-side προγράμματα που παρέχουν μια σταθερή διαπροσωπεία (interface) στο σύστημα βάσεων δεδομένων Entrez του Εθνικού Κέντρου Βιοτεχνολογίας (National Center for Biotechnology Information – NCBI). Το E-Utilities χρησιμοποιεί μια συγκεκριμένη σύνταξη URL η οποία μεταφράζει ένα τυποποιημένο σύνολο από παραμέτρους εισόδου στις τιμές που απαιτούνται από το λογισμικό του NCBI για αναζήτηση και ανάκτηση δεδομένων. Έτσι το E-Utilities είναι ένα δομημένο περιβάλλον του συστήματος Entrez που σήμερα περιλαμβάνει 38 βάσεις δεδομένων που καλύπτουν μια ποικιλία από βιοϊατρικά δεδομένα που συμπεριλαμβάνουν αλληλουχίες νουκλεοτιδίων και πρωτεϊνών, αρχεία γονιδίων, τρισδιάστατες μοριακές δομές αλλά και βιοϊατρική βιβλιογραφία (Sayers, A General Introduction to the E-utilities - Entrez Programming Utilities Help - NCBI Bookshelf, 2009).

Η πρόσβαση στα δεδομένα γίνεται από ένα κομμάτι λογισμικού το οποίο δίνει το κατάλληλο URL στο NCBI και ακολούθως ανακτά τα αποτελέσματα και τα επεξεργάζεται. Επομένως απαιτείται λογισμικό που να μπορεί να στείλει μια διεύθυνση URL στον server του E-Utilities και ακολούθως να ερμηνεύσει την XML απάντηση.

Όσον αφορά στο πρόγραμμα για την εύρεση τάσεων, το κομμάτι του E-Utilities που μας ενδιαφέρει περισσότερο είναι το E-Search. Το E-Search απαντά σε ένα ερώτημα από κείμενο (text query) με ένα κατάλογο από UIDs (PMIDs) από μια βάση δεδομένων, μαζί με τις μεταφράσεις των όρων (όπως τις κατανοεί το σύστημα) που χρησιμοποιήθηκαν στο ερώτημα.

Στο πρόγραμμά μας χρησιμοποιούμε το E-Search κάνοντας αναζήτηση στην PubMed με τους τίτλους των βιβλιογραφικών αναφορών του κάθε άρθρου που ανακτήσαμε. Παίρνουμε το κάθε τίτλο, σχηματίζουμε το κατάλληλο ερώτημα και το κατάλληλο URL και ακολούθως με το E-Search ρωτάμε τη βάση δεδομένων για το ποιο είναι το PMID που αντιστοιχεί στο άρθρο με τον τίτλο αυτό. Η απάντηση έρχεται σε μορφή XML.

4.5 Λεπτομέρειες Υλοποίησης

Στα πλαίσια της παρούσας εργασίας, επιλέξαμε να χρησιμοποιήσουμε τη γλώσσα προγραμματισμού Java. Η Java μας προσφέρει όλες τις δυνατότητες και τα εργαλεία, για να πετύχουμε το σκοπό μας αφού είναι μια γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται ευρέως και διαθέτει πάρα πολλές βιβλιοθήκες λογισμικού. Επίσης υπάρχει μια πλειάδα πληροφοριών για την Java και τις βιβλιοθήκες της στο διαδύκτιο γεγονός που διευκολύνει αισθητά τον προγραμματισμό σε Java. Παρακάτω ακολουθούν κάποιες λεπτομέρειες της υλοποίησης του προγράμματος για εύρεση τάσεων.

4.5.1 HTTP GET Method

Για την επικοινωνία με την PubMed αλλά και με την PLOS ONE χρησιμοποιούμε τη μέθοδο του HTTP GET. Αυτό μας βολεύει κυρίως στην περίπτωση του E-Search της PubMed αφού οι ερωτήσεις στη βάση δεδομένων γίνονται μέσω URL queries ακριβώς όπως ορίζει και η μέθοδος GET. Έτσι για την επικοινωνία με τους διάφορους εξυπηρετητές χρησιμοποιήθηκε η παρακάτω ρουτίνα:

```
public static String getHTML(String urlToRead) {
            URL url;
            HttpURLConnection conn;
            BufferedReader rd;
            String line;
            String result = "";
            try {
               url = new URL(urlToRead);
               conn = (HttpURLConnection) url.openConnection();
               conn.setRequestMethod("GET");
               rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
               while ((line = rd.readLine()) != null) {
                  result += line;
               }
               rd.close();
            } catch (Exception e) {
               e.printStackTrace();
            }
            return result;
         }
```

4.5.2 URL για E-Search

Όπως είδαμε παραπάνω για να επικοινωνήσουμε με τους εξυπηρέτες του E-Utilities και να κάνουμε μια αναζήτηση χρησιμοποιούμε την E-Search που απαιτεί το ερώτημα να είναι κωδικοποιημένο σε μορφή URL (Sayers, E-Utilities Quick Start, 2008).

Όλες οι κλήσεις E-Utilities ξεκινούν με την ίδια βασική URL:

`http://eutils.ncbi.nlm.nih.gov/entrez/eutils/`

Εμείς, για να αναζητήσουμε στη βάση δεδομένων της PubMed με συγκεκριμένους τίτλους άρθρων, χρησιμοποιούμε την παραπάνω URL και προσθέτουμε στο τέλος της, τις κατάλληλες παραμέτρους:

`http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=`

και συγκεκριμένα το `esearch.fcgi?` για να δηλώσουμε πως μας ενδιαφέρει το E-Search, το `db=pubmed` για δηλώσουμε πως η βάση δεδομένων στην οποία θέλουμε να κάνουμε αναζήτηση είναι η pubmed και το `&term=` ακολουθόμενο από το ερώτημά μας για να πούμε στο E-Search τι θα αναζητήσει.

Ο όρος που θα ακολουθήσει μετά το `&term=` πρέπει να έχει συγκεκριμένη μορφή. Θα είναι το κείμενο που θέλουμε να αναζητήσουμε, στην περίπτωσή μας οι τίτλοι των διάφορων βιλβιογραφικών αναφορών, χωρίς όμως κενά και σημεία στίξης. Στη θέση των κενών χρησιμοποιείται το σύμβολο + και στη θέση των σημείων στίξης χρησιμοποιούνται κάποια αναγνωριστικά για κάθε σημείο στίξης.

Για παράδειγμα αν θέλουμε να κάνουμε αναζήτηση για το άρθρο που έχει τίτλο:

```
Breast Cancer, Sickness Absence, Income and Marital Status. A Study
on Life Situation 1 Year Prior Diagnosis Compared to 3 and 5 Years
after Diagnosis
```

Πρέπει να μετατρέψουμε τον τίτλο αρχικά σε:

```
Breast+Cancer%2C+Sickness+Absence%2C+Income+and+Marital+Status.+A+Stu
dy+on+Life+Situation+1+Year+Prior+Diagnosis+Compared+to+3+and+5+Years
+after+Diagnosis
```

Και ακολούθως να δημιουργήσουμε το URL:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=Breast+Cancer%2C+Sickness+Absence%2C+Income+and+Marital+Status.+A+Study+on+Life+Situation+1+Year+Prior+Diagnosis+Compared+to+3+and+5+Years+after+Diagnosis\

4.5.3 Ρουτίνες για SAX Parser

Όπως είπαμε παραπάνω χρησιμοποιούμε τρεις ρουτίνες για τον SAX Parser. Μια για όταν συναντάται το tag και επομένως είναι η αρχή του element, μια για επεξεργασία των δεδομένων του element και μια για το κλείσιμο του element οταν συνανταται το end-tag.

Η java προσφέρει έτοιμες βιβλιοθήκες για τον SAX Parser. Εμείς χρησιμοποιήσαμε το SAXParserFactory για την υλοποίηση του Parser μας και τις εξής τρεις ρουτίνες:

```
public void startElement(String uri, String localName,String qName,
            Attributes attributes) throws SAXException

public void endElement(String uri, String localName,
            String qName) throws SAXException

public void characters(char ch[], int start, int length) throws
            SAXException
```

4.5.4. InputStream και Scanner για διάβασμα αρχείου με τις ακμές του γράφου

Αφού βρεθούν όλα τα PMIDs των άρθρων που μας ενδιαφέρουν αποθηκεύονται σε ένα αρχείο input με την εξής μορφή: Η μορφή του αρχείου input και άρα της εισόδου για την αναπαράσταση του γράφου είναι η εξής:

Στην πρώτη στήλη βρίσκεται το PMID του κάθε άρθρου προς ανάλυση και στη δεύτερη στήλη βρίσκονται τα PMIDs των βιβλιογραφικών του αναφορών.

Επομένως κάθε γραμμή του αρχείου αναπαριστά και μια ακμή του γράφου με αρχή τον κόμβο που βρίσκεται στα αριστερά και τέλος τον κόμβο που βρίσκεται δεξιά.

Για το διάβασμα αυτού του αρχείου χρησιμοποιήσαμε την κλάση InputStream.

Η InputStream είναι η βασική κλάση (υπερκλάση) όλων των εισόδων ρευμάτων στο Java IO API. Υποκλάσεις περιλαμβάνουν την FileInputStream, BufferedInputStream και το PushbackInputStream μεταξύ άλλων. Τα InputStreams χρησιμοποιούνται για την ανάγνωση δεδομένων byte, ένα byte ανά στιγμή.

Παρακάτω φαίνεται ο αντίστοιχος κώδικας:

```
InputStream is = new
FileInputStream("C:\\Users\\Costas\\workspace\\TrendMining\\InputGrap
h\\input.txt");
```

Έτσι το InputStream is, πλέον περιέχει όλα τα περιεχόμενα του αρχείου input, δηλαδή όλα τα δεδομένα που χρειαζόμαστε, για να κατασκευάσουμε το γράφο μας.

Για να διαβάσουμε τα περιεχόμενα του InputStream is χρησιμοποιούμε ένα Scanner sc.

Η κλάση scanner μας παρέχει ένα απλό σαρωτή κειμένου που μπορεί να χωρίσει (parse) πρωτόγονους τύπους (primitive types) και strings χρησιμοποιώντας κανονικές εκφράσεις.

Έτσι εμείς σαρώνουμε το InputStream is με ένα scanner έτσι ώστε να πάρουμε το PMID που ανιστοιχεί στον κάθε κόμβο. Το επόμενο PMID ανακτάται μέσω του scanner με τη μέθοδο `sc.nextInt()` που βρίσκει τον επόμενο ακέραιο αριθμό στο is.

4.5.5 Ρουτίνες ReadGraph και addEdge για δημιουργία γράφου

Στη ρουτίνα readGraph γίνεται το διάβασμα του inputStream με το scanner όπως αναφέρθηκε παραπάνω:

```
private static DirectedGraph readGraph( InputStream is ) throws
FileNotFoundException
```

Έτσι παίρνοντας τα PMIDs από το αρχείο input μέσω του scanner καλούμε την addEdge για να δημιουργήσουμε τις ακμές του γράφου:

```
while( sc.hasNext() ) {
            addEdge( gr, sc.nextInt(), sc.nextInt() );
        }
```

Η ρουτίνα addEdge ορίζεται ως εξής:

```
private static void addEdge(DirectedGraph gr, int tailId, int headId)
```

Παίρνει σαν ορίσματα το γράφο gr και τους δύο ακεραίους που διαβάζουμε με το scanner και αντιπροσωπευούν την ουρά και την κεφαλή της ακμής.

Ακολούθως δημιουργείται η ουρά και η κεφαλή:

```
DirectedVertex tail = gr.getVertex( tailId );
DirectedVertex head = gr.getVertex( headId );
```

και στη συνέχεια δημιουργείται η ίδια η ακμή και προστίθεται στο γράφο:

```
DirectedEdge edge = new DirectedEdge( tail, head );
gr.addEdge( edge );
tail.addOutgoingEdge( edge );
head.addIncomingEdge( edge );
```

Οι μέθοδοι DirectedGraph, DirectedEdge κλπ αποτελούν υποκλάσεις των αφηρημένων κλάσεων Graph και Edge που εξηγούνται παρακάτω.

4.5.6 Abstract class Graph και VertexFactory

Για τη δημιουργία του γράφου, χρησιμοποιούμε μια abstract κλάση τη Graph.

Μια αφηρημένη κλάση (abstract) είναι μια κλάση που έχει δηλωθεί αφηρημένη και μπορεί ή δεν μπορεί να περιλαμβάνει αφηρημένες μεθόδους. Οι αφηρημένες κλάσεις δεν μπορούν να γίνουν instantiated, αλλά μπορούν να δημιουργηθούν υποκλάσεις αυτών. 'Οταν δημιουργηθεί μια υποκλάση της αφηρημένης κλάσης αυτή συνήθως υλοποιεί τις αφηρημένες μεθόδους της υποκλάσης. Έτσι οι αφηρημένες κλάσεις δίνουν μια μερική υλοποίηση της κλάσης και αφήνουν την περαιτέρω υλοποίηση στις υποκλάσεις τους. Επομένως οι αφηρημένες κλάσεις δημιουργούνται, για να μοιράζονται μέρη της υλοποίησής τους με τις υποκλάσεις τους.

Στο πρόγραμμα μας έχουμε δημιουργήσει την αφηρημένη κλάση Graph που αποτελεί και την κύρια υλοποίηση για το γράφο μας:

```
public abstract class Graph<V extends AbstractVertex<E>, E extends
AbstractEdge<V>> {
```

Επίσης έχουμε δημιουργήσει και τις αφηρημένες κλάσεις `AbstractVertex` και `AbstractEdge`.

Οι αφηρημένες αυτές κλάσεις υλοποιούνται αργότερα με τις υποκλάσεις τους `Graph`, `Vertex`, `Edge`, `DirectedEdge` και περιέχουν την υλοποίηση των μεθόδων για τη δημιουργία του γράφου με τις κορυφές και τις ακμές του.

Για τη δημιουργία του γράφου χρησιμοποιούμε επίσης ένα VertexFactory

```
private VertexFactory<V, E> f;
```

Το μοτίβο μέθοδος εργοστάσιου (factory method pattern) είναι ένα αντικειμενοστραφές object-oriented πρότυπο σχεδίασης που ασχολείται με το πρόβλημα της δημιουργίας αντικειμένων (προϊόντων) χωρίς να προσδιορίζει την ακριβή κλάση του αντικειμένου που θα δημιουργηθεί. Η ουσία αυτού του προτύπου είναι να ορίσει μια διεπαφή για τη δημιουργία ενός αντικειμένου, αλλά να αφήσει τις κλάσεις που υλοποιούν τη διεπαφή να αποφασίσουν ποια κλάση να γίνει instantiated (να αποκτήσει υπόσταση). Η μέθοδος Factory επιτρέπει σε μια κλάση να αφήσει το instantiation για τις υποκλάσεις.

Στην περίπτωσή μας αυτό είναι βολικό αφού ο γράφος δημιουργείται στις υποκλάσεις των αφηρημένων κλάσεων που χρησιμοποιούμε και η κάθε κλάση μπορεί να κάνει instatiate τη δική της μέθοδο όπως αυτή θέλει ενώ παράλληλα μοιράζεται ένας μέρος της υλοποίησης με την αφηρημένη κλάση.

4.5.7 Ρουτίνες dfsLoop και dfs

Για την υλοποίηση του αλγορίθμου του Kosaraju για τον εντοπισμό ισχυρά συνεκτικών συνιστωσών υλοποίησαμε τις δύο ρουτίνες που απαιτούνται, την dsfLoop και την dfs όπως τις ορίσαμε παραπάνω.

Η ρουτίνα dfsLoop:

```
private static void dfsLoop(DirectedGraph gr, EdgeTraversalPolicy tp)
```

παίρνει σαν είσοδο ένα κατευθυνόμενο γράφημα `gr` αλλά και τον τρόπο με τον οποίο θα γίνει η προσπέλαση των κόμβων μέσω του `EdgeTraversalPolicy tp`. Το `EdgeTraversalPolicy` καθορίζει αν η προσπέλαση των κόμβων θα γίνει με τις ακμές ανεστραμμένες (κλήση: *dfsLoop*(gr, DirectedGraph.*BACKWARD_TRAVERSAL*)) ή κανωνικά (κλήση: *dfsLoop*(gr, DirectedGraph.*FORWARD_TRAVERSAL*)) όπως απαιτεί ο αλγόριθμος.

Το `EdgeTraversalPolicy tp` περνά με τον ίδιο τρόπο και στην ρουτίνα dfs που καλείται μέσα από την dfsLoop (μαζί με τον κόμβο από τον οποίο θα αρχίσει το dfs):

```
private static void dfs( EdgeTraversalPolicy tp, DirectedVertex v )
```

Η υλοποίηση των δύο αυτών ρουτίνων ακουλουθεί πιστά τον ψευδοκώδικα που δώσαμε παραπάνω.

Κεφάλαιο 5: Αποτίμηση Μηχανισμών

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση της εφαρμογής των μηχανισμών που υλοποιήσαμε δίνοντας κάποια στιγμιότυπα από την έξοδο του προγράμματος και αναφέροντας σχόλια για τη ροή εκτέλεσης του προγράμματος για τον εντοπισμό τάσεων.

5.1 Εφαρμογή Μηχανισμών

Παρακάτω φαίνονται τα αποτελέσματα (output) του προγράμματος που υλοποιήσαμε:

Για μια αναζήτηση στην PLOS ONE με τη λέξη κλειδί Breast Cancer παίρνουμε 25987 διαφορετικά άρθρα (εμείς το πρόγραμμά μας το τρέξαμε με πολύ λιγότερα άρθρα τόσο για εξοικονόμηση χρόνου όσο και για να μπορούμε να το ελέξουμε καλύτερα).

Τα άρθρα αυτά μπαίνουν σε ένα φάκελο XMLArticles σε μορφή XML όπου από εκεί περιμένει το πρόγραμμά μας να τα βρει.

Παρακάτω δίνεται ένα ενδεικτικό άρθρο στην PLOS ONE:

http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0018040

Τα άρθρα στο φάκελο αυτό έχουν την ονομασία: journal.pone.0018040 όπου οι τελευταίοι αριθμοί αλλάζουν ανάλογα με τον κωδικό του κάθε άρθρου στην PLOS ONE.

Ακολούθως τρέχουμε το πρόγραμμά μας στα διάφορα άρθρα:

Παρακάτω βλέπουμε μια τυπική έξοδο όπου φαίνεται το πώς το πρόγραμμά μας διαβάζει το άρθρο journal.pone.0018040 παρουσιάζοντας τον τίτλο του, πώς ετοιμάζει τον τίτλο για να προστεθεί στο τέλος του URL του E-Utilities, αλλά και το ολοκληρωμένο URL που μας δίνει ένα αρχείο σε XML μορφή με το PMID του κάθε άρθρου.

```
Trend Mining in Biomedical Literature Application
journal.pone.0018040.xml
Breast Cancer, Sickness Absence, Income and Marital Status. A Study
on Life Situation 1 Year Prior Diagnosis Compared to 3 and 5 Years
after Diagnosis
Encoded Text:
Breast+Cancer%2C+Sickness+Absence%2C+Income+and+Marital+Status.+A+Stu
dy+on+Life+Situation+1+Year+Prior+Diagnosis+Compared+to+3+and+5+Years
+after+Diagnosis
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=Breast+Cancer%2C+Sickness+Absence%2C+Income+and+Marital+Status.+A
+Study+on+Life+Situation+1+Year+Prior+Diagnosis+Compared+to+3+and+5+Y
ears+after+Diagnosis


Reference List starts here : References
1. Cancer survival in Sweden 1960-1998-developments across four
decades.
Encoded Text: Cancer+survival+in+Sweden+1960-
1998%96developments+across+four+decades.
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=Cancer+survival+in+Sweden+1960-
1998%96developments+across+four+decades.
```

```
2. Survival trends in European cancer patients diagnosed from 1988 to
1999.
Encoded Text:
Survival+trends+in+European+cancer+patients+diagnosed+from+1988+to+19
99.
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=Survival+trends+in+European+cancer+patients+diagnosed+from+1988+t
o+1999.


3. A systematic overview of chemotherapy effects in breast cancer.
Encoded Text:
A+systematic+overview+of+chemotherapy+effects+in+breast+cancer.
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=A+systematic+overview+of+chemotherapy+effects+in+breast+cancer.



4. Effects of chemotherapy and hormonal therapy for early breast
cancer on recurrence and 15-year survival: an overview of the
randomised trials.
Encoded Text:
Effects+of+chemotherapy+and+hormonal+therapy+for+early+breast+cancer+
on+recurrence+and+15-
year+survival%3A+an+overview+of+the+randomised+trials.
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=Effects+of+chemotherapy+and+hormonal+therapy+for+early+breast+can
cer+on+recurrence+and+15-
year+survival%3A+an+overview+of+the+randomised+trials.

<?xml version="1.0" ?><!DOCTYPE eSearchResult PUBLIC "-//NLM//DTD >
<OP>O

5. Effects of radiotherapy and of differences in the extent of
surgery for early breast cancer on local recurrence and 15-year
survival: an overview of the randomised trials.
Encoded Text:
Effects+of+radiotherapy+and+of+differences+in+the+extent+of+surgery+f
or+early+breast+cancer+on+local+recurrence+and+15-
year+survival%3A+an+overview+of+the+randomised+trials.
URL:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&t
erm=Effects+of+radiotherapy+and+of+differences+in+the+extent+of+surge
ry+for+early+breast+cancer+on+local+recurrence+and+15-
year+survival%3A+an+overview+of+the+randomised+trials.
```

Ακολούθως το πρόγραμμά μας ανακτά τα αρχεία xml που περιέχουν το PMID για την κάθε βιβλιογραφική αναφορά και τα αποθηκεύει σε ένα φάκελο ReferencesInXML:

```
C:\Users\Costas\workspace\TrendMining\ReferecesInXML\1.xml
C:\Users\Costas\workspace\TrendMining\ReferecesInXML\2.xml
C:\Users\Costas\workspace\TrendMining\ReferecesInXML\3.xml
C:\Users\Costas\workspace\TrendMining\ReferecesInXML\4.xml
C:\Users\Costas\workspace\TrendMining\ReferecesInXML\5.xml
```

Στη συνέχεια επεξεργάζεται όλα τα αρχεία xml που βρίσκονται στο φάκελο ReferencesInXML και ανακτά από αυτά το PMID της κάθε βιβλιογραφικής αναφοράς. Το κάθε PMID αποθηκεύεται σε μορφή .txt σε ένα άλλο φάκελο PMIDRefereces.

```
1.txt
21479209
2.txt
14690151
3.txt
16360786
4.txt
10807141
5.txt
12209737
```

Μετά το πρόγραμμά μας διαβάζει όλα τα αρχεία .txt που περιέχουν τα PMIDs και δημιουργεί ένα άλλο μοναδικό αρχείο, το input, που βρίσκεται στο φάκελο InputGraph. Το αρχείο input χρησιμεύει ως είσοδος για τη δημιουργία του δικτύου αναφορών και ακολούθως της εύρεσης των ισχυρά συνεκτικών συνιστωσών.

Η μορφή του αρχείου input και άρα της εισόδου για την αναπαράσταση του γράφου είναι η εξής:

Στην πρώτη στήλη βρίσκεται το PMID του κάθε άρθρου προς ανάλυση και στη δεύτερη στήλη βρίσκονται τα PMIDs των βιβλιογραφικών του αναφορών.

Παρακάτω φαίνεται το αρχείο input για τρία διαφορετικά άρθρα με PMIDs τα 21479209, 22984585, 23049907:

```
21479209 18670868
21479209 20672244
21479209 9890170
21479209 17627811
21479209 19148756
21479209 19907646
21479209 18792789
21479209 8880841
21479209 18504613
21479209 12915872
21479209 14690151
21479209 18585963
21479209 23182226
21479209 19036577
21479209 18032531
21479209 16360786
21479209 10807141
21479209 12209737
21479209 11918907
21479209 22009703
21479209 19224752
21479209 18937082
```

22984585 19587092
22984585 12902989
22984585 12947545
22984585 17074969
22984585 22156622
22984585 22412199
22984585 17965115
22984585 20565850
22984585 19126586
22984585 8592733
22984585 20098429
22984585 17276673
22984585 15911104
22984585 20857494
22984585 18334282
22984585 20966643
22984585 21626328
22984585 22193644
22984585 20711237
22984585 19843073
23049907 12832461
23049907 15634327
23049907 21136898
23049907 9307301
23049907 9664136
23049907 10197442
23049907 12096109
23049907 15734831
23049907 20527979
23049907 14754989
23049907 16675915
23049907 17031575
23049907 15169658
23049907 19444910
23049907 21844121
23049907 16831269
23049907 19908260
23049907 15731336
23049907 15797904
23049907 18430892
23049907 22084684
23049907 12750300
23049907 18062746
23049907 19383334
23049907 15574759
23049907 21036680

Ακολούθως αφού αναλυθούν όλα τα άρθρα, δημιουργείται ο *γράφος* και τρέχει ο αλγόριθμος του Kosaraju για την εύρεση των ισχυρά συνεκτικών συνιστωσών στο δίκτυο αναφορών.

Παρακάτω φαίνεται η έξοδος του προγράμματος με τον αριθμό των κόμβων και των ακμών αλλά και το μέγεθος των 5 μεγαλύτερων συνεκτικών συνιστωσών:

```
Read from file: 45 ms
Graph: 145 vertices, 563 edges.

ssc:3
ssc:1
ssc:1
ssc:1
ssc:1
```

Τα PMIDs των κόμβων που αποτελούν τη μεγαλύτερη ισχυρά συνεκτική συνιστώσα αποθηκεύονται σε ένα νέο φάκαλο SCCs. Από εκεί ο χρήστης μπορεί να μπει και να δει τα άρθρα και ακολούθως να αποφανθεί αν όντως αποτελούν μια ανερχόμενη τάση.

Κεφάλαιο 6: Σύνοψη και Μελλοντική Έρευνα

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα αλλά και τα συμπεράσματά μας σε μορφή πλεονεκτημάτων και μειονεκτημάτων της μεθοδολογίας που ακολουθήθηκε. Στη συνέχεια γίνεται μια σύνοψη της διπλωματικής εργασίας όπου περιγράφεται ο στόχος της διπλωματικής, η εργασία μας και το τι έχουμε πετύχει κατά την εκπόνησή της. Ακολούθως δίνονται κάποιες ιδέες για μελλοντική έρευνα.

6.1 Συμπεράσματα

Θα αναλύσουμε τη μεθοδολογία που επιλέξαμε αλλά και τα αποτελέσματα του προγράμματος που φτιάξαμε παραθέτοντας τα πλεονεκτήματα αλλά και τα μειονεκτήματα σε σχέση με την ικανότητά του να βρίσκει ή όχι τάσεις.

6.1.1 Πλεονεκτήματα

- Η μέθοδος που επιλέχθηκε δε χρειάζεται προεπεξεργασία των άρθρων.

  Τα διάφορα άρθρα μπορούν να ληφθούν όπως έχουν χωρίς να απαιτηθεί προηγουμένως επεξεργασία. Το μόνο που απαιτείται είναι τα αρχεία να βρίσκονται σε μορφή XML, για να μπορούν να διαβαστούν από το πρόγραμμα. Η βιβλιοθήκη PLOS ONE παρέχει όλα τα άρθρα σε μορφή XML. Αυτό διευκολύνει πολύ την ανάλυση των άρθρων αφού δεν απαιτούνται πολύπλοκες τεχνικές εξόρυξης δεδομένων.

- Η τεχνική της άμεσης αναφοράς δίνει πολύ καλύτερα αποτελέσματα σε σχέση με τη βιβλιογραφική αναφορά και τη συν-αναφορά.

  Όπως είδαμε παραπάνω το γεγονός πως ένα άρθρο αναφέρεται στη βιβλιογραφία ενός άλλου αποτελεί ένδειξη πως τα άρθρα σχετίζονται μεταξύ τους. Η άμεση αναφορά μας προσφέρει αυτή την πολύτιμη πληροφορία και έτσι μπορούμε να βρούμε και να ομαδοποιήσουμε τα άρθρα που σχετίζονται μεταξύ τους με σχετικά έυκολο τρόπο.

- Το γεγονός ότι τα αποτελέσματα παρουσιάζονται σε συστάδες βοηθά το χρήστη στην καλύτερη κατανόηση της εξόδου.

  Η ομαδοποίηση των αποτελεσμάτων κάνει πιο εύκολη την κατανόησή τους από το χρήστη αφού μπορεί να δει ποια άρθρα βρίσκονται σε κάθε συστάδα και να μάθει περισσότερες πληροφορίες γι’ αυτά αν το θελήσει.

6.1.2 Μειονεκτήματα

- Απαιτείται από το χρήστη στο τέλος του προγράμματος να αποφανθεί αν τα αποτελέσματα αποτελούν τάσεις.

  Η μεθοδολογία που ακολουθήσαμε παρουσιάζει τα άρθρα τα οποία αναφέρονται περισσότερο μεταξύ τους. Αυτό από μόνο του δε συνεπάγεται αμέσως πως και μια ισχυρά συνεκτική συνιστώσα αποτελεί ανερχόμενη τάση αλλά μια ένδειξη. Την τελική

απόφαση θα πρέπει να την πάρει ένας ειδικός ο οποίος θα γνωρίζει αν όντως τα αποτελέσματα αποτελούν τάσεις και να είναι σε θέση να απορρίψει τυχόντα σφάλματα της μεθόδου.

- Ο τομέας της Βιοϊατρικής περιλαμβάνει πολλά θέματα τα οποία αλληλοεπικαλύπτονται.

  Για είσοδο στο πρόγραμμά μας επιλέξαμε να έχουμε αρχεία που ασχολούνται με τη βιοϊατρική. Λόγω της φύσης όμως της βιοϊτρικής πολλά από τα θέματα τα οποία μπορούν να θεωρηθούν ως διαφορετικές τάσεις επικαλύπτονται και παρουσιάζονται ως ένα αφού τα άρθρα που τα αποτελούν αλληλοαναφέρονται αρκετά.

- Τα αποτελέσματα του προγράμματός μας δεν οπτικοποιούνται.

  Για καλύτερη κατανόηση του συνόλου των δεδομένων που έχουμε αλλά και των τάσεων που βρίσκουμε θα μπορούσαμε να οπτικοποιήσουμε τα αποτελέσματα. Θα ήταν χρήσιμο το δίκτυο αναφορών να οπτικοποιόταν και, για παράδειγμα, οι κόμβοι κάθε συνεκτικής συνιστώσας να παρουσιάζονταν με διαφορετικό χρώμα. Έτσι θα ήταν άμεσα εμφανές στο χρήστη ποιες είναι οι διάφορες συνεκτικές συνιστώσες και άρα και οι προτεινόμενες τάσεις.

- Η ταχύτητα με την οποία εντοπίζονται οι αναφορές κάθε άρθρου είναι ικανοποιητική αλλά όχι βέλτιστη.

  Λόγω του ότι το κάθε άρθρο είναι πολύ μεγάλο η XML μορφή του περιέχει πάρα πολλά tags και elements με αποτελέσμα ο SAX Parser να χρειάζεται λίγη ώρα, για να διαβάσει ολόκληρο το άρθρο και να βρει τις βιβλιογραφικές αναφορές. Αυτό θα μπορούσε να βελτιωθεί χρησιμοποιώντας άλλες τεχνικές προσπέλασεις XML αρχείων όπως το XPath.

6.2 Σύνοψη διπλωματικής Εργασίας

Σε αυτή τη διπλωματική εργασία προσπαθήσαμε να δημιουργήσουμε και να υλοποιήσουμε μια μεθοδολογία για την ανίχνευση ερευνητικών τάσεων σε βιοϊατρική βιβλιογραφία.

Ξεκινήσαμε την έρευνά μας προσπαθώντας να καταλάβουμε γενικά τι είναι ο εντοπσιμός τασεων μελετώντας την τρέχουσα τεχνολογική κατάσταση αλλά και διάφορες μεθοδολογιίες για την ευρεση τάσεων. Ακολούθως μελετήσαμε διάφορες υλοποιημένες εφαρμογές, που κάποιες από αυτές βρίσκονται στο εμπόριο και αφορούν στον εντοπισμό τάσεων.

Ακολούθως κάναμε μια ανασκόπηση στην εξόρυξη δεδομένων που είναι ένα αναγκαίο και άρρηκτο κομμάτι του εντοπισμού τάσεων αφού για την ανάλυση των διαθέσιμων

πληροφοριών απαιτούνται μηχανισμοί που να διαβάζουν τα ήδη υπάρχοντα δεδομένα και να εξαγάγουν τις πιο σημαντικές πληροφορίες. Στη συνέχεια μελετήσαμε τα προβλήματα που οδήγησαν στη δημιουργία του πεδίου αυτού αλλά και την τρέχουσα τεχνολογική κατάσταση με διάφορες μεθοδολογίες Επίσης αναλύσαμε το πού βρίσκουν χρησιμότητα οι εφαρμογές εξόρυξης δεδομένων.

Στη συνέχεια αναπτύξαμε μια εφαρμογή για τον εντοπισμό τάσεων αφού παρουσιάσαμε διάφορες εργασίες που έχουν ήδη γίνει από προηγούμενους ερευνητές. Επίσης δώσαμε μια θεωρητική και αναλυτική περιγραφή του μοντέλου που έχει ακολουθηθεί για τον εντοπισμό τάσεων μέσω της τεχνικής της άμεσης αναφοράς και παρουσιάσαμε τους διάφορους μηχανισμούς με τις λεπτομέρειές τους, που έχουν χρησιμοποιηθεί για την υλοποίηση της εφαρμογής. Επιπλέον δώσαμε μια αποτίμηση των μηχανισμών που έχουν χρησιμοποιηθεί με διάφορα παραδείγματα εκτέλεσης της εφαρμογής για επιβεβαίωση πως η εφαρμογή μας εκτελείται σωστά. Επίσης δώσαμε ιδέες και μεθοδολογίες για μελλοντική έρευνα για τη βελτίωση των μηχανισμών που έχουν αναπτυχθεί.

Έτσι με τη διπλωματική αυτή καταφέραμε να μελετήσουμε εκτενώς το πεδίο της εύρεσης τάσεων και να κατανοήσουμε καλύτερα τους διαφορετικούς τρόπους με τους οποίους μπορεί να προσεγγιστεί. Επίσης με την υλοποίηση της μεθόδου της άμεσης αναφοράς είδαμε και στην πράξη, πως μπορούμε να φτιάξουμε ένα σύστημα που να είναι σε θέση να εντοπίζει τάσεις, ενώ παράλληλα μας έδωσε τα εφόδια για περαιτέρω ενασχόληση με τον εντοπισμό τάσεων στο μέλλον.

6.3 Μελλοντική Έρευνα

Παρακάτω δίνονται κάποιες εισηγήσεις για βελτιώσεις που μπορούν να γίνουν στο πρόγραμμα έτσι ώστε να μπορεί να εντοπίζει καλύτερα τις διάφορες τάσεις.

6.3.1 Εύρεση συστάδων σε κάθε συνεκτική συνιστώσα

Τα άρθρα-κόμβοι που αποτελούν την κάθε συνεκτική συνιστώσα αποτελούν από μόνα τους ένα γενικό θέμα που ίσως να αποτελεί μια ανερχόμενη τάση. Παρ’ όλα αυτά το θέμα είναι αρκετά γενικό και μπορεί να εξειδικευθεί σε επιμέρους θέματα που θα αποτελούν και αυτά με τη σειρά τους πιο συγκεκριμένες τάσεις.

Αυτό μπορεί να γίνει αν χωρίσουμε τις μεγαλύτερες συνεκτικές συνιστώσες (τις μικρότερες τις αγνοούμε) σε συστάδες. Για το σκοπό αυτό μπορούμε να χρησιμοποιήσουμε τη μέθοδο τοπολογικών συστάδων (topological clustering) και ιδιαίτερα τον αλγόριθμο του Newman. Ο αλγόριθμος αυτός μπορεί να εφαρμοσθεί σε μεγάλα δίκτυα μιας και ο χρόνος εκτέλεσής του είναι $O((m+n)n)$ ή $O(n^2)$ όπου m οι ακμές και n οι κόμβοι του δικτύου.

Ο αλγόριθμος του Newman βασίζεται στην ιδέα του modularity.

Το modularity Q ορίζεται ως εξής:

$$Q = \sum_s (e_{st} - \alpha_s^2) = Tr(e) - \| e \|^2$$

όπου το e_{st} ορίζεται ως ο αριθμός των ακμών που ενώνουν κόμβους στη συστάδα s με αυτούς στη συστάδα t και $\alpha_s = \sum_t e_{st}$.

Το πρώτο μέρος της εξίσωσης , *Tr(e),* αντιπροσωπεύει το άθροισμα της πυκνότητας των ακμών σε κάθε συστάδα. Μεγάλη τιμή αυτής της παραμέτρου σημαίνει πως οι κόμβοι είναι πυκνά συνδεδεμένοι σε κάθε συστάδα. Tr(e)=1 όταν όλοι οι κόμβοι θεωρηθούν σαν μια συστάδα.

Το δεύτερο μέρος της εξίσωσης $\| e \|^2$ αντιπροσωπεύει το άθροισμα των πυκνοτήτων των ακμών σε κάθε συστάδα όταν όλες οι ακμές τοποθετηθούν τυχαία.

Υψηλή τιμή για το Q σημαίνει καλός διαχωρισμός σε συστάδες αφού μόνο πυκνές ακμές παραμένουν στις συστάδες και οι αραιές ακμές αποκόπτονται, ενώ Q=0 σημαίνει πως ο οποιοσδήποτε τυχαίος διαχωρισμός σε συστάδες θα μας δώσει περίπου το ίδιο αποτέλεσμα.

Ακολούθως, για να βρούμε τον καλύτερο διαχωρισμό σε συστάδες πρέπει να μεγιστοποιήσουμε το Q σε όλους τους πιθανούς διαχωρισμούς. Στην αρχική κατάσταση κάθε κόμβος θα αποτελεί και μια συστάδα, επομένως θα έχουμε n συστάδες. Ακολούθως θα ενώνουμε κόμβους-συστάδες σε ζευγάρια διαλέγοντας κάθε φορά το ζευγάρι που οδηγεί στη μεγαλύτερη αύξηση του Q. Η διαφορά στο Q όταν ενώνονται δύο κόμβοι-συστάδες δίνεται από την εξής σχέση:

$$\Delta Q = e_{st} + e_{ts} - 2a_s a_t = 2(e_{st} - a_s a_t)$$

Θα σταματήσουμε να ενώνουμε κόμβους-συστάδες όταν το ΔQ γίνει αρνητικό.

6.3.2 Εντοπισμός ανερχόμενων τάσεων

Το πρόγραμμα που έχουμε υλοποιήσει είναι σε θέση να βρίσκει διάφορες τάσεις σε βιοϊατρικές δημοσιεύσεις. Οι τάσεις όμως που ανακαλύπτονται δεν ξέρουμε αν αποτελούν ανερχόμενες τάσεις ή υπήρξαν τάσεις κάποτε στο παρελθόν (και σε κάποιες περιπτώσεις ίσως και να έχουν πεθάνει). Επομένως μια βελτίωση στο πρόγραμμα θα ήταν να μπορεί να αποφανθεί για κάθε τάση που βρίσκει αν είναι ανερχόμενη ή όχι.

Το πότε έχουν εκδοθεί τα διάφορα άρθρα είναι μια χρήσιμη μετρική που βοηθά στον εντοπισμό των ανερχόμενων τάσεων, αφού αν γνωρίζουμε το χρόνο δημοσίευσης μπορούμε να αποφανθούμε καλύτερα αν τα άρθρα που θεωρούνται τάσεις αποτελούν όντως κάποιο καινούργιο επιστημονικό ενδιαφέρον ή είναι συνέχιση προηγούμενων προϋπάρχουσων ερευνών ή αν αποτελούν παλαιότερες τάσεις.

Για καλύτερα αποτελέσματα θα μπορούσαμε να πάρουμε τα άρθρα από την PLOS ONE και να τα κατατάξουμε ανά χρονολογία δημοσίευσης και έτσι να κατασκευάσουμε εκτός από το συνολικό δίκτυο αναφορών και ένα δίκτυο αναφορών ανά χρονολογία. Επίσης για κάθε ισχυρά συνεκτική συνιστώσα θα μπορούσαμε να βρούμε τη μέση χρονολογία δημοσίευσης των άρθρων που την αποτελούν. Νεότερη μέση χρονολογία δημοσίευσης μιας συνιστώσας θα σήμαινε πως το ερευνητικό πεδίο που αντιπροσωπεύει η συνιστώσα μεγαλώνει σε ενδιαφέρον και ενασχόληση γρηγορότερα. Έτσι θα έχουμε μια καλύτερη εικόνα για το αν το πιθανό ανερχόμενο πεδίο είναι καινούργιο ή απλώς οι ερευνητές ξανασχολούνται μαζί του.

6.3.3 Αξιολόγηση της μεθόδου χρησιμοποιώντας διάφορες μετρικές

Όπως είδαμε παραπάνω η εφαρμογή μας επιστρέφει κάποιες ομάδες άρθρων που ίσως να αποτελούν τάσεις και είναι στο χέρι του χρήστη να αποφασίσει για το αν πραγματικά είναι η όχι.

Παρ' όλα αυτά μπορούμε να ορίσουμε κάποιες μετρικές τις οποίες θα υπολογίζει το πρόγραμμα και τις οποίες θα χρησιμοποιεί, για να παρουσιάσει βελτιωμένης ακρίβειας αποτελέσματα, ενώ ταυτοχρόνως οι μετρικές αυτές θα δίνουν περισσότερες πληροφορίες στο χρήστη, για να αποφασίσει ποιες ομάδες άρθρων αποτελούν τάσεις ή όχι. Επίσης η παράμετρος του χρόνου μπορεί να μας δώσει περισσότερες πληροφορίες για το αν μια τάση είναι ανερχόμενη ή όχι.

Μέγεθος συνιστώσας/συστάδας: Όταν το μέγεθος της συστάδας είναι μεγάλο, αυτό σημαίνει πως μπορούμε να διαχωρίσουμε πιο εύκολα την ύπαρξη μιας αναδευόμενης συστάδας από άλλες συστάδες γιατί σημαίνει πως περισσότεροι ερευνητές ασχολούνται και παράγουν άρθρα για το συγκεκριμένο επιστημονικό πεδίο.

Μέση χρονολογία δημοσίευσης άρθρων συνιστώσας/συστάδας: Όταν η μέση χρονολογία δημοσίευσης είναι μικρή τότε σημαίνει πως η συστάδα μπορεί εύκολα να ανιχνευθεί και μπορούμε να ανακαλύψουμε γρηγορότερα αναδυόμενες συστάδες με σημαντικά άρθρα σε αυτές. Επίσης, νεότερη μέση χρονολογία δημοσίευσης σημαίνει πως το ερευνητικό πεδίο που αντιπροσωπεύει η συνιστώσα μεγαλώνει σε ενδιαφέρον και ενασχόληση γρηγορότερα σε σχέση με άλλα πεδία. Έτσι υπολογίζοντας την παράμετρο χρόνος μπορούμε να αποφανθούμε αν οι τάσεις που έχουμε υπολογίσει αποτελούν παλαιότερες τάσεις στην έρευνα, τάσεις που πλέον έχουν πεθάνει ή ανερχόμενες τάσεις που μπορεί να έχουν έντονο ερευνητικό ενδιαφέρον στο μέλλον.

Πυκνότητα συνιστώσας/συστάδας: Αν η συστάδα είναι πυκνή, δηλαδή οι κόμβοι που την αποτελούν συνδέονται με όσο το δυνατόν περισσότερες ακμές, τότε τα άρθρα σε αυτήν έχουν τοπολογική διάταξη και με αυτόν τον τρόπο μπορούμε να ελέγξουμε αν ο διαχωρισμός σε συστάδες έγινε σωστά.

Ακρωνύμια

Ακρωνύμια	Πλήρης Όρος
API	Application Programming Interface (διεπαφή προγραμματισμού εφαρμογών)
DFS	Depth-first search (Αναζήτηση πρώτα κατά βάθος)
ETD	Emerging Trend Detection (ανίχνευση ανερχόμενων τάσεων)
HDDI	Hierarchical Distributed Dynamic Indexing (Ιεραρχική Κατανεμημένη Δυναμική Ταξινόμηση)
HTTP	Hypertext Transfer Protocol (Πρωτόκολλο Μεταφοράς Υπερκειμένου)
MesH	Medical Subject Headings (Θεματικές Επικεφαλίδες)
NCBI	National Center for Biotechnology Information (Εθνικό Κέντρο Βιοτεχνολογικής Πληροφορίας
PLoS One	Public Library of Science (Δημόσια Βιβλιοθήκη της Επιστήμης)
PMID	PubMed Identification (Αναγνωριστικό PubMed)
SAX	Simple API for XML (απλό API για XML)
SCC	Strongly Connected Components (Ισυχρά Συνεκτικές Συνιστώσες
TOA	Technology Opportunities Analysis (Ανάλυση Ευκαιριών Τεχνολογίας)
UID	Unique Identification Number (Μοναδικόω αριθμόω αναγνώρισης)
URL	Uniform Resource Locator (Ενιαίος Εντοπιστής Πόρων)
USPTO	United States Patent Office (βάση δεδομένων διπλωμάτων ευρεσιτεχνίας των ΗΠΑ)
WHO	World Healt Organisation (Παγκόσιμιος Οργανισμός Υγείας)
XML	Extensible Markup Language

Βιβλιογραφία

Alfred V. Aho, J. E. (1983). *Data Structures and Algorithms.* Addison-Wesley.

B. Wuthrich, D. P. (1998). Daily prediction of major stock indices from textual WWW data. *Knowledge Discovery and Data Mining* , 364–368.

Berkhin, P. (2002). Survey of clustering data mining techniques. *Technical report, Accrue Software* .

Dmitriy Fradkin, P. F. (2008). Emerging Trend Prediction in Biomedical Literature. *AMIA 2008 Symposium Proceedings*, (σσ. 485-489).

Egghe, L. &. (2002). Co-citation, bibliographic coupling and a characterization of lattice citation networks. *Scientometrics, 55(3), 349–361.*

Fabian Mörchen, M. D. (2008). *Anticipating Annotations and Emerging Trends in Biomedical Literature.* Las Vegas, Nevada, USA.

Fang, Y. &. (2001). Lattices in citation networks: An investigation into the structure of citation graphs. *Scientometrics, 50(2), 273–287.*

Fayyad Usama, P.-S. G. (1996). From Data Mining to Knowledge Discovery in Databases.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7, 179–188* .

Garfield, E. (2004). Historiographic mapping of knowledge domains literature. *Journal of Information. Science, 30(2), 119–145.*

Hans-Peter Kriegel, P. K. (2009). Outlier Detection Techniques (Tutorial). *3th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009).* Bangkok, Thailand.

Ichiro Sakata, H. S. (2012). Detecting emerging research fronts based on topological measures in citation networks of scientific publications. *Technological Forecasting & Social Change* .

Klavans, R. K. (2006). Identifying a better measure of relatedness for mapping science. *Journal of theAmerican Society for Information Science and Technology, 57, 251–263* .

Kontostathis, A. (2002). An overview of Emerging Trend Detection.

L. Breiman, J. H. (1984). *Classification and Regression Trees.* Wadsworth & Brooks.

Leon M. Galitsky, W. M. (2003). A Survey of Emerging Trend.

Lindeberg, D. A. (2000). *Hisory of PubMed and MEDLINE*. Ανάκτηση από Internet Access to the National Library of Medicine: http://www.acponline.org/clinical_information/journals_publications/ecp/sepoct00/nlm.pdf

Megginson, D. (2004). *SAX*. Ανάκτηση από saxproject.org: http://www.saxproject.org/

Minh-Hoang Le, T.-B. H. (2005). Detecting Emerging Trends from Scientific Corpora. *International Journal of Knowledge and Systems Sciences Vol. 2, No. 2, June 2005* .

Monk Ellen, W. B. (2006). *Concepts in Enterprise Resource Planning, Second Edition.* Boston, MA: Thomson Course Technology.

Naoki Shibata, Y. K. (2009). Comparative Study on Methods of Detecting Research.

Naoki Shibata, Y. K. (2008). Comparative Study on Methods of Detecting Research Fronts Using Different Types of Citation.

O'Brien J. A., M. G. (2011). *Management Information Systems.* New York, NY: McGraw-Hill/Irwin.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA.*

PLOS. (n.d.). Ανάκτηση από What is PLOS: http://www.plos.org/about/what-is-plos

PLOS ONE. (n.d.). Ανάκτηση από http://www.plosone.org/static/information;jsessionid=3B74A7A753F0812BD29F7F34D211FA99

PLOS ONE: accelerating the publication of peer-reviewed science. (n.d.). Ανάκτηση από plosone.org: http://www.plosone.org/

Porter, J. (1998). Disk drives evolution. *Proceedings of 100th Anniversary Conference: Magnetic Recording and Information Storage.*

PubMed - NCBI. (n.d.). Ανάκτηση από http://www.ncbi.nlm.nih.gov/pubmed

Rada Mihalcea, P. T. (2004). TextRank: Bringing Order into Texts,.

Roughgarden, T. (2012). *Coursera: Algorithms: Design and Analalysis, Part 1.* Ανάκτηση από Coursera.org: https://class.coursera.org/algo/lecture/index

Ryan, B. (2007). *Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model.*

Saurabh Goorha, L. U. (2010). Discovery of Significant Emerging Trends. *KDD'10* .

Sayers, E. (2009). *A General Introduction to the E-utilities - Entrez Programming Utilities Help - NCBI Bookshelf*. Ανάκτηση από ncbi.nlm.nih.gov: http://www.ncbi.nlm.nih.gov/books/NBK25497/

Sayers, E. (2008). *E-Utilities Quick Start*. Ανάκτηση από ncbi.nlm.nih.gov: http://www.ncbi.nlm.nih.gov/books/NBK25500/

Schiminovich, S. (1971). Automatic classification and retrieval of documents by means of a bibliographic pattern discovery algorithm. *Information Storage and Retrieval, 6, 417–435.*

Shibata, N. K. (2008). Detecting emerging research fronts based on topological measures in citation networks of scientific publications. *Technovation, 28(11), 758–775.*

Small, H. (2006). Tracking and predicting growth areas in science. Scientometrics. *Scientometrics, 68(3), 595–610.*

Thomas H. Cormen, C. E. (2001). Introduction to Algorithms, Second Edition, 22.5, pp. 552–557. MIT Press and McGraw-Hill Section.

U. Fayyad, R. U. (2002). Evolving data mining into solutions for insights. *Communications of the ACM, 45(8):28–31* .

U. Fayyad, S. D. (n.d.). Data. *AI Magazine, 17(2):51–66* .

What is SAX? (n.d.). Ανάκτηση από webopedia.com: http://www.webopedia.com/TERM/S/SAX.html

XML Introduction - What is XML? (n.d.). Ανάκτηση από w3schools.com: http://www.w3schools.com/xml/xml_whatis.asp

Zhu Xingquan, D. I. (2007). *Knowledge Discovery and Data Mining: Challenges and Realities.* New York, NY: Hershey.

Παράρτημα: Κώδικας Υλοποίησης

Παρακάτω διατίθεται ο κώδικας που γράφτηκε για την υλοποίηση της εφαρμογής εντοπισμού τάσεων. Ο κώδικας έχει γραφτεί σε Java και είναι χωρισμένος σε κλάσεις.

TrendMining.java

```
package trendMining;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

import org.omg.CORBA.portable.InputStream;

public class TrendMining {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("Trend Mining in Biomedical Literature Application");

		File file = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\InputGraph\\input.txt");
		file.delete();

		File dir = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\XMLArticles");


		for (String fn : dir.list()) {
			ReadXMLFile.XMLPath =
"C:\\Users\\Costas\\workspace\\TrendMining\\XMLArticles\\";
			ReadXMLFile.isFirstTime = true;
			System.out.println(fn);
			ReadXMLFile.fileName = fn;
			ReadXMLFile.main(null);
			ReadXMLFile.XMLPath =
"C:\\Users\\Costas\\workspace\\TrendMining\\ReferecesInXML\\";
			ReadXMLFile.isFirstTime = false;
			ReadXMLFile.main(null);
			InputToGraph.main(null);

			File dir2 = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces");
```

```
                        File dir3 = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\ReferecesInXML");

                        for (File fl : dir2.listFiles()) {
                    fl.delete();
                  }

                        for (File fl2 : dir3.listFiles()) {
                    fl2.delete();
                  }
                }


                try {
                        KosarajuSCC.main(null);
                } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }

        //HTTP Get method
        public static String getHTML(String urlToRead) {
            URL url;
            HttpURLConnection conn;
            BufferedReader rd;
            String line;
            String result = "";
            try {
              url = new URL(urlToRead);
              conn = (HttpURLConnection) url.openConnection();
              conn.setRequestMethod("GET");
              rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
              while ((line = rd.readLine()) != null) {
                result += line;
              }
              rd.close();
            } catch (Exception e) {
              e.printStackTrace();
            }
            return result;
          }
}
```

ReadXMLFile.java

```
package trendMining;

//import TrendMining;

import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class ReadXMLFile {

  public static String XMLPath;
  public static String fileName;
  public static boolean isFirstTime;
  public static int numOfRef = 0;
  public static int j=1;


  public static void main(String argv[]) {


   try {

        SAXParserFactory factory = SAXParserFactory.newInstance();
        SAXParser saxParser = factory.newSAXParser();


        DefaultHandler handler = new DefaultHandler() {

        boolean isRefList = false;
        boolean refStart = false;
        boolean isNewArticle = false;
        boolean isNewArticle2 = false;
        boolean isArticleTitle = false;
        boolean isCount = false;
        boolean flag = false;
        boolean isPMID = false;
        //int numOfRef = 0;


        public void startElement(String uri, String localName,String qName,
         Attributes attributes) throws SAXException {
```

```
		if (qName.equalsIgnoreCase("title-group")) {
			isNewArticle = true;
		}

		if (qName.equalsIgnoreCase("article-title") && isNewArticle) {
			isNewArticle2 = true;
			isNewArticle = false;
		}

		if (qName.equalsIgnoreCase("ref-list")) {
			isRefList = true;
			refStart = true;
		}

		if (isRefList == true){

			if (qName.equalsIgnoreCase("label")) {
				numOfRef++;
				System.out.print(numOfRef+". ");
			}

			if (qName.equalsIgnoreCase("article-title")) {
				isArticleTitle = true;
			}

		}

		if (isFirstTime == false){

			if (qName.equalsIgnoreCase("count")) {
				isCount = true;
				//flag = false;
			}
		}

		if (isFirstTime == false && flag == true){

			if (qName.equalsIgnoreCase("id")) {
				isPMID = true;
			}
		}

	}

	public void endElement(String uri, String localName,
		String qName) throws SAXException {

		//System.out.println("End Element :" + qName);

	}
```

```
	@SuppressWarnings("deprecation")
	public void characters(char ch[], int start, int length) throws SAXException {

		if (isNewArticle2) {
			String text = new String(ch, start, length);
			System.out.println(text);
			isNewArticle2 = false;

			String encodedText = URLEncoder.encode(text);
			System.out.println("Encoded Text: " + encodedText);


			String encodedURL =
"http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=" +
encodedText;
			System.out.println("URL: " + encodedURL);
			System.out.println("");


			PrintWriter writer;
			String xmlString = TrendMining.getHTML(encodedURL);

			try {
				writer = new
PrintWriter("C:\\Users\\Costas\\workspace\\TrendMining\\ReferecesInXML\\" +
"newArticle.xml", "UTF-8");
				writer.println(xmlString);
				writer.close();
			} catch (FileNotFoundException | UnsupportedEncodingException e)
{
				// TODO Auto-generated catch block
				e.printStackTrace();
			}

			System.out.println(xmlString);
			System.out.println("");
			System.out.println("");
		}

		if (isRefList && refStart) {
			String text = new String(ch, start, length);
			System.out.println("Reference List starts here : " + text);
			refStart = false;
			//refList = false;
		}

		if (isRefList && isArticleTitle) {
			String text = new String(ch, start, length);
			System.out.println(text);
			isArticleTitle = false;
```

```
                        String encodedText = URLEncoder.encode(text);
                        System.out.println("Encoded Text: " + encodedText);


                        String encodedURL =
"http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=" +
encodedText;
                        System.out.println("URL: " + encodedURL);
                        System.out.println("");


                        PrintWriter writer;
                        String xmlString = TrendMining.getHTML(encodedURL);

                        try {
                                writer = new
PrintWriter("C:\\Users\\Costas\\workspace\\TrendMining\\ReferecesInXML\\" + numOfRef
+ ".xml", "UTF-8");
                                writer.println(xmlString);
                                writer.close();
                        } catch (FileNotFoundException | UnsupportedEncodingException e)
{
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }

                        System.out.println(xmlString);
                        System.out.println("");
                        System.out.println("");
                }

                if (isCount && !isFirstTime) {
                        String text = new String(ch, start, length);

                        if (text.compareToIgnoreCase("1") == 0) {
                                flag = true;
                        }

                        if (flag && isPMID){
                                PrintWriter writer;
                                try {
                                        writer = new
PrintWriter("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces\\" + j + ".txt",
"UTF-8");
                                        writer.println(text);
                                        j++;
                                        writer.close();
                                        flag = false;
                                        isPMID = false;
                                } catch (FileNotFoundException |
UnsupportedEncodingException e) {
```

```
                                        // TODO Auto-generated catch block
                                        e.printStackTrace();
                                }
                        }

                }

        }

    };

    if (isFirstTime){
        saxParser.parse(XMLPath + fileName, handler);
    } else {
        saxParser.parse(XMLPath + "newArticle.xml", handler);
        for (int i=1; i <= numOfRef ; i++){
            try {
                saxParser.parse(XMLPath + i + ".xml", handler);
                System.out.println(XMLPath + i + ".xml");
            } catch (FileNotFoundException e){
                e.printStackTrace();
            }
        }
    }

    } catch (Exception e) {
      e.printStackTrace();
    }
   }
}
```

InputToGraph.java

```
package trendMining;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;


public class InputToGraph {

	/**
	 * @param args
	 */
	public static void main(String[] args) {

		File dir = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces");
		boolean isFirst = true;
		String articlePMID = "";

		for (String fn : dir.list()) {
			if (fn == "newArticle.txt"){
				FileInputStream fstream;
				try {
					fstream = new
FileInputStream("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces\\" + fn);
				  DataInputStream in = new DataInputStream(fstream);
				  BufferedReader br = new BufferedReader(new
InputStreamReader(in));
				  String strLine;

				  if ((strLine = br.readLine()) != null){
					articlePMID = strLine;
				  }
				  in.close();
				} catch (FileNotFoundException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}
			}
		}
```

```
                for (String fn : dir.list()) {
                        System.out.println(fn);
                   FileInputStream fstream;
                        try {
                                fstream = new
FileInputStream("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces\\" + fn);
                           DataInputStream in = new DataInputStream(fstream);
                           BufferedReader br = new BufferedReader(new
InputStreamReader(in));
                           String strLine;


                           while ((strLine = br.readLine()) != null){
                                   System.out.println (strLine);

                                   if (isFirst){
                                          articlePMID = strLine;
                                          isFirst = false;
                                   } else {
                                             FileWriter input = new
FileWriter("C:\\Users\\Costas\\workspace\\TrendMining\\InputGraph\\input.txt", true);
                                             input.append(articlePMID);
                                             input.append(" ");
                                             input.append(strLine);
                                             input.append("\r\n");
                                             input.close();
                                   }
                           }
                           in.close();




//FileUtils.cleanDirectory("C:\\Users\\Costas\\workspace\\TrendMining\\PMIDRefereces\\"
);

                           //File file2 = new
File("C:\\Users\\Costas\\workspace\\TrendMining\\PIMDReferences\\" + fn);
                                   //file2.delete();

                        } catch (FileNotFoundException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                }
        }
}
```

Graph.java

```
package trendMining;

import java.util.*;

public abstract class Graph<V extends AbstractVertex<E>, E extends AbstractEdge<V>> {

   private final TreeMap<Integer, V> vertices = new TreeMap<Integer, V>(
         new Comparator<Integer>() {
            //for pretty printing
            @Override
            public int compare( Integer arg0, Integer arg1 ) {
               return arg0.compareTo( arg1 );
            }
         } );

   //need list for random access
   private final List<E> edges = new ArrayList<E>();

   private VertexFactory<V, E> f;

   public Graph( VertexFactory<V, E> f ) {
      if( f == null )
         throw new IllegalArgumentException( "Vertex factory needs to be specified" );
      this.f = f;
   }

   public void addVertex( V v ) {
      vertices.put( v.getLbl(), v );
   }

   public void addEdge( E e ) {
      edges.add( e );
   }

   public V getVertex( int lbl ) {
      V v;
      if ( ( v = vertices.get( lbl ) ) == null ) {
         v = f.newInstance( lbl );
         addVertex( v );
      }
      return v;
   }

   /**
    * @return the vertices
    */
   public Map<Integer, V> getVertices() {
      return vertices;
```

```
    }

    public Map<Integer, V> getVerticesInReversedOrder() {
        return vertices.descendingMap();
    }

    /**
     * @return the edges
     */
    public List<E> getEdges() {
        return edges;
    }

    public void reset() {
        for ( V v : vertices.values() ) {
            v.reset();
        }
    }

}

class UndirectedGraph extends Graph<Vertex, Edge> {

    public UndirectedGraph() {
        super( Vertex.getFactory() );
    }
}

class DirectedGraph extends Graph<DirectedVertex, DirectedEdge> {

    public interface EdgeTraversalPolicy {
        public Set<DirectedEdge> edges( DirectedVertex v );

        public DirectedVertex vertex( DirectedEdge e );
    }

    public final static EdgeTraversalPolicy FORWARD_TRAVERSAL = new EdgeTraversalPolicy()
{

        @Override
        public Set<DirectedEdge> edges( DirectedVertex v ) {
            return v.getOutgoingEdges();
        }

        @Override
        public DirectedVertex vertex( DirectedEdge e ) {
            return e.getHead();
        }
    };
```

```
   public final static EdgeTraversalPolicy BACKWARD_TRAVERSAL = new
EdgeTraversalPolicy() {

      @Override
      public Set<DirectedEdge> edges( DirectedVertex v ) {
         return v.getIncomingEdges();
      }

      @Override
      public DirectedVertex vertex( DirectedEdge e ) {
         return e.getTail();
      }
   };

   public DirectedGraph() {
      super( DirectedVertex.getFactory() );
   }
}

interface VertexFactory<V extends AbstractVertex<E>, E extends AbstractEdge<V>> {
   public V newInstance( int _lbl );
}

class AbstractVertex<E extends AbstractEdge<? extends AbstractVertex<?>>> {

   private final int lbl;
   private final Set<E> edges = new HashSet<E>();

   public AbstractVertex( int lbl ) {
      this.lbl = lbl;
   }

   public void addEdge( E edge ) {
      edges.add( edge );
   }

   public E getEdgeTo( AbstractVertex<E> v2 ) {
      for ( E edge : edges ) {
         if ( edge.contains( this, v2 ) )
            return edge;
      }
      return null;
   }

   /**
    * @return the lbl
    */
   public int getLbl() {
      return lbl;
   }
```

```
   /**
    * @return the edges
    */
   public Set<E> getEdges() {
      return edges;
   }

   public void reset() {}

   @Override
   public String toString() {
      return Integer.toString( getLbl() );
   }
}

class Vertex extends AbstractVertex<Edge> {

   private final static VertexFactory<Vertex, Edge> factory = new VertexFactory<Vertex,
Edge>() {

      @Override
      public Vertex newInstance( int _lbl ) {
         return new Vertex( _lbl );
      }
   };

   public Vertex( int lbl ) {
      super( lbl );
   }

   public static VertexFactory<Vertex, Edge> getFactory() {
      return factory;
   }

}

class Edge extends AbstractEdge<Vertex> {

   public Edge( Vertex fst, Vertex snd ) {
      super( fst, snd );
   }

}

abstract class AbstractEdge<V extends AbstractVertex<? extends AbstractEdge<?>>> {

   private final List<V> ends = new ArrayList<V>();

   public AbstractEdge( V fst, V snd ) {
      if ( fst == null || snd == null ) {
         throw new IllegalArgumentException(
```

```
                "Both vertices are required" );
        }
        ends.add( fst );
        ends.add( snd );
    }

    public boolean contains( AbstractVertex<? extends AbstractEdge<?>> v1,
AbstractVertex<? extends AbstractEdge<?>> v2 ) {
        return ends.contains( v1 ) && ends.contains( v2 );
    }

    public V getOppositeVertex( V v ) {
        if ( !ends.contains( v ) ) {
            throw new IllegalArgumentException( "Vertex " + v.getLbl() );
        }
        return ends.get( 1 - ends.indexOf( v ) );
    }

    public void replaceVertex( V oldV, V newV ) {
        if ( !ends.contains( oldV ) ) {
            throw new IllegalArgumentException( "Vertex " + oldV.getLbl() );
        }
        ends.remove( oldV );
        ends.add( newV );
    }

    public V getFirst() {
        return ends.get( 0 );
    }

    public V getSecond() {
        return ends.get( 1 );
    }
}

class DirectedVertex extends AbstractVertex<DirectedEdge> {

    private final static VertexFactory<DirectedVertex, DirectedEdge> factory = new
VertexFactory<DirectedVertex, DirectedEdge>() {

        @Override
        public DirectedVertex newInstance( int _lbl ) {
            return new DirectedVertex( _lbl );
        }
    };

    private final Set<DirectedEdge> incomingEdges = new HashSet<DirectedEdge>();
    private boolean visited;
    private int f;

    public DirectedVertex( int lbl ) {
```

```
    super( lbl );
}

public static VertexFactory<DirectedVertex, DirectedEdge> getFactory() {
    return factory;
}

public void addIncomingEdge( DirectedEdge e ) {
    incomingEdges.add( e );
}

public void addOutgoingEdge( DirectedEdge e ) {
    super.addEdge( e );
}

//this vertex is head
public Set<DirectedEdge> getIncomingEdges() {
    return incomingEdges;
}

//this vertex is tail
public Set<DirectedEdge> getOutgoingEdges() {
    return super.getEdges();
}

@Override
public Set<DirectedEdge> getEdges() {
    return getOutgoingEdges();
}

/**
 * @return the visited
 */
public boolean isVisited() {
    return visited;
}

/**
 * @param visited the visited to set
 */
public void setVisited( boolean visited ) {
    this.visited = visited;
}

@Override
public void reset() {
    setVisited( false );
}

public void setF( int f ) {
    this.f = f;
```

```
    }

    public int getF() {
        return f;
    }
}

class DirectedEdge extends AbstractEdge<DirectedVertex> {

    public DirectedEdge( DirectedVertex tail, DirectedVertex head ) {
        super( tail, head );
    }

    public DirectedVertex getTail() {
        return getFirst();
    }

    public DirectedVertex getHead() {
        return getSecond();
    }
}
```

KosarajuSCC.java

```
package trendMining;

import java.io.*;
import java.util.*;
import java.util.zip.*;

import trendMining.DirectedGraph.EdgeTraversalPolicy;

/**
 * https://class.coursera.org/algo/quiz/attempt?quiz_id=57
 *
 * Reads the graph directly from the zip file
 */
public class KosarajuSCC {

   private static int t;
   private static ArrayList<Integer> scc = new ArrayList<Integer>();
   private static int pass = 0;

   private static void dfsLoop( DirectedGraph gr, EdgeTraversalPolicy tp ) {
      t = 0;

      Collection<DirectedVertex> vs;
      if( pass == 0 )
         vs = gr.getVerticesInReversedOrder().values();
      else {
         vs = new TreeSet<DirectedVertex>(new Comparator<DirectedVertex>() {
            @Override
            public int compare( DirectedVertex v1, DirectedVertex v2 ) {
               return new Integer( v2.getF() ).compareTo( v1.getF() );
            }
         });
         vs.addAll( gr.getVertices().values() );
      }

      for ( DirectedVertex v : vs ) {
         if( !v.isVisited() ) {

            dfs( tp, v );

            if( pass == 1 ) {
               scc.add( t );
               t = 0;
            }
         }
      }

      pass++;
   }
```

```
private static void dfs( EdgeTraversalPolicy tp, DirectedVertex v ) {

   v.setVisited( true );

   for ( DirectedEdge edge : tp.edges( v ) ) {
      DirectedVertex next = tp.vertex( edge );
      if( !next.isVisited() )
         dfs( tp, next );
   }
   t++;
   if( pass == 0 ) {
      v.setF( t );
   }
}

private static DirectedGraph readGraph( InputStream is ) throws FileNotFoundException {
   Scanner sc = new Scanner( is );
   DirectedGraph gr = new DirectedGraph();
   while( sc.hasNext() ) {
      addEdge( gr, sc.nextInt(), sc.nextInt() );
   }
   sc.close();

   return gr;
}

private static void addEdge( DirectedGraph gr, int tailId, int headId ) {
   DirectedVertex tail = gr.getVertex( tailId );
   DirectedVertex head = gr.getVertex( headId );
   DirectedEdge edge = new DirectedEdge( tail, head );
   gr.addEdge( edge );
   tail.addOutgoingEdge( edge );
   head.addIncomingEdge( edge );
}

private static void test( DirectedGraph gr ) {
   System.out.println("First pass:");
   dfsLoop( gr, DirectedGraph.BACKWARD_TRAVERSAL );

   gr.reset();
   System.out.println("Second pass:");
   dfsLoop( gr, DirectedGraph.FORWARD_TRAVERSAL );

   int count = 0;
   Collections.sort( scc );
   for( int i = scc.size()-1; i >= 0; i-- ) {
      if( count >= 30 ) break;
      System.out.println("ssc:" + scc.get( i ));
      count++;
   }
```

```
    cleanup();
  }

  private static void cleanup() {
    t = 0;
    pass = 0;
    scc.clear();
  }

  private static DirectedGraph example1() {
    DirectedGraph gr = new DirectedGraph();
    addEdge( gr, 1, 2 );
    addEdge( gr, 2, 4 );
    addEdge( gr, 4, 3 );
    addEdge( gr, 3, 1 );
    return gr;
  }

  private static DirectedGraph example2() {
    DirectedGraph gr = new DirectedGraph();
    addEdge( gr, 1, 4 );
    addEdge( gr, 2, 8 );
    addEdge( gr, 3, 6 );
    addEdge( gr, 4, 7 );
    addEdge( gr, 5, 2 );
    addEdge( gr, 6, 9 );
    addEdge( gr, 7, 1 );
    addEdge( gr, 8, 5 );
    addEdge( gr, 8, 6 );
    addEdge( gr, 9, 3 );
    addEdge( gr, 9, 7 );
    return gr;
  }

  private static DirectedGraph example3()
      throws Exception {

    long start = System.currentTimeMillis();

    //File f = new File("C:\\Users\\Costas\\Downloads\\SCC.zip");
    InputStream is = new
FileInputStream("C:\\Users\\Costas\\workspace\\TrendMining\\InputGraph\\input.txt");
    DirectedGraph g = readGraph(is);

    System.out.println( "Read from file: " + ( System.currentTimeMillis() - start ) + " ms");
    System.out.println( "Graph: " + g.getVertices().size() + " vertices, "
        + g.getEdges().size() + " edges." );
    return g;
  }
```

```
    /**
     * @param args
     * @throws IOException
     */
    public static void main( String[] args ) throws Exception {

        //test(example1());
        //test(example2());
        test(example3());
    }

}
```