ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΑΝΙΧΝΕΥΣΗΣ ΔΙΠΛΟΤΥΠΩΝ ΕΓΓΡΑΦΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΤΩΝΟΠΟΥΛΟΥ ΓΑΡΥΦΑΛΙΑ

ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ: ΒΑΣΙΛΕΙΟΣ ΒΕΡΥΚΙΟΣ
ΓΕΩΡΓΙΟΣ ΜΟΥΣΤΑΚΙΔΗΣ

Μάιος 2007

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά :

- Τους επιβλέποντες καθηγητές κ.κ. Β. Βερύκιο και Γ. Μουστακίδη για τη συνεργασία και την υποστήριξη τους σε όλα τα στάδια της εργασίας, για την απαραίτητη καθοδήγηση που μου παρείχαν και τέλος για τις καθοριστικές συμβουλές τους.

- Την οικογένεια μου για τη συνεχή υποστήριξη και συμπαράστασή τους καθ' όλη τη διάρκεια της εργασίας.

Περιεχόμενα

Λίστα Πινάκων

Λίστα Σχημάτων

Αντιστοίχηση Ελληνικής Ορολογίας στην Αγγλική

Ελληνικός Όρος	Αγγλικός Όρος
Αλφαριθμητικό	String
Αναγνωριστής	Identifier
Ανάθεση ετικετών	Tag assignment
Δημιουργία συνόλων δεδομένων	Data set generation
Διασύνδεση εγγραφών	Record Linkage
Διαχωριστικό	Separator
Διπλότυπο	Duplicate
Ευρετήριο	Index
Καθαρισμός και Τυποποίηση εγγραφών	Record Cleaning and Standardization
Κατώφλι	Threshold
Μετατοπιζόμενο παράθυρο	Sliding window
Οντότητα	Entity
Υπό-ρουτίνα	Subroutine
Ταξινομημένη Ευρετηριοποίηση	Sorting Neighborhood
Κατηγοριοποιητής	Classifier
Τμηματοποίηση	Segmentation
Ταίριασμα	Matching
Διάνυσμα βάρους	Weight Vector
Ετερογένεια Δεδομένων	Data Heterogeneity
Απόσταση Σύνταξης	Edit Distance
Απόσταση Χάσματος	Gap Distance
Απόσταση Q-γραμμάτων	Q-grams Distance
Δέσμη ομοειδών	Cluster

Περίληψη

Δεδομένου ότι ο κόσμος γίνεται όλο και περισσότερο αυτοματοποιημένος, οι βάσεις δεδομένων που αποθηκεύουν πληροφορίες έχουν γίνει πολύ σημαντικές. Πολλές κυβερνήσεις καθώς επίσης και πολλές εταιρίες μεμονωμένα κάνουν εκτενή χρήση της τεχνολογίας των βάσεων δεδομένων. Παραδείγματος χάριν, οι βάσεις δεδομένων χρησιμοποιούνται για να αποθηκεύουν ιατρικά δεδομένα, οικονομικές και τραπεζικές πληροφορίες, αριθμούς τηλεφώνου και διευθύνσεις, και πολλές άλλες πληροφορίες.

Μερικές βάσεις δεδομένων είναι τεράστιες. Η διατήρηση των μεγάλων βάσεων δεδομένων μπορεί να είναι δύσκολη, χρονοβόρα και ακριβή. Ένα σημαντικό πρόβλημα που παρατηρείται συχνά σε μεγάλου μεγέθους βάσεις δεδομένων είναι η ύπαρξη των *διπλότυπων* (duplicates) εγγραφών. Αν υποθέσουμε ότι όταν ένας πελάτης που ονομάζεται «Joseph Smith» ξεκινά να συνεργάζεται με μια επιχείρηση, το όνομά του εισάγεται αρχικά στη βάση δεδομένων ως «Joe Smith». Την επόμενη φορά που γίνεται μια παραγγελία από το ίδιο πρόσωπο, ο υπάλληλος πωλήσεων αποτυγχάνει να παρατηρήσει ή να αναγνωρίσει ότι είναι ο ίδιος «Joe Smith» που είναι ήδη στη βάση δεδομένων, και δημιουργεί μια νέα εγγραφή με το όνομα «Joseph Smith». Μια επιπλέον συναλλαγή μπορεί να δημιουργήσει μια νέα εγγραφή «J. Smith». Όταν η επιχείρηση στείλει ένα μήνυμα αλληλογραφίας σε όλους τους πελάτες της, ο κος Smith θα λάβει τρία αντίγραφα ένα προς τον «Joe Smith», άλλο που απευθύνεται στον «Joseph Smith», και ένα τρίτο στον «J. Smith».

Είναι δυνατό να προγραμματιστεί ένας υπολογιστής ώστε να αναγνωρίζει τις εγγραφές που είναι ακριβή αντίγραφα και να τις απορρίπτει. Εντούτοις, στο παραπάνω παράδειγμα, οι εγγραφές δεν είναι ακριβή αντίγραφα, αλλά άντ' αυτού διαφέρουν σε κάποια σημεία. Είναι δύσκολο για τον υπολογιστή αυτόματα να καθορίσει εάν οι εγγραφές είναι πράγματι διπλότυπες. Παραδείγματος χάριν, η εγγραφή «J Smith» μπορεί να αντιστοιχεί σε Joe Smith, ή να αντιστοιχεί στην έφηβη κόρη του, Jane Smith Joe, που ζει στην ίδια διεύθυνση. Η Jane Smith δεν θα πάρει ποτέ το αντίγραφο του μηνύματος εάν ο υπολογιστής είναι προγραμματισμένος να διαγράψει όλα τα αντίγραφα εκτός από ένα «J_Smith». Τα λάθη εισαγωγής δεδομένων, όπως για παράδειγμα τα ορθογραφικά, μπορούν να προκαλέσουν ακόμα χειρότερα προβλήματα ανίχνευσης διπλοτύπων.

Υπάρχουν καταστάσεις στις οποίες διαφορετικές εγγραφές πρέπει να συνδεθούν ή να ταιριάξουν. Παραδείγματος χάριν, υποθέστε ότι ο κος Smith έχει ένα αυτοκινητιστικό ατύχημα και αποθηκεύεται στη βάση δεδομένων μια ασφαλιστική διεκδίκηση με το πλήρες όνομά του «Joseph Smith». Ας υποθέσουμε ότι αρχειοθετείται αργότερα μια δεύτερη διεκδίκηση για ένα άλλο ατύχημα με το όνομα «J. P. Smith». Θα ήταν χρήσιμο ένας υπολογιστής να μπορούσε αυτόματα να ταιριάξει τις δύο διαφορετικές εγγραφές διεκδίκησης εξασφαλίζοντας ότι ο κος Smith δεν προσπαθεί ψευδώς να πάρει διπλή αποζημίωση για το ίδιο ατύχημα.

Η διαδικασία της *διασύνδεσης εγγραφών* (record linkage) είναι μια επίπονη διαδικασία για μεγάλες βάσεις δεδομένων. Ένας τρόπος να βελτιωθεί αυτή η διαδικασία όσον αφορά το *ταίριασμα* (matching) των εγγραφών είναι το αποκαλούμενο «*blocking*» βήμα. Το blocking βήμα αναφέρεται γενικά σε έναν γρήγορο αλγόριθμο ταιριάσματος εγγραφών που χρησιμοποιείται ως ένα βήμα προ-επεξεργασίας της διασύνδεσης εγγραφών. Ο στόχος του blocking βήματος είναι να βρεθούν όλες οι πιθανές αντιστοιχίες μιας εγγραφής χωρίς να δίνεται μεγάλη σημασία στον καθορισμό της σωστής αντιστοιχίας. Αυτές οι εγγραφές

μπορούν έπειτα να εξεταστούν από τον άνθρωπο για την εύρεση της σωστής αντιστοιχίας, ή μπορούν να δηλωθούν αυτόματα ως διπλότυπες, εάν ο χρήστης δεν απαιτεί μεγάλη ακρίβεια.

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικό Υπόβαθρο

Οι βάσεις δεδομένων διαδραματίζουν έναν σημαντικό ρόλο στη σημερινή οικονομία. Πολλά συστήματα, που χρησιμοποιούνται κυρίως σε διάφορες βιομηχανίες, εξαρτώνται από την ακρίβεια των βάσεων δεδομένων για να φέρουν σε πέρας διάφορες καθημερινές διαδικασίες. Επομένως, η ποιότητα των πληροφοριών που αποθηκεύονται στις βάσεις δεδομένων, μπορεί να έχει σημαντικές επιπτώσεις σε ένα σύστημα που στηρίζεται στις πληροφορίες για να λειτουργήσει. Σε ένα σύστημα με τελείως καθαρά δεδομένα, η κατασκευή μιας περιεκτικής δομής των δεδομένων αποτελείται από τη σύνδεση -ένωση- των δεδομένων με βάση ένα χαρακτηριστικό τους, που αποτελεί το κλειδί. Δυστυχώς, τα δεδομένα στερούνται συχνά ενός μοναδικού, σφαιρικού προσδιοριστικού που θα επέτρεπε μια τέτοια λειτουργία. Επιπλέον, τα δεδομένα ούτε προσεκτικά ελέγχονται για την ποιότητα τους ούτε ορίζονται με έναν συνεπή τρόπο μέσα στις διαφορετικές πηγές δεδομένων. Κατά συνέπεια, η ποιότητα των δεδομένων συμβιβάζεται συχνά με πολλούς παράγοντες, συμπεριλαμβανομένων των λαθών εισαγωγής δεδομένων (π.χ., Microsft αντί Microsoft), της έλλειψης περιορισμών ακεραιότητας (π.χ., επιτρέπουν τις καταχωρήσεις όπως EmployeeAge=567), και των πολλαπλών συμβάσεων για την καταγραφή πληροφοριών (π.χ., 44 W. 4th St αντί 44 West Fourth Street). Μάλιστα σε πολλές βάσεις δεδομένων όχι μόνο οι τιμές, αλλά και η σημασιολογία των δεδομένων μπορούν επίσης να διαφέρουν.

Συχνά, ενσωματώνοντας δεδομένα από διαφορετικές πηγές σε μια αποθήκη δεδομένων δημιουργούνται κάποιες συγκρούσεις. Τέτοια προβλήματα υπάγονται στον όρο *ετερογένεια δεδομένων* (data heterogeneity). Ο *καθαρισμός δεδομένων* (data cleaning, data scrubbing), αναφέρεται στη διαδικασία επίλυσης τέτοιων προβλημάτων προσδιορισμού δεδομένων. Διακρίνουμε δύο τύπους ετερογένειας δεδομένων: *δομική* (structural) και *λεξικολογική* (lexical). Η δομική ετερογένεια εμφανίζεται όταν τα πεδία των εγγραφών δομούνται διαφορετικά, στις διαφορετικές βάσεις δεδομένων. Παραδείγματος χάριν, σε μια βάση δεδομένων, η διεύθυνση ενός πελάτη καταγράφεται σε ένα πεδίο που ονομάζεται, για παράδειγμα, addr, ενώ σε μια άλλη βάση δεδομένων η ίδια πληροφορία είναι αποθηκευμένη σε πολλαπλά πεδία όπως street, city, state, και zipcode. Η λεξικολογική ετερογένεια εμφανίζεται όταν όλες οι εγγραφές που βρίσκονται μέσα στις βάσεις δεδομένων έχουν όμοια πεδία, αλλά χρησιμοποιούν διαφορετική έκφραση για να αναφερθούν στο ίδιο πραγματικό αντικείμενο (π.χ., StreetAddress=44 W. 4th St. αντί StreetAddress=44 West Fourth Street). Το πρόβλημα αυτό που αναφέρουμε ως λεξικολογική ετερογένεια είναι γνωστό ως *διασύνδεση εγγραφών*. Ο όρος διασύνδεση εγγραφών προήλθε από το χώρο της δημόσιας υγείας όταν αρχεία που αφορούσαν μεμονωμένους ασθενείς χρειαζόταν να ενωθούν χρησιμοποιώντας το όνομα, την ημερομηνία γέννησης ή κάποιες άλλες πληροφορίες. Ο στόχος του ταιριάσματος εγγραφών είναι να προσδιοριστούν οι εγγραφές, στις ίδιες ή διαφορετικές βάσεις δεδομένων, που αναφέρονται στην ίδια πραγματική οντότητα.

1.2 Προετοιμασία Δεδομένων

Η ανίχνευση διπλότυπων εγγραφών είναι η διαδικασία της αναγνώρισης διαφορετικών πολλαπλών εγγραφών που αναφέρονται σε μια μοναδική πραγματική οντότητα ή αντικείμενο. Χαρακτηριστικά, η διαδικασία της ανίχνευσης διπλοεγγραφών γίνεται μετά τη διαδικασία της προετοιμασίας των δεδομένων, κατά τη διάρκεια της οποίας, τα δεδομένα αποθηκεύονται με ένα συγκεκριμένο τρόπο στη βάση δεδομένων, επιλύνοντας (τουλάχιστον μερικώς) το πρόβλημα της δομικής ετερογένειας. Το στάδιο προετοιμασίας δεδομένων περιλαμβάνει την *ανάλυση* (parsing) και το *μετασχηματισμό* (transformation) των δεδομένων, καθώς και την *τυποποίηση* (standardization) τους. Αυτά τα βήματα βελτιώνουν την ποιότητα των δεδομένων και τα καθιστούν συγκρίσιμα και έτοιμα για χρήση. Ενώ η προετοιμασία δεδομένων δεν είναι αυτό που θα απασχολήσει τη συγκεκριμένη εργασία, αναφέρονται περιληπτικά τα βήματα που εκτελούνται σε αυτό το στάδιο.

Η ανάλυση είναι το κρίσιμο συστατικό του σταδίου προετοιμασίας των δεδομένων. Η ανάλυση εντοπίζει, αναγνωρίζει και απομονώνει ατομικά στοιχεία δεδομένων μέσα στην πηγή των εγγραφών. Η ανάλυση καθιστά ευκολότερο το να διορθωθούν, να τυποποιηθούν, και να ταιριάξουν τα δεδομένα επειδή επιτρέπει τη σύγκριση μεμονωμένων συστατικών, και όχι σύνθετων και μεγάλων σειρών αλφαριθμητικών από δεδομένα. Παραδείγματος χάριν, η κατάλληλη ανάλυση των συστατικών του ονόματος και της διεύθυνσης είναι ένα κρίσιμο μέρος της διαδικασίας καθαρισμού δεδομένων.

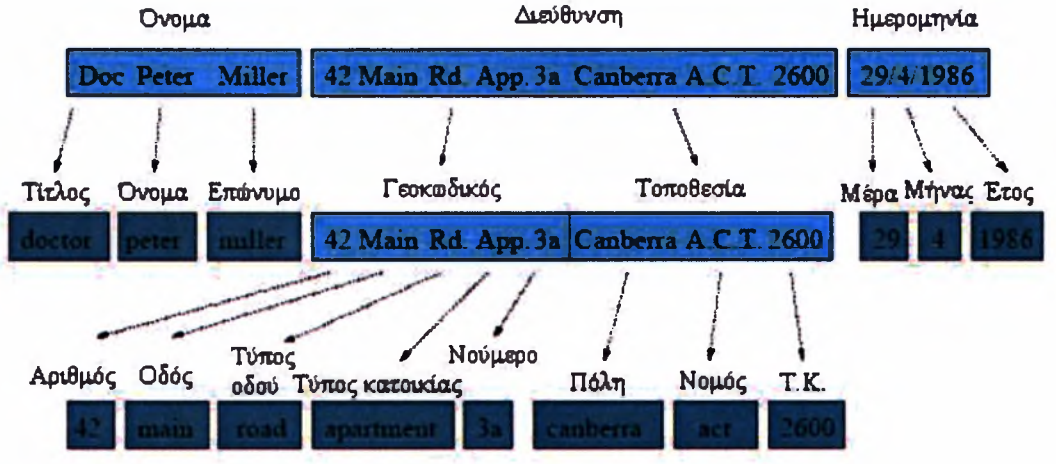
Ο μετασχηματισμός δεδομένων αναφέρεται στις απλές μετατροπές που μπορούν να εφαρμοστούν στα δεδομένα, για να προσαρμοστούν στους τύπους δεδομένων των αντίστοιχων πεδίων τους. Με άλλα λόγια, αυτός ο τύπος μετατροπής εστιάζεται στο χειρισμό ενός πεδίου τη φορά, χωρίς να λαμβάνονται υπόψη οι τιμές σε σχετικά πεδία. Η πιο κοινή μορφή ενός απλού μετασχηματισμού είναι η μετατροπή ενός τύπου δεδομένων σε ένα άλλο. Ένας τέτοιος τύπος μετατροπής δεδομένων απαιτείται συνήθως, όταν μια εφαρμογή κληρονομιών αποθήκευσε δεδομένα σε ένα τύπο δεδομένων που έχει νόημα μέσα στο πλαίσιο της αρχικής εφαρμογής, αλλά όχι σε ένα επόμενο σύστημα. Η μετονομασία ενός πεδίου από ένα όνομα σε ένα άλλο θεωρείται επίσης μετασχηματισμός δεδομένων.

Η τυποποίηση των δεδομένων αναφέρεται στη διαδικασία της μορφοποίησης των πληροφοριών που αναπαρίστανται σε ορισμένα πεδία, με ένα συγκεκριμένο σχήμα περιεχομένου. Αυτό χρησιμοποιείται για τις πληροφορίες που μπορούν να αποθηκευτούν με πολλούς τρόπους στις διάφορες πηγές δεδομένων, είναι απαρχαιωμένες, τους λείπουν ορισμένα αντικείμενα ή περιέχουν λάθη, και πρέπει να έχουν μια ομοιόμορφη αναπαράσταση πριν αρχίσει η διαδικασία ανίχνευσης διπλοεγγραφών. Χωρίς τυποποίηση, πολλές διπλές καταχωρήσεις θα μπορούσαν λανθασμένα να θεωρηθούν ως μη-διπλές.

Χαρακτηριστικό παράδειγμα όπου η τυποποίηση κρίνεται απαραίτητη, αποτελεί η περίπτωση όπου σε μια βάση δεδομένων καταγράφονται στο ίδιο πεδίο το όνομα και το επώνυμο ενός ατόμου, ενώ σε μια άλλη καταχωρούνται σε δυο διαφορετικά πεδία. Μια ακόμα εφαρμογή τυποποίησης, αφορά τις πληροφορίες διευθύνσεων. Δεν υπάρχει κανένας τυποποιημένος τρόπος να αποθηκευτούν οι διευθύνσεις, έτσι η ίδια διεύθυνση μπορεί να αντιπροσωπευθεί με πολλούς διαφορετικούς τρόπους. Η τυποποίηση διευθύνσεων εντοπίζει (χρησιμοποιώντας τις διάφορες τεχνικές ανάλυσης) πληροφορίες όπως αριθμοί σπιτιών, ονόματα οδών, γραμματοκιβώτια, αριθμοί διαμερισμάτων, οι οποίες καταγράφονται μέσα στη βάση δεδομένων χρησιμοποιώντας ένα τυποποιημένο σχήμα (π.χ., 44 West Fourth Street αποθηκεύεται ως 44 W4th St.). Η μορφοποίηση ημερομηνίας και η μορφοποίηση ονόματος

θέτουν άλλες δυσκολίες τυποποίησης σε μια βάση δεδομένων. Επειδή τα περισσότερα λειτουργικά περιβάλλοντα έχουν πολλά διαφορετικά σχήματα για την αντιπροσώπευση των ημερομηνιών, υπάρχει η ανάγκη να μετασχηματιστούν οι ημερομηνίες σε ένα τυποποιημένο σχήμα. Η τυποποίηση ονόματος αναγνωρίζει πληροφορίες όπως είναι τα μικρά ονόματα, τα επώνυμα και τα αρχικά και τις καταγράφει όλες χρησιμοποιώντας κάποια τυποποιημένη σύμβαση. Η τυποποίηση δεδομένων είναι ένα μάλλον ανέξοδο βήμα που μπορεί να οδηγήσει στη γρήγορη αναγνώριση των αντιγράφων. Παραδείγματος χάριν, εάν η μόνη διαφορά μεταξύ δύο εγγραφών είναι η διαφορετικά καταχωρημένη διεύθυνση (44 West Fourth Street αντί 44 W4th St.), το βήμα τυποποίησης δεδομένων θα έκανε τις δύο εγγραφές ίδιες.

Στο Σχήμα 1, φαίνεται ο τρόπος με τον οποίο η διαδικασία της τυποποίησης επιδρά σε μια εγγραφή. Ενώ η εγγραφή έχει στην αρχή τρία πεδία, μετατρέπεται κατάλληλα ώστε να έχει είτε οκτώ είτε δεκατέσσερα πεδία, ανάλογα με τις εκάστοτε απαιτήσεις. Κατά αυτόν τον τρόπο, η συγκεκριμένη εγγραφή αποκτά τον ίδιο τύπο με μια άλλη που έχει περισσότερα πεδία, και παράλληλα, αυξάνοντας τα πεδία αυξάνεται και η ακρίβεια των αποτελεσμάτων που θα δώσει η διασύνδεση.



Σχήμα 1 – Παράδειγμα Τυποποίησης Προσωπικών Πληροφοριών

1.3 Κεντρική Ιδέα Εργασίας

Στην ενότητα 1.1, έγινε μια μικρή αναφορά στον όρο της διασύνδεσης εγγραφών και με τι ουσιαστικά ασχολείται. Η συγκεκριμένη έρευνα εστιάζεται κυρίως στην εξέταση και τη σύγκριση μεθόδων που χρησιμοποιούνται στη διαδικασία διασύνδεσης εγγραφών για να εξαλειφθούν τα διπλότυπα. Ουσιαστικά εστιάζουμε στο βήμα εκείνο της διασύνδεσης εγγραφών, που σκοπό έχει, τη μείωση των συνολικών συγκρίσεων ζευγαριών εγγραφών προκειμένου να βρεθούν τα διπλότυπα, δηλαδή το blocking. Μελετώνται κάποιες από τις υπάρχουσες Blocking μεθόδους, ενώ γίνονται κάποια πειράματα προκειμένου να συγκριθούν οι μέθοδοι του *Τυποποιημένου Blocking* (Standard Blocking), της *Ταξινομημένης Γειτονιάς* (Sorted Neighborhood) και της *Ευρετηριοποίησης Δι-γραμμάτων* (Bigram Indexing).

1.4 Συνεισφορά της Εργασίας

Η εργασία υποκινήθηκε κυρίως από την ανάγκη να ερευνηθεί ένας τομέας των βάσεων δεδομένων η ανάπτυξη του οποίου δεν βρίσκεται σε ικανοποιητικό στάδιο. Η συνεισφορά λοιπόν της συγκεκριμένης έρευνας είναι δυνατόν να αποτυπωθεί στα παρακάτω σημεία:

- **Λογισμικό Febrl:** παρουσιάζεται η χρήσιμη πλατφόρμα λογισμικού Febrl η οποία αποτελείται από μια βιβλιοθήκη προγραμμάτων που επιτυγχάνουν την τυποποίηση και διασύνδεση των εγγραφών. Διερευνώνται οι δυνατότητες του συγκεκριμένου λογισμικού, ιδιαίτερα σε ότι αφορά το blocking, καθώς και ο τρόπος λειτουργίας του. Γίνονται κάποιες αλλαγές στο πρόγραμμα υλοποίησης της απαλοιφής διπλότυπων εγγραφών που χρησιμοποιεί, με σκοπό την μελέτη των αλγορίθμων blocking που διαθέτει ενώ επισημαίνονται και κάποια σημαντικά στοιχεία που θα μπορούσαν να προστεθούν στην περαιτέρω επέκτασή του.
- **Τεχνικές Blocking:** περιγράφονται αναλυτικά οι μέθοδοι του Τυποποιημένου Blocking, της Ταξινομημένης Γειτονιάς και της Ευρετηριοποίησης Δι-γραμμάτων, συγκρίνονται αρκετά λεπτομερώς και συμπεραίνονται κάποια στοιχεία σχετικά με τη συμπεριφορά τους. Μελετάται η συμπεριφορά τους για διαφορετικά μεγέθη συνόλων δεδομένων, καθώς και για διαφορετικό πλήθος λαθών ανά εγγραφή. Η σύγκρισή τους στηρίζεται ουσιαστικά στη χρήση δύο μέτρων, της *Αναλογίας Μείωσης* (Reduction Ratio) και της *Πληρότητας των Ζευγαριών Εγγραφών* (Pairs Completeness), τα οποία είναι ενδεικτικά της αποτελεσματικότητας τους. Ελέγχεται η συμπεριφορά τους με διαφορετικές παραμέτρους και συμπεραίνεται ποια από τις παραμέτρους προσφέρει τα καλύτερα αποτελέσματα για την κάθε μέθοδο.

1.5 Δομή της Εργασίας

Η δομή της υπόλοιπης εργασίας έχει ως εξής:

- Στο δεύτερο κεφάλαιο παρουσιάζονται διάφορα θέματα που αφορούν την διασύνδεση εγγραφών. Αρχικά γίνεται μια πρώτη αναφορά στο τι ορίζουμε ως διασύνδεση εγγραφών ενώ αναφέρονται οι υπάρχουσες τεχνικές που επιτελούν αυτήν την διαδικασία. Στη συνέχεια, μελετώνται τα βήματα της απαλοιφής διπλοτύπων ένα-ένα και ποιες τεχνικές χρησιμοποιούνται για την υλοποίηση αυτών των βημάτων. Τέλος, αναφέρονται τα κριτήρια που καθορίζουν την αποδοτικότητα της διασύνδεσης εγγραφών και διάφορα ζητήματα ιδιωτικότητας που σχετίζονται με αυτήν τη διαδικασία.
- Στο τρίτο κεφάλαιο περιγράφονται κάποιες τεχνικές Blocking, με ιδιαίτερη έμφαση σε αυτές του Τυποποιημένου Blocking, της Ταξινομημένης Γειτονιάς και της Ευρετηριοποίησης Δι-γραμμάτων. Επιπλέον, μελετώνται κάποιοι φωνητικοί αλγόριθμοι, που χρησιμοποιούνται κυρίως για αλφαριθμητικά που είναι φωνητικά παρόμοια ακόμα και αν δεν μοιάζουν σε επίπεδο χαρακτήρα. Τέλος, αναφέρονται τα κριτήρια πολυπλοκότητας που καθορίζουν την αποδοτικότητα των Blocking μεθόδων ενώ παρουσιάζεται και το φιλτράρισμα, ένα είδος βελτίωσης του βήματος blocking.
- Στο τέταρτο κεφάλαιο γίνεται η παρουσίαση του λογισμικού Febrl. Ουσιαστικά περιγράφονται οι εφαρμογές του που παρουσίασαν ενδιαφέρον κατά τη διάρκεια της

έρευνας και αφορούν τη δημιουργία συνόλων δεδομένων, τον καθαρισμό και την τυποποίηση τους καθώς και τη διασύνδεση των εγγραφών τους. Ιδιαίτερη προσοχή δίνεται στις μεθόδους Blocking που υλοποιεί και οι οποίες συγκρίνονται.

- Στο πέμπτο κεφάλαιο περιγράφονται τα πειράματα που έγιναν αναλυτικά, ενώ γίνεται και μια προσπάθεια αξιολόγησης των αποτελεσμάτων τους κυρίως με την απεικόνιση τους σε γραφικές παραστάσεις. Η αξιολόγηση αφορά τόσο τον χρόνο εκτέλεσης των πειραμάτων όσο και την ποιότητα των αποτελεσμάτων που προσφέρουν. Όσον αφορά την ποιότητα, θα δοθούν κάποια στοιχεία ώστε να φανεί ποια από τις τρεις μεθόδους που συγκρίνονται είναι η καλύτερη.
- Τέλος, στο έκτο κεφάλαιο δίνεται ο επίλογος της εργασίας, όπου παρουσιάζονται τα τελικά συμπεράσματα, καθώς και οι κατευθύνσεις που μπορεί να ακολουθήσει κάποιος που θέλει να επεκτείνει την συγκεκριμένη έρευνα.

Κεφάλαιο 2

Διασύνδεση εγγραφών

2.1 Γενικά

Στην εποχή μας διάφοροι οργανισμοί, συλλέγουν καθημερινά μεγάλα ποσά δεδομένων που απαιτούνται για τη σωστή λειτουργία τους. Όλα αυτά τα δεδομένα δεν προέρχονται μόνο από μια βάση δεδομένων αλλά συχνά ανακτώνται από πολλαπλές πηγές δεδομένων. Προκειμένου λοιπόν, οι πληροφορίες που θα προκύψουν από αυτά τα δεδομένα να μπορούν να χρησιμοποιηθούν σωστά, τα δεδομένα αυτά πρέπει να αναλυθούν λεπτομερώς, να συνδυαστούν και να συνδεθούν με τέτοιο τρόπο ώστε να βρίσκονται σε μια συνεπή μορφή.

Για τη σωστή διαχείριση του μεγάλου όγκου των πληροφοριών, η κοινότητα των Βάσεων Δεδομένων ασχολήθηκε με την δημιουργία ειδικών τεχνικών και μεθόδων ενοποίησης των δεδομένων. Μια από τις μεθόδους αυτές είναι η *διασύνδεση των εγγραφών*. Ουσιαστικά με την *ενοποίηση* των δεδομένων εννοούμε τον συνδυασμό, ή αλλιώς την συνάθροιση των εγγραφών, και γενικότερα των πληροφοριών, που διαθέτουν δυο ή περισσότερες βάσεις δεδομένων από διαφορετικές πηγές. Σκοπός αυτής της διαδικασίας, είναι να εντοπιστούν οι εγγραφές που αντιστοιχούν στην ίδια οντότητα, και στην συνέχεια να απαλειφθούν οι διπλοεγγραφές.

Ένα σημαντικό κομμάτι της ενοποίησης των πληροφοριών είναι το ταίριασμα των αντικειμένων και η συνένωση των δεδομένων. Αποτελεί σύνηθες φαινόμενο, σε διαφορετικές βάσεις δεδομένων, να υπάρχουν αναφορές στο ίδιο αντικείμενο, οι οποίες πολλές φορές είναι δύσκολο να αναγνωριστούν. Η διασύνδεση των εγγραφών είναι μια διαδικασία που είναι σε θέση να οδηγήσει στην απαλοιφή των διπλότυπων από διαφορετικές πηγές.

Η διασύνδεση των εγγραφών είναι βασικά η διαδικασία σύγκρισης εγγραφών που προέρχονται από δύο ή περισσότερες πηγές δεδομένων με σκοπό να αποφασιστεί ποια ζεύγη εγγραφών αναπαριστούν την ίδια οντότητα. Επίσης, μπορεί να οριστεί ως η διαδικασία του εντοπισμού των διπλοεγγραφών σε ένα αρχείο. Το πρόβλημα που υπάρχει στη διασύνδεση των εγγραφών είναι ότι τα δεδομένα που πρόκειται να συνδεθούν δεν είναι καθαρά. Με άλλα λόγια, αν τα δεδομένα ήταν ακριβή, η διασύνδεση εγγραφών θα ήταν μια παρόμοια διαδικασία με αυτή της απαλοιφής των διπλοτύπων. Για παράδειγμα, σε μια βάση δεδομένων πελατών, μια ή περισσότερες εγγραφές είναι πιθανό να αναφέρονται στο ίδιο άτομο εξαιτίας ενός ορθογραφικού λάθους στο όνομα του πελάτη.

2.2 Τεχνικές Διασύνδεσης Εγγραφών

Η διασύνδεση εγγραφών θεωρείται τμήμα της διαδικασίας καθαρισμού δεδομένων. Οι τεχνικές διασύνδεσης εγγραφών χρησιμοποιούνται κυρίως για τη βελτίωση της ποιότητας και της ακεραιότητας των δεδομένων. Συμβάλλουν επίσης, στην εκμετάλλευση πηγών δεδομένων που έχουν χρησιμοποιηθεί στο παρελθόν για νέες μελέτες, καθώς και στη μείωση του κόστους και της προσπάθειας στην απόκτηση δεδομένων για διάφορες έρευνες.

Εάν σε όλα τα σύνολα δεδομένων που πρόκειται να συνενωθούν, υπάρχει ένα κοινό κλειδί αναγνώρισης (αναγνωριστής) για κάθε οντότητα, τότε το πρόβλημα της διασύνδεσης λύνεται

εύκολα με μια απλή *ένωση* (join) των δεδομένων. Στην περίπτωση όμως που δεν υπάρχει αυτό το μοναδικό κλειδί, εφαρμόζονται πολύπλοκες τεχνικές. Ένας πιθανός διαχωρισμός των τεχνικών αυτών, ανάλογα με την μεθοδολογία που ακολουθούν, είναι σε *ντετερμινιστικές* και *πιθανοτικές*.

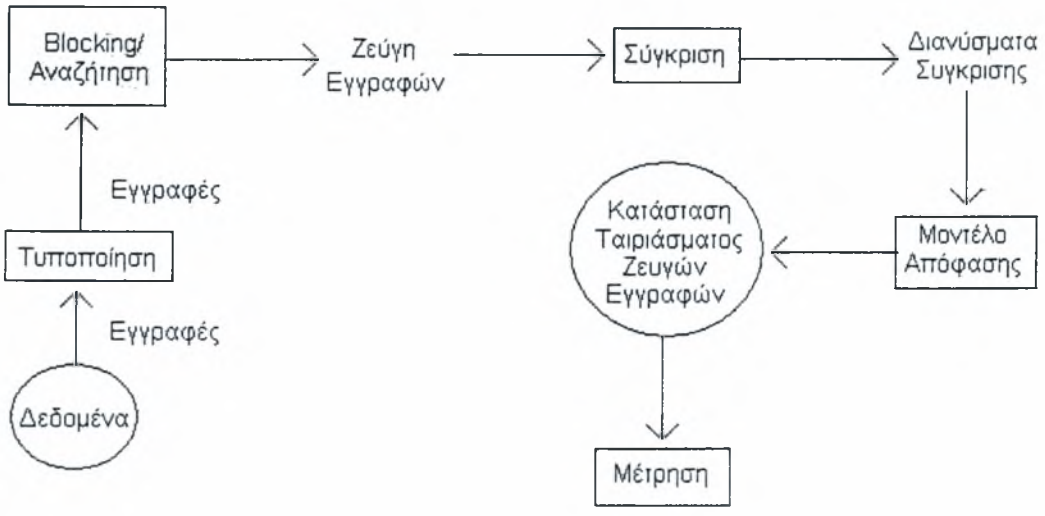
Οι ντετερμινιστικές μέθοδοι έχουν στην διάθεση τους μεγάλα σύνολα, ιδιαίτερα πολύπλοκων κανόνων, που ορίζουν πότε ταιριάζουν οι εγγραφές. Με βάση αυτούς τους κανόνες κατηγοριοποιούν ένα ζευγάρι εγγραφών στα ταιριάζοντα, αν αντιστοιχούν στην ίδια οντότητα, ή στα μη ταιριάζοντα, αν αντιστοιχούν σε διαφορετικές οντότητες. Αντίθετα, οι πιθανοτικές μέθοδοι χρησιμοποιούν στατιστικά μοντέλα για να αποφανθούν για την ομοιότητα των εγγραφών. Χωρίζονται σε δυο κατηγορίες, στις μεθόδους που βασίζονται στην κλασική πιθανοτική θεωρία διασύνδεσης εγγραφών, όπως αυτή επεξηγήθηκε από τους Fellegi και Sunter [17], και σε αυτές που βασίζονται σε νεότερες τεχνικές με χρήση μηχανισμών εκμάθησης [3]. Το 1969, οι Fellegi και Sunter ήταν οι πρώτοι που παρουσίασαν το μαθηματικό μοντέλο για τη διασύνδεση εγγραφών, ακολουθώντας ένα μεγάλο πλήθος μελετών που δημοσιεύονταν από το 1959. Το μοντέλο που προτάθηκε από τους Fellegi και Sunter χαρακτηρίζεται σαν ένα πιθανοτικό μοντέλο εφόσον βασίζεται ολοκληρωτικά στη θεωρία πιθανοτήτων.

Κάθε φορά που πρόκειται να συνδεθούν κάποιες εγγραφές, ανεξάρτητα από την τεχνική που θα χρησιμοποιηθεί, πρέπει να διευθετούνται κάποια ζητήματα. Για να αντιμετωπιστούν λάθη σε οντότητες συνόλων δεδομένων ή τυπογραφικά λάθη, χρησιμοποιούνται συναρτήσεις που κάνουν κατά προσέγγιση συγκρίσεις συμβολοσειρών καθώς και συγκρίσεις αριθμητικών δεδομένων. Για πολύ μεγάλες ομάδες δεδομένων, εκατομμύρια εγγραφών, εφαρμόζονται ειδικές τεχνικές απαραίτητες για να χειριστούν δεδομένα τέτοιου μεγέθους.

2.3 Απαλοιφή Διπλοτύπων

Η διασύνδεση εγγραφών είναι η διαδικασία της σύγκρισης εγγραφών και της απόφασης του ταιριάσματος τους (εάν αναπαριστούν την ίδια οντότητα) ή του μη-ταιριάσματος τους (εάν αναπαριστούν διαφορετικές οντότητες). Είναι πιθανό κάποιες φορές η απόφαση αυτή να παρθεί χρησιμοποιώντας την παρέμβαση του ανθρώπου. Δεδομένου ότι οι εγγραφές είναι καθαρές και τυποποιημένες, η διαδικασία της σύνδεσης εγγραφών ή της απαλοιφής διπλότυπων από ομάδες δεδομένων, αποτελείται από τα ακόλουθα βήματα [1].

- Ένα ή περισσότερα ευρετήρια κατασκευάζονται με σκοπό την ομαδοποίηση των εγγραφών που ενδεχομένως ταιριάζουν, οπότε μειώνεται και ο τεράστιος αριθμός των πιθανών συγκρίσεων.
- Οι εγγραφές που ανήκουν στο ίδιο ευρετήριο συγκρίνονται πεδίο προς πεδίο, χρησιμοποιώντας κατάλληλες συναρτήσεις, καταλήγοντας σε ένα διάνυσμα βαρύτητας για κάθε ζευγάρι εγγραφών που συγκρίνεται.
- Αυτά τα διανύσματα βαρύτητας δίνονται σε ένα κατηγοριοποιητή που αποφασίζει εάν ένα ζευγάρι εγγραφών ταιριάζει, δεν ταιριάζει ή είναι πιθανόν να ταιριάζει.



Σχήμα 2 - Διαδικασία διασύνδεσης εγγραφών

2.3.1 Ευρετηριοποίηση

Σκοπός της ευρετηριοποίησης, είναι η μείωση του τεράστιου αριθμού πιθανών συγκρίσεων, εξαλείφοντας συγκρίσεις ανάμεσα σε εγγραφές που είναι φανερό ότι δεν ταιριάζουν. Είναι σημαντικό να τονισθεί ότι, όταν συγκρίνονται εγγραφές δύο διαφορετικών συνόλων δεδομένων, πρέπει να χρησιμοποιείται η ίδια ακριβώς μέθοδος για τη δημιουργία των ευρετηρίων και των δύο συνόλων, αλλιώς η σύγκριση τους θα είναι αδύνατη.

2.3.2 Συναρτήσεις Σύγκρισης Πεδίων

Αφού οι εγγραφές οργανωθούν σε ευρετήρια, τότε γίνεται η σύγκριση τους, διαδικασία που αποτελεί το σημαντικότερο κομμάτι της διασύνδεσης εγγραφών. Οι συναρτήσεις σύγκρισης πεδίων ελέγχουν ξεχωριστά τα πεδία των εγγραφών, και ανάλογα με την συνάρτηση που χρησιμοποιείται κάθε φορά, συγκρίνονται αλφαριθμητικά, αριθμοί, ημερομηνίες, ηλικίες και χρονικές περίοδοι. Αυτές οι συναρτήσεις επιστρέφουν τα βασικά βάρη ταιριάσματος για κάθε ζευγάρι εγγραφών που συγκρίνεται, τα οποία αποθηκεύονται σε ένα *διάνυσμα βάρους* (weight vector). Πολλαπλές μέθοδοι έχουν αναπτυχθεί γι' αυτόν το στόχο, και κάθε μέθοδος λειτουργεί καλά για συγκεκριμένους τύπους λαθών. Έπειτα, δίνονται σε ένα κατηγοριοποιητή, που υπολογίζει αν οι εγγραφές ταιριάζουν ή όχι. Σε αυτό το σημείο, γίνεται μια μικρή αναφορά στις τεχνικές που έχουν εφαρμοστεί για το ταίριασμα πεδίων με αλφαριθμητικά δεδομένα καθώς και με αριθμητικά δεδομένα.

2.3.2.1 Τεχνικές βασισμένες στην ομοιότητα χαρακτήρων

Οι τεχνικές βασισμένες στην ομοιότητα χαρακτήρων [3] έχουν ως σκοπό να χειριστούν καλά τα τυπογραφικά λάθη. Μερικές τέτοιες τεχνικές είναι οι: *απόσταση σύνταξης* (edit-distance), *απόσταση χάσματος* (gap-distance), *απόσταση Smith-Waterman*, *απόσταση Jaro*, και *απόσταση q-γραμμάτων* (q-grams distance).

Η απόσταση σύνταξης μεταξύ δύο αλφαριθμητικών σ_1 και σ_2 είναι ο ελάχιστος αριθμός τροποποιήσεων στη σύνταξη των χαρακτήρων που χρειάζονται, για να μετασχηματιστεί το αλφαριθμητικό σ_1 σε σ_2. Υπάρχουν τρεις τύποι διαδικασιών σύνταξης:

- παρεμβολή ενός χαρακτήρα στο αλφαριθμητικό,
- διαγραφή ενός χαρακτήρα από το αλφαριθμητικό, και
- αντικατάσταση ενός χαρακτήρα με έναν διαφορετικό χαρακτήρα.

Η απόσταση σύνταξης δεν εργάζεται καλά όταν πρέπει να ταιριάξει αλφαριθμητικά που έχουν περικοπεί, ή μικρύνει (π.χ., «John R. Smith» αντί «Jonathan Richard Smith»). Η απόσταση χάσματος προσφέρει μια λύση σε αυτό το πρόβλημα με την εισαγωγή δύο πρόσθετων διαδικασιών σύνταξης: το *ανοικτό χάσμα* και το *επεκτεινόμενο χάσμα*.

Οι Smith και Waterman περιέγραψαν μια επέκταση της απόστασης σύνταξης και της απόστασης χάσματος, στην οποία λανθασμένα ταιριάσματα στην αρχή και το τέλος των αλφαριθμητικών έχουν χαμηλότερο κόστος από τα λανθασμένα ταιριάσματα στη μέση. Αυτή η τεχνική επιτρέπει καλύτερη τοπική συμμόρφωση των αλφαριθμητικών (δηλαδή, ταίριασμα υπο-αλφαριθμητικών). Επομένως, τα αλφαριθμητικά «Prof. John R. Smith, University of Calgary» και «John R. Smith, Prof.» μπορούν να ταιριάξουν με μικρό κόστος χρησιμοποιώντας την απόσταση Smith-Waterman, δεδομένου ότι τα προθέματα και οι καταλήξεις αγνοούνται.

Ο Jaro εισήγαγε έναν αλγόριθμο σύγκρισης αλφαριθμητικών που τον χρησιμοποιούσαν κυρίως για τη σύγκριση των επωνύμων και των ονομάτων. Στην τεχνική Jaro για δύο αλφαριθμητικά σ_1 και σ_2 υπολογίζουμε αρχικά τα μήκη τους, βρίσκουμε τους κοινούς χαρακτήρες στα δύο αλφαριθμητικά και τέλος, υπολογίζουμε τον αριθμό μεταθέσεων. Ο αριθμός μεταθέσεων υπολογίζεται ως εξής: συγκρίνουμε τον πρώτο, δεύτερο κ.τ.λ. κοινό χαρακτήρα του σ_1 με τον αντίστοιχο κοινό χαρακτήρα του σ_2 και κάθε χαρακτήρας που δεν ταιριάζει είναι μια μετάθεση.

Τα q-γράμματα είναι σύντομοι χαρακτήρες μήκους q, των αλφαριθμητικών της βάσης δεδομένων. Όταν δυο αλφαριθμητικά είναι όμοια μοιράζονται έναν μεγάλο αριθμό q-γραμμάτων από κοινού. Λαμβάνοντας υπόψη ένα αλφαριθμητικό σ, τα q-γράμματα του λαμβάνονται με την ολίσθηση ενός παραθύρου μήκους q πάνω από τους χαρακτήρες του σ.

2.3.2.2 Τεχνικές βασισμένες στην ομοιότητα συμβόλων

Οι τεχνικές βασισμένες στην ομοιότητα χαρακτήρων, όπως ήδη αναφέρθηκε [3], δουλεύουν καλά για τα τυπογραφικά λάθη. Εντούτοις, συχνά τυπογραφικές συμβάσεις οδηγούν στην αλλαγή της θέσης των λέξεων (π.χ., «John Smith» αντί «Smith, John»). Σε τέτοιες περιπτώσεις, οι τεχνικές βασισμένες στην ομοιότητα χαρακτήρων αποτυγχάνουν να εντοπίσουν την ομοιότητα των οντοτήτων. Οι τεχνικές βασισμένες στην ομοιότητα συμβόλων προσπαθούν να αντισταθμίσουν αυτό το πρόβλημα. Δύο τέτοιες τεχνικές είναι τα *ατομικά αλφαριθμητικά* (atomic string) και το *WHIRL*.

Ένα ατομικό αλφαριθμητικό είναι μια ακολουθία αλφαβητικών χαρακτήρων και αριθμών οριοθετημένη από τους χαρακτήρες στίξης. Δύο ατομικά αλφαριθμητικά ταιριάζουν εάν είναι ίσα, ή εάν το ένα είναι το πρόθεμα του άλλου. Με βάση αυτόν τον αλγόριθμο, η ομοιότητα δύο πεδίων, είναι ο αριθμός που προκύπτει από τη διαίρεση των ατομικών τους αλφαριθμητικών ταιριάσματος, με το μέσο αριθμό των ατομικών αλφαριθμητικών τους.

Το σύστημα WHIRL πήρε το όνομα του από το συνδυασμό της τεχνικής ομοιότητας *συνημιτόνου* με το σχέδιο στάθμισης *tf.idf*, που χρησιμοποιείται για να υπολογίσει την ομοιότητα δύο πεδίων. Η μετρική ομοιότητας συνημιτόνου δουλεύει καλά για μια μεγάλη ποικιλία καταχωρήσεων, και δεν επηρεάζεται από τη θέση των λέξεων, επιτρέποντας κατά συνέπεια κινήσεις και ανταλλαγές λέξεων (π.χ., το «John Smith» αντιστοιχεί με το «Smith, John»). Επίσης, η εισαγωγή λέξεων που εμφανίζονται συχνά, επηρεάζει μόνο ελάχιστα την ομοιότητα των δύο αλφαριθμητικών, λόγω του χαμηλού βάρους *idf* των λέξεων που εμφανίζονται συχνά. Παραδείγματος χάριν, τα John Smith και Mr. John Smith, θα είχαν ομοιότητα κοντά στο 1. Δυστυχώς, αυτή η τεχνική ομοιότητας δεν συλλαμβάνει τα λάθη ορθογραφίας, ειδικά εάν είναι κυρίαρχα και επηρεάζουν πολλές από τις λέξεις στα αλφαριθμητικά. Παραδείγματος χάριν, τα "Compter Science Department" και "Deprtment of Computer Scence" θα έχουν 0 ομοιότητα σύμφωνα με αυτή την τεχνική.

2.3.2.3 Τεχνικές φωνητικής ομοιότητας

Τα αλφαριθμητικά μπορούν να είναι φωνητικά παρόμοια ακόμα κι αν δεν είναι παρόμοια σε επίπεδο χαρακτήρα ή συμβόλων. Παραδείγματος χάριν, η λέξη Kageonne είναι φωνητικά παρόμοια με την Cajun παρά το γεγονός ότι οι αναπαραστάσεις αλφαριθμητικών είναι πολύ διαφορετικές. Οι τεχνικές φωνητικής ομοιότητας [3] προσπαθούν να αντιμετωπίσουν τέτοια ζητήματα και να ταιριάξουν τέτοια αλφαριθμητικά. Υπάρχουν πολλές τέτοιες τεχνικές οι οποίες περιγράφονται αναλυτικά στην παράγραφο 3.3 της εργασίας.

2.3.2.4 Αριθμητικές τεχνικές ομοιότητας

Ενώ πολλαπλές μέθοδοι υπάρχουν για την ανίχνευση των ομοιοτήτων των δεδομένων βασισμένων σε αλφαριθμητικά, οι μέθοδοι για την εύρεση ομοιοτήτων στα αριθμητικά δεδομένα [3] είναι λιγότερες. Χαρακτηριστικά, οι αριθμοί αντιμετωπίζονται ως αλφαριθμητικά και επεξεργάζονται με τις παραπάνω τεχνικές.

2.3.3 Κατηγοριοποίηση

Το τελευταίο βήμα στην διαδικασία διασύνδεσης δεδομένων, αφού πρώτα οι εγγραφές έχουν συγκριθεί και τα διανύσματα βάρους έχουν υπολογισθεί, είναι η *κατηγοριοποίηση* (classification) των εγγραφών με βάση την ομοιότητα τους [1].

2.4 Κριτήρια Απόδοσης της Διασύνδεσης Εγγραφών

Η ποιότητα της διασύνδεσης εγγραφών εξαρτάται από τις ακόλουθες παραμέτρους [5,22,25,26]:

- Ο αριθμός των ζευγαριών εγγραφών που ταιριάζουν και σωστά συνδέονται (true positives) n_m.
- Ο αριθμός των ζευγαριών εγγραφών που αν και δεν ταιριάζουν συνδέονται (false positives) n_{fp}.
- Ο αριθμός των ζευγαριών εγγραφών που δεν ταιριάζουν και σωστά δεν συνδέονται (true negatives) n_u.
- Ο αριθμός των ζευγαριών εγγραφών που ενώ ταιριάζουν δεν συνδέονται (false negatives) n_{fn}.

Για αυτές τις παραμέτρους ισχύουν και οι ακόλουθες σχέσεις $N_m=n_m+n_{fn}$, $N_u=n_u+n_{fp}$, $\sim m=n_m+n_{fp}$, $\sim u=n_u+n_{fn}$ όπου ως N_m εννοούμε τα ζευγάρια εγγραφών που πραγματικά ταιριάζουν, ως N_u τα ζευγάρια εγγραφών που πραγματικά δεν ταιριάζουν, ως $\sim m$ τα ζευγάρια εγγραφών που σύμφωνα με τον κατηγοριοποιητή ταιριάζουν και ως $\sim u$ τα ζευγάρια εγγραφών που σύμφωνα με τον κατηγοριοποιητή δεν ταιριάζουν.

Μαζί με τους δύο βασικούς αριθμούς (N_m και N_u) και τις παραπάνω παραμέτρους, ποικίλα μέτρα μπορούν να οριστούν. Μερικά από αυτά τα μέτρα είναι τα ακόλουθα:

- *Ευαισθησία* (Sensitivity): n_m/N_m, ο αριθμός των σωστά συνδεδεμένων ζευγαριών εγγραφών διαιρεμένος με τον συνολικό αριθμό των σωστά ταιριασμένων ζευγαριών εγγραφών.
- *Ιδιαιτερότητα* (Specificity): n_u/N_u, ο αριθμός των σωστά μη συνδεδεμένων ζευγαριών εγγραφών διαιρεμένος με τον συνολικό αριθμό των σωστά μη ταιριασμένων ζευγαριών εγγραφών.
- *Ρυθμός Ταιριάσματος* (Match Rate): $(n_m+n_{fp})/N_m$, ο συνολικός αριθμός των συνδεδεμένων ζευγαριών εγγραφών διαιρεμένος με τον συνολικό αριθμό των σωστά ταιριασμένων ζευγαριών εγγραφών.
- *Αναμενόμενη Αποδεκτή Τιμή* (Positive Predictive Value (ppv)): $n_m/(n_m+n_{fp})$, ο αριθμός των σωστά συνδεδεμένων ζευγαριών εγγραφών διαιρεμένος με τον συνολικό αριθμό των συνδεδεμένων ζευγαριών εγγραφών.
- *Ορθότητα* (Accuracy): υπολογίζεται ως $acc=(n_m+n_u)/(n_m+n_{fp}+n_u+n_{fn})$. Είναι ένα ευρέως χρησιμοποιούμενο κριτήριο και κυρίως κατάλληλο για προβλήματα κατηγοριοποίησης. Καθώς αυτό το κριτήριο περιέχει τον αριθμό n_u εξαρτάται από το μέγεθος του, δηλαδή το μεγάλο πλήθος των ζευγαριών εγγραφών που δεν ταιριάζουν και σωστά δεν συνδέονται. Οι τιμές του acc θα είναι πολύ υψηλές, αν όλα τα ζευγάρια των εγγραφών που συγκρίθηκαν θεωρήθηκαν μη ταιριάζοντα. Ωστόσο. η ορθότητα δεν είναι ένα καλό κριτήριο ποιότητας της διασύνδεσης εγγραφών και δεν θα πρέπει να χρησιμοποιείται.

Μπορεί να παρατηρηθεί ότι η ευαισθησία μετρά το ποσοστό των σωστά κατηγοριοποιημένων ταιριασμένων εγγραφών, ενώ η ιδιαιτερότητα μετρά το ποσοστό των σωστά κατηγοριοποιημένων μη ταιριασμένων εγγραφών.

Δύο επιπλέον μέτρα είναι η *ακρίβεια* (precision) και η *ανάκληση* (recall). Η ακρίβεια μετρά την καθαρότητα των αποτελεσμάτων αναζήτησης, ή πόσο καλά μια αναζήτηση αποφεύγει να επιστρέφει αποτελέσματα που δεν είναι σχετικά. Η ανάκληση αναφέρεται στην πληρότητα της ανάκτησης σχετικών στοιχείων. Για τη διασύνδεση εγγραφών, η ακρίβεια

μπορεί να οριστεί σαν τον αριθμό των σωστά συνδεδεμένων ζευγαριών εγγραφών διαιρεμένο με το συνολικό αριθμό των συνδεδεμένων ζευγαριών εγγραφών. Είναι όμοια δηλαδή, με την αναμενόμενη αποδεκτή τιμή. Παρόμοια, η ανάκληση ορίζεται ως ο αριθμός των σωστά συνδεδεμένων ζευγαριών εγγραφών διαιρεμένος με το συνολικό αριθμό των σωστά ταιριασμένων ζευγαριών εγγραφών. Έτσι, η ανάκληση είναι αντίστοιχη της ευαισθησίας. Φυσικά, η ανάκληση και η ακρίβεια μπορούν να οριστούν και για τις μη ταιριασμένες εγγραφές.

Κάποια επιπλέον κριτήρια για τη διασύνδεση εγγραφών έχουν να κάνουν με το χρόνο και τον αριθμό των εγγραφών που απαιτούν χειροκίνητη αναθεώρηση:

- Χρόνος που απαιτείται: Ο χρόνος πολυπλοκότητας ενός αλγορίθμου διασύνδεσης εγγραφών εξαρτάται κυρίως από τον αριθμό των συγκρίσεων των εγγραφών που γίνονται. Ο χρόνος που χρειάζεται για την ταξινόμηση με βάση ένα blocking κλειδί για πολύ μεγάλα σύνολα δεδομένων, είναι επίσης αρκετά μεγάλος.
- Αριθμός εγγραφών που απαιτούν *παρέμβαση από τον άνθρωπο* (clerical review): Η χειροκίνητη αναθεώρηση των εγγραφών είναι αρκετά χρονοβόρα, ακριβή και έχει μια τάση να δημιουργεί λάθη.

Όλα τα κριτήρια που αναφέρθηκαν παραπάνω, υποθέτουν ότι η κατηγοριοποίηση γίνεται σε εγγραφές που ταιριάζουν και σε εγγραφές που δεν ταιριάζουν, δεν υπάρχει η τρίτη κατηγορία όπου απαιτείται η ανθρώπινη παρέμβαση για να ληφθεί απόφαση σχετικά με τις εγγραφές. Έτσι, για την περίπτωση που υπάρχουν εγγραφές που είναι πιθανόν να ταιριάζουν (possible matches) υπάρχει το κριτήριο *pp* που ορίζεται από τη σχέση $pp=(N_{P,M}+N_{P,U})/(n_m+n_{fp}+n_u+n_{fn})$, όπου $N_{P,M}$ είναι ο αριθμός των πραγματικά ταιριασμένων εγγραφών που έχουν κατηγοριοποιηθεί σαν εγγραφές που είναι πιθανόν να ταιριάζουν, και $N_{P,U}$ είναι ο αριθμός των πραγματικά μη ταιριασμένων εγγραφών που έχουν κατηγοριοποιηθεί σαν εγγραφές που είναι πιθανόν να ταιριάζουν. Αυτό το κριτήριο ποσοτικοποιεί το ποσοστό των ζευγαριών εγγραφών που κατηγοριοποιούνται σαν εγγραφές που είναι πιθανόν να ταιριάζουν, οπότε απαιτείται η ανθρώπινη παρέμβαση για να παρθεί απόφαση. Χαμηλές τιμές του *pp* αντιστοιχούν σε μικρότερη απαίτηση ανθρώπινης παρέμβασης.

2.5 Το Ζήτημα της Ιδιωτικότητας στη Διασύνδεση Εγγραφών

Διάφορα ζητήματα που έχουν να κάνουν με την ιδιωτικότητα και τις νομοθετικές υποχρεώσεις του ατόμου συναντώνται συχνά σε μια μελέτη διασύνδεσης εγγραφών [5]. Σε πολλές μελέτες που σκοπό έχουν την διασύνδεση εγγραφών, πληροφορίες που αφορούν την ταυτοποίηση ατόμων δεν είναι πάντα απαραίτητες. Τέτοιες πληροφορίες θα πρέπει λοιπόν, να μπορούν να αφαιρεθούν από το σύνολο δεδομένων που πρόκειται να συνδεθεί. Η ανωνυμία των ατόμων, για τα οποία πληροφορίες περιέχονται σε σύνολα δεδομένων που πρόκειται να συνδεθούν, είναι συχνά επιθυμητή. Μέθοδοι που πετυχαίνουν αυτό το ζητούμενο, περιλαμβάνουν την κωδικοποίηση της πληροφορίας ταυτοποίησης στις πηγές δεδομένων, με τρόπο συνεπή με τη διασύνδεση. Η διασύνδεση εγγραφών, που περιλαμβάνουν πληροφορίες με κωδικοποιημένα χαρακτηριστικά ταυτοποίησης, είναι πιθανό να μειώνουν την ακρίβεια της.

Προκειμένου να επιτευχθεί η ανώνυμη διασύνδεση εγγραφών πρέπει:

– Να ξεχωρίζεται η πληροφορία ταυτοποίησης, όπως είναι το όνομα και η διεύθυνση, από την υπόλοιπη πληροφορία, όπως για παράδειγμα ζητήματα υγείας, στις πηγές δεδομένων. Αυτό επιτρέπει ένα διαχωρισμό ανάμεσα στα δεδομένα για τη διασύνδεση εγγραφών και στα δεδομένα για τον ερευνητή.
– Τα κλειδιά ταυτοποίησης ατόμων να έχουν περιορισμένη εμβέλεια. Η εμβέλεια αυτή περιορίζεται στην εκάστοτε συγκεκριμένη μελέτη.

Κεφάλαιο 3

Τεχνικές Blocking

3.1 Εισαγωγή

Κάθε εγγραφή σε ένα σύνολο δεδομένων πρέπει να συγκριθεί με όλες τις εγγραφές σε ένα δεύτερο σύνολο δεδομένων, με αποτέλεσμα ο αριθμός των συγκρίσεων να αυξάνεται τετραγωνικά με τον αριθμό των εγγραφών που ταιριάζουν. Αυτή η προσέγγιση είναι υπολογιστικά ανέφικτη για μεγάλα σύνολα δεδομένων. Για να μειωθεί ο τεράστιος αριθμός των πιθανών συγκρίσεων, οι παραδοσιακές τεχνικές διασύνδεσης εγγραφών λειτουργούν με βάση μια blocking λογική. Χρησιμοποιούν δηλαδή, μια ιδιότητα των εγγραφών (ή υποσύνολο ιδιοτήτων) για να χωρίσουν τα σύνολα δεδομένων σε blocks. Το blocking χρησιμοποιείται για να μειώνει τον αριθμό των συγκρίσεων ζευγών εγγραφών, το οποίο επιτυγχάνεται με την ένωση πιθανών όμοιων εγγραφών. Οι εγγραφές που πρόκειται στη συνέχεια να συγκριθούν λεπτομερώς, παράγονται από τις εγγραφές που ανήκουν στο ίδιο block (δηλαδή εγγραφές με την ίδια τιμή σε μια blocking ιδιότητα). Είναι σημαντικό να δημιουργούνται blocks σωστού μεγέθους. Η αναλογία ανάμεσα στον αριθμό και στο μέγεθος των blocks είναι ιδιαιτέρως σημαντική, ειδικά όταν πρόκειται να ταιριάξουν μεγάλες πηγές δεδομένων.

Η κύρια ιδέα στην οποία στηρίζεται το blocking, είναι η ομαδοποίηση παρόμοιων εγγραφών σε blocks ή *δέσμες ομοειδών* (clusters), με βάση την πληροφορία που αυτές διαθέτουν. Μια μεταβλητή blocking θα πρέπει να περιλαμβάνει ένα σύνολο χαρακτηριστικών, τα οποία έχουν μικρή πιθανότητα λάθους. Λάθη στα χαρακτηριστικά που χρησιμοποιούνται ως blocking μεταβλητές, μπορούν να οδηγήσουν στην αποτυχία σύνδεσης εγγραφών. Για blocking μεταβλητές τύπου αλφαριθμητικών, έχουν αναπτυχθεί ποικίλοι φωνητικοί κώδικες, προκειμένου να αντιμετωπιστούν οι συνέπειες από ορθογραφικά και προφορικά λάθη στην καταγραφή ονομάτων. Φωνητικοί κώδικες που χρησιμοποιούνται πολύ συχνά είναι ο *Soundex* [8] και ο *NYSIIS* [1].

Η επιλογή μιας καλής Blocking μεθόδου μπορεί να μειώσει πολύ τον αριθμό των εγγραφών που πρόκειται να συγκριθούν, με αποτέλεσμα μια σημαντική αύξηση της απόδοσης στην ταχύτητα. Νέες μέθοδοι Blocking έχουν εφαρμοστεί πρόσφατα χρησιμοποιώντας αλγορίθμους ευρετηριοποίησης. Παρακάτω, αναλύουμε τα πρότυπα *Τυποποιημένο Blocking* (Standard Blocking) [2], *Ταξινομημένη Γειτονιά* (Sorted Neighborhood) [2, 10], *Ευρετηριοποίηση Δι-γραμμάτων* (Bigram Indexing) [1, 2], *Canopy Clustering* [2, 9], *Ουρά Προτεραιότητας* (Priority Queue Method) [5, 11] και *Blocking ως Προεπιλογή* (Blocking as Preselection) [5, 12]. Γίνεται μια ιδιαίτερη αναφορά στην ακρίβεια απόδοσης αυτών των Blocking μεθόδων καθώς και στην ευαισθησία (εάν οι εγγραφές που ταιριάζουν δεν είναι στον ίδιο block, δεν θα συγκριθούν και δεν μπορούν ποτέ να αντιστοιχηθούν).

3.2 Blocking Μέθοδοι

3.2.1 Τυποποιημένο Blocking

Η μέθοδος Τυποποιημένο Blocking (Sb) [1, 2, 7] συγκεντρώνει τις εγγραφές σε blocks όπου μοιράζονται το ίδιο blocking κλειδί. Επιλέγουμε μια από τις ιδιότητες των εγγραφών ενός συνόλου δεδομένων και θεωρούμε πως αποτελεί το blocking κλειδί. Ένα παράδειγμα ενός blocking κλειδιού είναι οι πρώτοι τέσσερις χαρακτήρες της ιδιότητας «επώνυμο». Ένα blocking κλειδί μπορεί επίσης να αποτελείται από περισσότερες από μια ιδιότητες, παραδείγματος χάριν, η ιδιότητα «ταχυδρομικός κώδικας» θα μπορούσε να συνδυαστεί με την ιδιότητα «κατηγορία ηλικίας».

Αυτό που θα πρέπει να προσέχουμε στην επιλογή του blocking κλειδιού, είναι τα λάθη που είναι πιθανό να περιέχουν τα χαρακτηριστικά εκείνα των εγγραφών που θα επιλέξουμε σαν κλειδί. Προκειμένου να έχουμε μέγιστη ακρίβεια σύνδεσης, είναι καλό να επιλέγουμε εκείνα τα χαρακτηριστικά που περιέχουν τα λιγότερα λάθη. Εάν τα blocks που προκύψουν περιέχουν ένα μεγάλο αριθμό εγγραφών, θα παραχθούν περισσότερες εγγραφές από τις απαραίτητες, οδηγώντας σε έναν μεγάλο αριθμό συγκρίσεων. Παραδείγματος χάριν, χρησιμοποιώντας την ιδιότητα «γένος» σαν blocking κλειδί, οι διαθέσιμες εγγραφές χωρίζονται σε δύο πολύ μεγάλα blocks. Από την άλλη πλευρά, εάν τα blocks είναι πάρα πολύ μικρά, οι εγγραφές που περιέχονται σε κάθε block είναι λιγότερες, με αποτέλεσμα να μειώνεται η ακρίβεια της διασύνδεσης (ευαισθησία). Παραδείγματος χάριν, αν χρησιμοποιήσουμε σαν blocking κλειδί την ιδιότητα «αριθμός κοινωνικής ασφάλισης» (SSN), προκύπτουν μικρά blocks, σε πλήθος ίσο με τα άτομα μέσα στα σύνολα δεδομένων.

Πολλαπλά blocking κλειδιά χρησιμοποιούνται επίσης, για να μετριαστούν τα λάθη στα blocking κλειδιά. Κάνοντας πολλές επαναλήψεις με διαφορετικά blocking κλειδιά δημιουργούνται διαφορετικά blocks και διαφορετικές συγκρίσεις εγγραφών. Επίσης, οι πολλές επαναλήψεις βελτιώνουν την ακρίβεια διασύνδεσης. Ωστόσο, κάποιες φορές είναι δύσκολο να επιτευχθεί η εφαρμογή και ο συντονισμός των πολλαπλών blocks, καθώς και οι πολλαπλές συγκρίσεις των συνόλων εγγραφών. Τέλος, μια άλλη στρατηγική που χρησιμοποιείται για να μειώσει τα λάθη που είναι πιθανόν να έχουν τα χαρακτηριστικά των εγγραφών που χρησιμοποιούνται σαν blocking κλειδιά (ιδιότητες όπως «όνομα» και «διεύθυνση»), είναι να χρησιμοποιηθούν φωνητικές κωδικοποιήσεις όπως ο Soundex.

Ο αριθμός των ζευγών εγγραφών, που δημιουργούνται με τις Blocking μεθόδους και πρόκειται να συγκριθούν, εξαρτάται από τον αριθμό των blocks και το μέγεθός τους. Έτσι, αν υποθέσουμε ότι πρόκειται να συνδεθούν δύο σύνολα δεδομένων με n εγγραφές το κάθε ένα και χρησιμοποιήσουμε μια Blocking μέθοδο που οδηγεί σε β blocks (όλα ίδιου μεγέθους, περιέχουν n/β εγγραφές), οι συγκρίσεις που προκύπτουν είναι $O(n^2/\beta)$. Αυτό είναι φυσικά ιδανική περίπτωση η οποία είναι δύσκολο να επιτευχθεί με πραγματικά στοιχεία. Κατά συνέπεια, ο αριθμός των ζευγών εγγραφών που πρόκειται να συγκριθούν εξαρτάται από το μεγαλύτερο block.

Η μέθοδος του Τυποποιημένου Blocking μπορεί να αυξήσει αισθητά την ταχύτητα της διαδικασίας σύγκρισης, ωστόσο μπορεί να οδηγήσει και σε έναν αυξανόμενο αριθμό λανθασμένων αντιστοιχιών λόγω των λαθών που μπορεί να συμβούν κατά τη διάρκεια της διαδικασίας ταιριάσματος των εγγραφών.

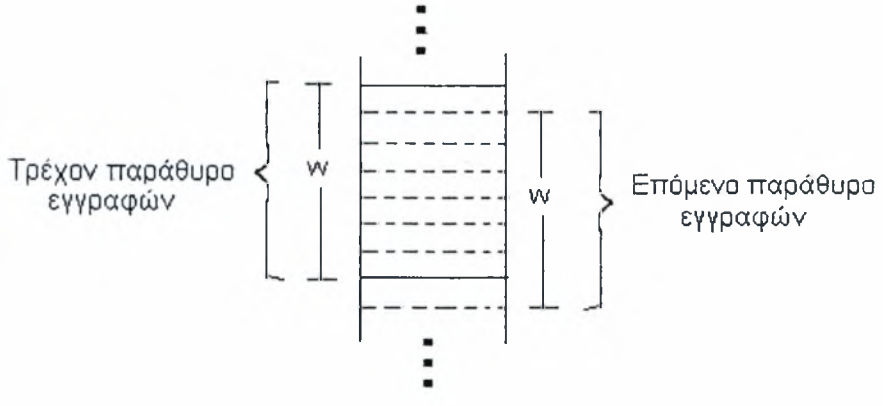
3.2.2 Ταξινομημένη Γειτονιά

Η μέθοδος Ταξινομημένης Γειτονιάς (SN) [1, 2, 7, 10] ταξινομεί τις εγγραφές με βάση ένα κλειδί ταξινόμησης και κινεί έπειτα ένα παράθυρο, σταθερού μεγέθους w, διαδοχικά πάνω από τις ταξινομημένες εγγραφές. Οι εγγραφές μέσα στο παράθυρο συνδυάζονται η μια με την άλλη και στη συνέχεια κατατάσσονται στον κατάλογο των εγγραφών που είναι υποψήφιες να συγκριθούν. Η χρήση του παραθύρου περιορίζει τον αριθμό των πιθανών συγκρίσεων για κάθε εγγραφή σε $2w-1$. Ο συνολικός αριθμός συγκρίσεων εγγραφών, υποθέτοντας δύο σύνολα δεδομένων με n εγγραφές το κάθε ένα, που προκύπτει από τη μέθοδο της ταξινομημένης γειτονιάς είναι $O(wn)$.

Όπως και με το Τυποποιημένο Blocking, είναι πιο αποδοτικό να γίνουν αρκετές επαναλήψεις με διαφορετικά κλειδιά ταξινόμησης και ένα μικρότερο μέγεθος παραθύρου, παρά να γίνει μια επανάληψη μόνο με ένα μεγάλο μέγεθος παραθύρου. Πρόβλημα με την μέθοδο Ταξινομημένης Γειτονιάς προκύπτει, εάν ένας αριθμός εγγραφών μεγαλύτερος από το μέγεθος του παραθύρου έχει την ίδια τιμή στο κλειδί ταξινόμησης. Παραδείγματος χάριν, έχοντας ως κλειδί ταξινόμησης το επώνυμο, εκατοντάδες εγγραφές μπορούν να έχουν την τιμή «Smith», και εάν το μέγεθος του παραθύρου είναι μικρό δεν θα συγκριθούν όλες οι εγγραφές με αυτή την τιμή κλειδιού.

Η μέθοδος Ταξινομημένης Γειτονιάς αποτελείται από τα ακόλουθα τρία βήματα:

- Δημιουργία κλειδιού: Υπολογίζεται ένα κλειδί, για κάθε εγγραφή που βρίσκεται στη λίστα των εγγραφών που είναι υποψήφιες να συγκριθούν. Ένα κλειδί ταξινόμησης είναι μια ακολουθία ιδιοτήτων των εγγραφών.
- Ταξινόμηση δεδομένων: Οι εγγραφές στη βάση δεδομένων ταξινομούνται με βάση το κλειδί που βρίσκεται στο πρώτο βήμα.
- Συγχώνευση: Ένα παράθυρο καθορισμένου μεγέθους, κινείται πάνω στη λίστα των εγγραφών, προκειμένου να περιορίσει τις συγκρίσεις των εγγραφών αυτών με εκείνες του παραθύρου. Εάν το μέγεθος του παραθύρου είναι w εγγραφές, κάθε νέα εγγραφή που εισέρχεται στο παράθυρο, συγκρίνεται με τις προηγούμενες $w-1$ εγγραφές για να βρεθούν αυτές που ταιριάζουν. Στο Σχήμα 3 φαίνεται η διαδικασία της συγχώνευσης.



Σχήμα 3 – Διαδικασία συγχώνευσης

Οι τιμές που μπορεί να πάρει το μέγεθος του παραθύρου κυμαίνονται από *2* (όπου συγκρίνονται δύο μόνο συνεχόμενα στοιχεία) έως *n* (όπου κάθε στοιχείο συγκρίνεται με όλα τα άλλα).

Η προσέγγιση Ταξινομημένης Γειτονιάς στηρίζεται στην υπόθεση ότι οι διπλότυπες εγγραφές θα είναι κοντά στην ταξινομημένη λίστα, και επομένως θα συγκριθούν κατά τη διάρκεια του βήματος της συγχώνευσης. Η αποδοτικότητα αυτής της προσέγγισης εξαρτάται από το κλειδί που επιλέγεται για να ταξινομήσει τις εγγραφές. Γενικά, κανένα κλειδί δεν θα είναι αρκετό να ταξινομήσει τις εγγραφές κατά τέτοιο τρόπο ώστε όλες αυτές που ταιριάζουν να μπορούν να ανιχνευθούν. Εάν το λάθος σε μια εγγραφή εμφανίζεται στο κλειδί ταξινόμησης, υπάρχει πολύ μικρή πιθανότητα αυτή η εγγραφή να καταλήξει κοντά σε μια άλλη που ταιριάζει μετά την ταξινόμηση.

3.2.3 Ευρετηριοποίηση Δι-γραμμάτων

Η μέθοδος Ευρετηριοποίησης Δι-γραμμάτων (BI) [1, 2] επιτρέπει το *συγκεχυμένο* (fuzzy) blocking. Σύμφωνα με αυτή τη μέθοδο, οι τιμές του blocking κλειδιού μετατρέπονται σε μια λίστα δι-γραμμάτων (υπο-συμβολοσειρές που περιέχουν δύο χαρακτήρες). Στη συνέχεια, δημιουργούνται υπο-λίστες όλων των πιθανών συνδυασμών των δι-γραμμάτων μεταξύ τους, χρησιμοποιώντας ένα κατώτατο όριο, το *κατώφλι* (μεταξύ 0.0 και 1.0). Οι τελικές λίστες δι-γραμμάτων ταξινομούνται και εισέρχονται σε ένα ανεστραμμένο ευρετήριο, το οποίο χρησιμοποιείται για να ανακτηθούν οι αντίστοιχες εγγραφές ενός block.

Παραδείγματος χάριν, επιλέγοντας για τιμή του blocking κλειδιού την τιμή «baxter» οδηγούμαστε στη λίστα δι-γραμμάτων ('ba','ax','xt','te','er'). Με ένα κατώτατο όριο 0.8 οι ακόλουθες υπο-λίστες μήκους 4 (υπολογίζεται ως μήκος των λιστών δι-γραμμάτων το κατώτατο όριο: 5×0.8) εισέρχονται στο ευρετήριο:

('ax','xt','te','er')
('ba','xt','te','er')
('ba','ax','te','er')
('ba','ax','xt','er')
('ba','ax','xt','te')

Όλες οι εγγραφές που περιέχουν την τιμή blocking κλειδιού «baxter» θα κατηγοριοποιηθούν σε πέντε blocks, με αποτέλεσμα την αύξηση του αριθμού των εγγραφών που θα συγκριθούν έναντι αυτών που προκύπτουν από το τυποποιημένο blocking.

Ο αριθμός των υπο-λιστών που δημιουργούνται με βάση την τιμή του blocking κλειδιού, εξαρτάται από το μήκος της τιμής του κλειδιού και το κατώτατο όριο. Όσο χαμηλότερο το κατώτατο όριο, τόσο κοντύτερες οι υπο-λίστες, αλλά και περισσότερες υπο-λίστες ανά τιμή του blocking κλειδιού, καταλήγοντας σε περισσότερα μικρότερα blocks στο ευρετήριο.

Όπως στο Τυποποιημένο Blocking, ο αριθμός των συγκρίσεων των ζευγών εγγραφών, με δύο σύνολα δεδομένων με *n* εγγραφές το κάθε ένα και β blocks, είναι $O(n^2/\beta)$. Ωστόσο, ο αριθμός των blocks β είναι πολύ μεγαλύτερος στην Ευρετηριοποίηση Δι-γραμμάτων.

3.2.4 Canopy Clustering

Πρόκειται για μια μέθοδο ομαδοποίησης εγγραφών [2, 9], η οποία ολοκληρώνεται σε δύο βήματα. Χρησιμοποιείται μια μετρική απόστασης, για τον αποδοτικό διαχωρισμό των δεδομένων σε υποσύνολα που ονομάζονται *canopies*. Ένα canopy δημιουργείται αρχικά, διαλέγοντας μια εγγραφή τυχαία από ένα σύνολο υποψήφιων εγγραφών (αρχικά αυτό το σύνολο περιλαμβάνει όλες τις εγγραφές). Στη συνέχεια, δημιουργούνται canopies που περιέχουν εγγραφές, οι οποίες απέχουν μια συγκεκριμένη απόσταση από αυτό που δημιουργήθηκε αρχικά. Η εγγραφή που επιλέχθηκε τυχαία αρχικά, και όλες οι εγγραφές που απέχουν από αυτή μια καθορισμένη απόσταση αφαιρούνται στη συνέχεια από το σύνολο των υποψήφιων εγγραφών. Στην περίπτωση του blocking, μόνο το πρώτο βήμα εκτελείται για να δημιουργήσει τα canopies χρησιμοποιώντας την *tf.idf* (Term Frequency/Inverse Document Frequency) μετρική απόστασης. Μόνο εγγραφές που υπάρχουν στο ίδιο canopy θα συγκριθούν λεπτομερώς.

Ο αριθμός συγκρίσεων εγγραφών που προκύπτει από το Canopy Clustering είναι $O(fn^2/c)$ όπου n είναι ο αριθμός των εγγραφών μέσα σε κάθε ένα από τα δύο σύνολα δεδομένων, c είναι ο αριθμός των canopies και f είναι ο μέσος αριθμός των canopies που ανήκουν σε μια εγγραφή. Το f πρέπει να είναι μικρό και το c να είναι μεγάλο, προκειμένου να μειωθεί το κόστος υπολογισμού. Ωστόσο, εάν το f είναι πάρα πολύ μικρό, η μέθοδος δεν θα είναι σε θέση να ανιχνεύσει τα τυπογραφικά λάθη.

3.2.5 Μέθοδος Ουράς Προτεραιότητας

Αυτή η μέθοδος [5, 11] σχετίζεται με την SN μέθοδο, με τη διαφορά ότι οι πιο πρόσφατες ομάδες εγγραφών, που ανήκουν στην ταξινομημένη λίστα, αποθηκεύονται σε μια ουρά προτεραιότητας. Ειδικές μέθοδοι απαιτούνται για τη συλλογή αυτών των εγγραφών. Το πλεονέκτημα της μεθόδου είναι η αποφυγή της ταξινόμησης των πηγών δεδομένων σε κάθε σημείο blocking. Αυτό το γεγονός μπορεί να σώσει πολύτιμο υπολογιστικό χρόνο για πολύ μεγάλες πηγές δεδομένων.

Ο αλγόριθμος περνάει τις εγγραφές δύο φορές. Την πρώτη αντιμετωπίζει την κάθε εγγραφή σαν ένα μεγάλο αλφαριθμητικό και τις ταξινομεί λεξικογραφικά, διαβάζοντας από τα αριστερά προς τα δεξιά. Την δεύτερη φορά κάνει ακριβώς το ίδιο, διαβάζοντας όμως από δεξιά προς τα αριστερά. Ο αλγόριθμος ψάχνει την βάση δεδομένων με μια σειρά και αποφασίζει αν κάθε εγγραφή που συναντά είναι ή δεν είναι μέλος μιας ομάδας εγγραφών. Η ομάδα αυτή των εγγραφών αναπαρίσταται στην ουρά προτεραιότητας. Αν είναι μέλος της ουράς προτεραιότητας προχωρά στην επόμενη εγγραφή, ενώ αν δεν είναι, η εγγραφή συγκρίνεται με κάποιες άλλες αντιπροσωπευτικές εγγραφές της ουράς προτεραιότητας. Αν μια από αυτές τις συγκρίσεις είναι επιτυχής, η εγγραφή ενσωματώνεται στην ομάδα της εγγραφής με την οποία συγκρίθηκε. Απ΄ την άλλη πλευρά, αν αποτύχουν όλες οι συγκρίσεις, η εγγραφή αποτελεί μόνη της μια νέα ομάδα.

3.2.6 Blocking σαν Προεπιλογή

Η μέθοδος αυτή [5, 12] βασίζεται σε γρήγορα υπολογιζόμενους κανόνες απόρριψης, εφόσον σχεδόν όλα τα ζευγάρια εγγραφών μπορούν να κατηγοριοποιηθούν σαν μη-ταιριάζοντα, μέσω ενός απλού υπολογισμού. Ως προεπιλογή ορίζεται η εφαρμογή επαρκών κανόνων απόρριψης, για την μείωση του αριθμού των συγκρίσεων. Οι κανόνες απόρριψης προέρχονται από τις συναρτήσεις σύγκρισης που χρησιμοποιούνται. Η Canopies Clustering είναι μια από τις πιο αποτελεσματικές μεθόδους ομαδοποίησης η οποία μπορεί να χρησιμοποιηθεί στην προεπιλογή.

3.3 Φωνητικοί Αλγόριθμοι

Οι αλγόριθμοι αυτοί κωδικοποιούν μια ακολουθία αλφαριθμητικών και υπολογίζουν ένα συγκεκριμένο βάρος. Έπειτα βασιζόμενοι στο βάρος, κάνουν την σύγκριση των αλφαριθμητικών για να ελέγξουν κατά πόσο ταιριάζουν ή όχι. Οι μετρικές φωνητικής ομοιότητας προσπαθούν να ταιριάξουν αλφαριθμητικά που είναι φωνητικά παρόμοια.

3.3.1 Soundex

Ο *Soundex* [3, 8], είναι ο πιο κοινός αλγόριθμος κωδικοποίησης. Είναι ουσιαστικά ένας φωνητικός αλγόριθμος, βασισμένος στην ανάθεση κωδικοποιημένων ψηφίων. Τα ψηφία αυτά είναι τα ίδια για κάθε φωνητικά παρόμοια ομάδα συμφώνων και χρησιμοποιείται κυρίως, για να ταιριάξει επώνυμα. Οι κανόνες της κωδικοποίησης Soundex είναι οι ακόλουθοι:

- Κρατείται το πρώτο γράμμα του επωνύμου ως πρόθεμα χωρίς να κωδικοποιηθεί.
- Αγνοούνται εντελώς όλες οι εμφανίσεις των *W* και *H*.
- Τα υπόλοιπα γράμματα κωδικοποιούνται με βάση τους παρακάτω κανόνες:
  - *B, F, P, V→ 1*
  - *C, G, J, K, Q, S, X, Z→2*
  - *D, T→3*
  - *L→4*
  - *M, N→5*
  - *R→6*
- Τα *A, E, I, O, U* και το *Y* δεν κωδικοποιούνται αλλά χρησιμεύουν ως διαχωριστές.
- Εξαιρέσεις αποτελούν γράμματα που ακολουθούν γράμματα που έχουν τον ίδιο κωδικό, ή τα προθέματα τα οποία, αν κωδικοποιηθούν, θα έχουν τον ίδιο κωδικό. Αυτά αγνοούνται σε όλες τις περιπτώσεις, εκτός εάν ένας διαχωριστής προηγείται από αυτά.

Ο κώδικας σχεδιάστηκε αρχικά για τα Καυκάσια επώνυμα, αλλά εργάζεται καλά και για ονόματα με διαφορετική προέλευση (όπως εκείνα που εμφανίζονται στα αρχεία της αμερικάνικης μετανάστευσης). Ωστόσο, για τα ονόματα ανατολικής ασιατικής προέλευσης, αυτός ο κώδικας είναι λιγότερο ικανοποιητικός, επειδή ένα μεγάλο μέρος της διακριτικής

δύναμης αυτών των ονομάτων οφείλεται στους ήχους φωνήεντος, τους οποίους ο κώδικας αγνοεί.

3.3.2 NYSIIS

Το σύστημα *NYSIIS* (New York State Identification and Intelligence System) [1, 2, 3], διαφέρει από το Soundex στο ότι διατηρεί τις πληροφορίες για τη θέση των φωνηέντων στην κωδικοποιημένη λέξη, μετατρέποντας τα περισσότερα φωνήεντα στο γράμμα *A*. Επιπλέον, το NYSIIS δεν χρησιμοποιεί αριθμούς για να αντικαταστήσει γράμματα. Αντίθετα, αντικαθιστά τα σύμφωνα με άλλα φωνητικά παρόμοια γράμματα, επιστρέφοντας κατά συνέπεια έναν κώδικα γραμμάτων (κανένα αριθμητικό συστατικό). Το NYSIIS υλοποιείται με βάση κάποια καλώς ορισμένα βήματα. Το πρώτο γράμμα ενός ονόματος αλλάζεται αν θεωρηθεί απαραίτητο. Το ίδιο συμβαίνει και με το τελευταίο γράμμα του ονόματος. Μετά, αρχίζοντας από το δεύτερο γράμμα, συμβαίνει μια αναζήτηση για κάθε γράμμα του ονόματος χρησιμοποιώντας ένα φανταστικό δείκτη. Σε κάθε βήμα συμβαίνουν αλλαγές σύμφωνα με κάποιους κανόνες που ονομάζονται *'loop'*. Οι κανόνες εφαρμόζονται κάθε φορά που μετακινείται ο δείκτης στο επόμενο γράμμα του ονόματος. Κατά τη διάρκεια της διαδικασίας, χαρακτήρες μπορεί να μεταφερθούν από το μεταλλαγμένο όνομα για να σχηματίσουν τον NYSIIS κωδικό. Τελικά, το τελευταίο κομμάτι του NYSIIS κωδικού που διαμορφώνεται με αυτόν τον τρόπο, υποβάλλεται σε μια πρόσθετη δοκιμασία και αν χρειαστεί τροποποιείται.

Ο *Tafts* σύγκρινε το Soundex με το NYSIIS, χρησιμοποιώντας μια βάση δεδομένων ονομάτων της πολιτείας της Νέας Υόρκης. Κατέληξε στο γεγονός ότι το NYSIIS είναι 98.72% ακριβές, ενώ το Soundex είναι 95.99% ακριβές για τον εντοπισμό των επωνύμων. Το σύστημα κωδικοποίησης NYSIIS χρησιμοποιείται ακόμα και σήμερα από το δικαστικό σώμα της πολιτείας της Νέας Υόρκης.

3.3.3 ONCA

Μια μελέτη που αναφέρεται σαν *ONCA* (Oxford Name Compression Algorithm) [3, 13], χρησιμοποιεί μια έκδοση του NYSIIS σαν το αρχικό στάδιο, και στο τροποποιημένο όνομα εφαρμόζει στη συνέχεια την κωδικοποίηση Soundex. Αυτή η τεχνική δύο βημάτων χρησιμοποιείται επιτυχώς στη διαδικασία του blocking και αντιμετωπίζει όλα τα χαρακτηριστικά που δεν έχουν ικανοποιηθεί με την κωδικοποίηση Soundex. Η κωδικοποίηση που προκύπτει από αυτήν την τεχνική έχει μήκος τεσσάρων χαρακτήρων.

Τα blocks που παράγονται χρησιμοποιώντας μόνο την κωδικοποίηση ONCA ποικίλουν σε μέγεθος, από πολύ μικρά και εύχρηστα για τα λιγότερο κοινά επώνυμα, σε πιο μεγάλα και δαπανηρά για τα ευρέως κοινά επώνυμα. Το μέγεθος των blocks που προκύπτουν από την κωδικοποίηση ONCA σε ένα αρχείο, μπορεί να επηρεαστεί αν χρησιμοποιηθεί το φύλο, η ημερομηνία γέννησης ή και ένας συνδυασμός αυτών των στοιχείων.

3.3.4 Metaphone και Double Metaphone

Ο *Philips* πρότεινε τον αλγόριθμο *Metaphone* [3, 8] σαν καλύτερη εναλλακτική λύση του Soundex. Χρησιμοποίησε δεκαέξι ήχους συμφώνου που μπορούν να περιγράψουν έναν μεγάλο αριθμό ήχων που χρησιμοποιούνται σε πολλές αγγλικές και μη-αγγλικές λέξεις. Ο αλγόριθμος αυτός αγνοεί φωνήεντα μετά από το πρώτο γράμμα και μειώνει το υπόλοιπο αλφάβητο σε δεκαέξι ήχους συμφώνου. Τα φωνήεντα διατηρούνται μόνο αν είναι το πρώτο γράμμα. Διπλά γράμματα δεν προστίθενται στον κώδικα. Το μηδέν χρησιμοποιείται για να αναπαραστήσει τον ήχο *'th'* αφού μοιάζει και με το ελληνικό γράμμα *θ*, ενώ το *'X'* χρησιμοποιείται για τον ήχο *'sh'*. Οι δεκαέξι ήχοι συμφώνου είναι οι : *B, X, S, K, J, T, F, H, L, M, N, P, R, 0, W, Y.* Στον παρακάτω πίνακα, Πίνακας 1, φαίνεται και η αντιστοίχηση των γραμμάτων για την κωδικοποίηση.

Συγκριτικά με την κωδικοποίηση Soundex, αυτή η τεχνική είναι ικανή να διακρίνει ονόματα, όπως π.χ Bonner και Baymore (BNR και BMR), για τα οποία η Soundex δίνει την ίδια κωδικοποίηση.

Το *Double Metaphone* [3] είναι μια καλύτερη εκδοχή της κωδικοποίησης Metaphone. Βελτιώνει μερικές επιλογές κωδικοποίησης του Metaphone και επιτρέπει πολλαπλές κωδικοποιήσεις ονομάτων, τα οποία έχουν διάφορες προφορές. Έτσι, όλες οι πιθανές κωδικοποιήσεις εξετάζονται στην προσπάθεια να ανακτηθούν παρόμοια ονόματα. Η εισαγωγή πολλαπλών φωνητικών κωδικοποιήσεων ενισχύει το ταίριασμα με μικρό κόστος.

Γράμμα	Κωδικός	Σχόλια
B	B	εκτός αν είναι στο τέλος μιας λέξης μετά το 'm' όπως στη λέξη 'dumb'
C	X	(sh) εάν "-cia-" ή "-ch-"
	S	είναι "-ci-", "-ce-" ή "-cy-"
		μη προφερόμενο εάν "-sci-", "-sce-" ή "-scy-"
	K	αλλιώς, περιλαμβάνοντας "-sch-"
D	J	εάν είναι στο "-dge-", "-dgy-", "-dgi"
	T	αλλιώς
F	F	
G		μη προφερόμενο εάν είναι στο "-gh-" και όχι στο τέλος ή πριν από ένα φωνήεν
		στο "-gn" ή "-gned"
		στο "-dge-", κ.τ.λ. όπως στον παραπάνω κανόνα
	J	εάν είναι πριν "i", ή "e", ή "y", και όχι διπλό "gg"
	K	αλλιώς
H		μη προφερόμενο εάν είναι μετά από φωνήεν και δεν ακολουθεί φωνήεν
	H	αλλιώς
J	J	
K		μη προφερόμενο εάν είναι μετά από "c"
	K	αλλιώς
L	L	
M	M	
N	N	
P	F	εάν είναι πριν "h"
	P	αλλιώς
Q	K	
R	R	
S	X	(sh) εάν είναι πριν "h" ή στο "-sio-" ή "-sia-"
	S	αλλιώς
T	X	(sh) εάν είναι "-tia-" ή "-tio-"
	0	(th) εάν είναι πριν "h"
		μη προφερόμενο εάν είναι στο "tch-"
	T	αλλιώς
V	F	
W		μη προφερόμενο εάν δεν ακολουθείται από ένα φωνήεν
	W	εάν ακολουθείται από ένα φωνήεν
X	KS	
Y		μη προφερόμενο εάν δεν ακολουθείται από ένα φωνήεν
	Y	εάν ακολουθείται από ένα φωνήεν
Z	S	

Πίνακας 1 – Αντιστοίχηση των γραμμάτων για την κωδικοποίηση Metaphone

3.3.5 Phonex

Ο *Phonex* [8] αποτελεί μια παραλλαγή των Soundex και Metaphone η οποία προσπαθεί να βελτιώσει την ποιότητα της κωδικοποίησης, κάνοντας προηγουμένως επεξεργασία των αλφαριθμητικών. Η κωδικοποιημένη μορφή που προκύπτει αποτελείται από έναν χαρακτήρα αρχής, που ακολουθείται από έναν τριψήφιο αριθμό.

Οι κανόνες που προετοιμάζουν τα δεδομένα για να κωδικοποιηθούν στη συνέχεια είναι οι ακόλουθοι:

- Αφαιρούνται όλοι οι χαρακτήρες *'S'* από το τέλος του ονόματος.
- Ζευγάρια γραμμάτων στην αρχή της λέξης μετατρέπονται σύμφωνα με τους ακόλουθους κανόνες:
  - *KN→N*
  - *WR→R*
  - *PH→F*
- Γράμματα στην αρχή της λέξης μετατρέπονται ως εξής:
  - *H→αφαιρείται*
  - *E, I, O, U, Y→A*
  - *P→B*
  - *V→F*
  - *K, Q→C*
  - *J→G*
  - *Z→S*

Εφόσον τα δεδομένα προετοιμαστούν με την παραπάνω διαδικασία ακολουθεί η κωδικοποίηση τους σύμφωνα με τους παρακάτω κανόνες:

- Διατηρείται το πρώτο γράμμα του ονόματος, και μετακινούνται όλες οι εμφανίσεις των *A, E, H, I, O, U, W, Y* σε άλλες θέσεις.
- Αναθέτουμε στα υπόλοιπα γράμματα μετά το πρώτο, τους ακόλουθους αριθμούς:
  - *B, F, P, V→1*
  - *C, G, J, K, Q, S, X, Z→2*
  - *D, T→3* εάν δεν ακολουθείται από το *C*
  - *L→4* εάν δεν ακολουθείται από φωνήεν ή από τέλος ονόματος
  - *M, N→5* αγνοείται το επόμενο γράμμα εάν είναι *D* ή *G*
  - *R→6* εάν δεν ακολουθείται από φωνήεν ή από τέλος ονόματος
- Αγνοείται το τρέχον γράμμα εάν έχει το ίδιο ψηφίο κωδικοποίησης με τον τελευταίο χαρακτήρα του κωδικού.
- Τελικά, μετατρέπεται ο κωδικός στη μορφή γράμμα, ψηφίο, ψηφίο, ψηφίο προσθέτοντας μηδενικά στο τέλος αν δεν υπάρχουν τόσα ψηφία όσα χρειάζονται.

Αν και ο τελικός κώδικας των τεσσάρων χαρακτήρων είναι όμοιος με αυτόν που παράγεται και με την κωδικοποίηση Soundex, αυτές οι δύο μορφές δεν είναι συμβατές.

3.4 Κριτήρια Πολυπλοκότητας για τις Blocking Μεθόδους

Όπως ήδη έχει αναφερθεί, οι μέθοδοι Blocking χρησιμοποιούνται για να μειώνουν τον αριθμό των ζευγαριών εγγραφών που πρόκειται να συγκριθούν, εφόσον ο συνολικός αριθμός

των συγκρίσεων που πρέπει να γίνουν είναι υπολογιστικά μη εφικτός για μεγάλα σύνολα δεδομένων. Σκοπός λοιπόν αυτών των μεθόδων, είναι να αφαιρούν από το σύνολο των συγκρίσεων εκείνα τα ζευγάρια εγγραφών που δεν ταιριάζουν, χωρίς όμως να χάνουν τα ζευγάρια εγγραφών που ταιριάζουν.

Τρία κριτήρια πολυπλοκότητας που καθορίζουν την αποδοτικότητα και την ποιότητα των Blocking μεθόδων είναι τα παρακάτω [2,25,26]:

- *Αναλογία Μείωσης* (Reduction Ratio RR): ορίζεται ως $RR=1-s/N$, όπου s είναι ο αριθμός των ζευγαριών εγγραφών που παράγονται για σύγκριση από μια Blocking μέθοδο και N είναι ο συνολικός αριθμός των πιθανών ζευγαριών εγγραφών που πρόκειται να συγκριθούν. Αν συνδέουμε δύο σύνολα δεδομένων με n εγγραφές το κάθε ένα τότε $N=n\times n$. Στην περίπτωση της εξάλειψης διπλοτύπων από ένα σύνολο δεδομένων με n εγγραφές, ο αριθμός των συγκρίσεων ζευγαριών εγγραφών είναι $N=n\times(n-1)/2$, καθώς κάθε εγγραφή στο σύνολο δεδομένων πρέπει να συγκριθεί με όλες τις άλλες εκτός από τον εαυτό της. Η αναλογία μείωσης μετρά τη σχετική μείωση του χώρου σύγκρισης, χωρίς όμως να υπολογίζεται η ποιότητα της μείωσης. Η αναλογία μείωσης δεν μετρά τον χρόνο που χρειάζεται για μια συγκεκριμένη εφαρμογή ενός blocking αλγορίθμου. Ο χρόνος που απαιτείται για δύο μεθόδους με ίδιο *RR* μπορεί να ποικίλει σημαντικά. Μερικές μέθοδοι μπορεί να απαιτούν μια ταξινόμηση, η οποία για μεγάλα σύνολα δεδομένων είναι μια διαδικασία που καταναλώνει αρκετό χρόνο.
- *Πληρότητα Ζευγαριών Εγγραφών* (Pairs Completeness PC): ορίζεται ως $PC=s_M/N_M$, όπου s_M είναι ο αριθμός των αληθινά ταιριασμένων ζευγαριών εγγραφών που παράγονται για σύγκριση από την Blocking μέθοδο και N_M είναι ο συνολικός αριθμός των αληθινά ταιριασμένων ζευγών εγγραφών σε όλο το σύνολο δεδομένων.
- Fscore: συνδυάζει το *RR* και το *PC* μέσω μιας αρμονικής σχέσης $Fscore=2\times PC\times RR/(PC+RR)$.

Όσον αφορά τις Blocking μεθόδους δίνουμε σημασία μόνο στις παραμέτρους *RR* και *PC* και όχι τόσο στην ακρίβεια (precision) και στην ανάκληση (recall), που αναφέρθηκαν στην παράγραφο 2.4. Αυτές σχετίζονται με όλη τη διαδικασία της διασύνδεσης των εγγραφών, ενώ οι παράμετροι *RR* και *PC* επιτρέπουν άμεση αποτίμηση των μεθόδων ευρετηριοποίησης, χωρίς να προκαλείται πιθανή σύγχυση των αποτελεσμάτων από τις μεθόδους σύγκρισης και τα μοντέλα κατηγοριοποίησης.

3.5 Φιλτράρισμα: Ένα Βήμα Βελτίωσης του Blocking

Ο αριθμός των ζευγαριών εγγραφών που παράγονται από οποιαδήποτε Blocking μέθοδο εξαρτάται από τον αριθμό των blocks που παράγονται και από το μέγεθος τους. Πολύ μεγάλα blocks έχουν σημαντικές επιδράσεις στην αποτελεσματικότητα των Blocking μεθόδων. Ωστόσο, είναι σχετικά δύσκολο να αποφευχθούν μεγάλα blocks ανεξάρτητα από το ποια μέθοδος Blocking χρησιμοποιείται.

Προκειμένου να βελτιωθεί η αποτελεσματικότητα του Blocking, προτάθηκε ένας προσαρμόσιμος αλγόριθμος *φιλτραρίσματος* [4], σαν ένα βήμα προ-επεξεργασίας της blocking διαδικασίας. Το φιλτράρισμα είναι προσαρμόσιμο με την έννοια ότι ο αριθμός των blocks που πρόκειται να υποστούν φιλτράρισμα, εξαρτάται από τα αποτελέσματα της Blocking μεθόδου. Το φιλτράρισμα γίνεται μόνο σε μεγάλα blocks.

Όλες οι εγγραφές που βρίσκονται σε δύο παρόμοια blocks δεν ταιριάζουν πάντα μεταξύ τους, καθώς το blocking κλειδί που χρησιμοποιείται μπορεί να μην είναι ακριβές ή να περιέχει λάθη. Έτσι πολλές φορές απαιτείται το φιλτράρισμα. Επιλέγεται δηλαδή κάποια επιπλέον πληροφορία σαν μεταβλητή φιλτραρίσματος, ώστε να εξαλειφθούν εγγραφές που δεν είναι δυνατόν να ταιριάζουν πριν γίνει η λεπτομερής διαδικασία σύγκρισης των εγγραφών. Μια μεταβλητή φιλτραρίσματος πρέπει να διαφέρει από το blocking κλειδί, για να είναι σε θέση να ξεχωρίσει τις εγγραφές που δεν ταιριάζουν. Για παράδειγμα, αν σαν blocking κλειδί χρησιμοποιούνται τα τέσσερα πρώτα γράμματα του επωνύμου, σαν μεταβλητή φιλτραρίσματος μπορούμε να χρησιμοποιήσουμε όλο το επώνυμο.

Προκειμένου να αποτιμηθεί η απόδοση του φιλτραρίσματος υπολογίζονται κάποιες παράμετροι παρόμοιες με αυτές που χρησιμοποιούνται και για τις Blocking μεθόδους. Έτσι, αντικαθιστούμε τον αριθμό των ζευγαριών εγγραφών που παράγονται με μια Blocking μέθοδο, *s*, με τον συνολικό αριθμό των ζευγαριών εγγραφών που παράγονται από μια μέθοδο Blocking αλλά και από τη διαδικασία φιλτραρίσματος, *SF*. Υπάρχουν τρία μέτρα που υπολογίζονται: η αναλογία μείωσης που ορίζεται σαν $RR=1-SF/N$ όπου N είναι ο αριθμός όλων των πιθανών ζευγαριών εγγραφών στο σύνολο δεδομένων, η πληρότητα ζευγαριών εγγραφών με $PC=SF_M/N_M$ όπου το M αναφέρεται στα ταιριασμένα ζευγάρια, και το *Fscore*.

Ο αριθμός, N, είναι συνήθως πολύ μεγάλος, για μεγάλα σύνολα δεδομένων, και έτσι το *RR* παίρνει τιμές κοντά στο ένα ανεξάρτητα πως αλλάζει η τιμή του *SF*. Γι' αυτό το λόγο έχει προταθεί ένα άλλο μέτρο, η αναλογία μείωσης φιλτραρίσματος (Filtering Reduction Ratio *FRR*), που μετρά τη μείωση του αριθμού των ζευγαριών εγγραφών που συγκρίνονται, λόγω του φιλτραρίσματος. Το *FRR* ορίζεται ως $FRR=1-SF/S$ όπου S είναι ο αριθμός των ζευγαριών εγγραφών που παράγονται μόνο από τη μέθοδο Blocking. Αν δεν υπάρχει φιλτράρισμα, το *FRR* είναι μηδέν. Η αποτελεσματικότητα του φιλτραρίσματος εξαρτάται από τα χαρακτηριστικά των δεδομένων που πρόκειται να συνδεθούν και από τη μέθοδο Blocking που επιλέγεται.

Κεφάλαιο 4

Το Σύστημα Διασύνδεσης Εγγραφών Febrl

4.1 Εισαγωγή

Το Febrl είναι ένα πρότυπο σύστημα λογισμικού, σχεδιασμένο για τον καθαρισμό και την τυποποίηση δεδομένων, την διαγραφή διπλότυπων σε ένα σύνολο δεδομένων, καθώς και την διασύνδεση εγγραφών δυο ή περισσοτέρων βάσεων δεδομένων [1, 16]. Το σύστημα Febrl υλοποιείται με βάση μια αντικειμενοστραφής δομή που περιλαμβάνει ένα σύνολο μοντέλων, κάθε ένα από τα οποία περιέχει ρουτίνες για συγκεκριμένες εργασίες. Όλες οι εφαρμογές του λογισμικού είναι υλοποιημένες στην γλώσσα προγραμματισμού 'Python' [18] ενώ όλο το σύστημα ελέγχεται κυρίως από το πρόγραμμα project.py.

Αρχικά το Febrl σχεδιάστηκε για να χρησιμοποιηθεί από ερευνητές που ασχολούνται με βιοϊατρικά δεδομένα, παρόλα αυτά είναι δυνατόν να εφαρμοστεί και σε οποιαδήποτε άλλη εφαρμογή. Αυτό φαίνεται και από την ονομασία του, που αποτελεί τα αρχικά των λέξεων «Ελευθέρως Επεκτάσιμη Διασύνδεση Βιοϊατρικών Εγγραφών».

Οι εφαρμογές του Febrl που απασχολούν αυτήν την εργασία έχουν να κάνουν κυρίως με τη δημιουργία συνόλων δεδομένων, που είναι απαραίτητα ώστε να γίνει η σύγκριση των blocking αλγορίθμων, καθώς και με το πρόγραμμα υλοποίησης της απαλοιφής διπλοτύπων. Ωστόσο, αναφέρονται κάποια στοιχεία για τον καθαρισμό και την τυποποίηση των δεδομένων με χρήση κάποιων πιθανοτικών μοντέλων και κανόνων απόφασης, καθώς και πως γίνεται η εξάλειψη διπλότυπων και η διασύνδεση εγγραφών με χρήση πιθανοτικών μοντέλων και ενός ειδικού κατηγοριοποιητή.

4.2 Δημιουργία Συνόλων Δεδομένων

Οι διαδικασίες της εξάλειψης διπλοτύπων και της διασύνδεσης εγγραφών, προκειμένου να παράγουν ορατά αποτελέσματα και να εξεταστεί η αποτελεσματικότητα τους, πρέπει να εφαρμοστούν σε δεδομένα για τα οποία τα αποτελέσματα της διασύνδεσης τους είναι γνωστά εκ των προτέρων. Ωστόσο, εξαιτίας θεμάτων ιδιωτικότητας και εμπιστευτικότητας είναι αδύνατο τέτοια δεδομένα να γίνουν ποτέ διαθέσιμα στο ευρύ κοινό. Επομένως, δεν είναι εφικτός ο αυτόματος έλεγχος των αλγορίθμων και των διαδικασιών της διασύνδεσης εγγραφών, εκτός αν είναι δυνατή η παραγωγή τεχνητών συνόλων δεδομένων. Είναι επίσης δύσκολο, να μάθει κανείς να χρησιμοποιεί συστήματα διασύνδεσης εγγραφών αποτελεσματικά, χωρίς ομάδες δεδομένων όπου το αποτέλεσμα της διασύνδεσης εγγραφών να είναι γνωστό.

Ένα σημαντικό πλεονέκτημα του Febrl είναι ότι παρέχει στους χρήστες την δυνατότητα παραγωγής τεχνητών συνόλων δεδομένων ανάλογα με τις απαιτήσεις τους. Τα τεχνητά σύνολα δεδομένων έχουν το πλεονέκτημα, ότι ο αριθμός των λαθών που παρουσιάζονται, καθώς επίσης και το αποτέλεσμα διασύνδεσης των εγγραφών, είναι γνωστά, ενώ μπορούν να προετοιμαστούν εύκολα κάποια ελεγχόμενα πειράματα. Ο πρώτος τέτοιος generator βάσεων δεδομένων [10], επέτρεπε τη δημιουργία βάσεων δεδομένων που περιέχουν διπλότυπες εγγραφές. Χρησιμοποιούσε λίστες ονομάτων, πόλεων, πολιτειών και ταχυδρομικούς κώδικες ενώ παρείχε ένα μεγάλο αριθμό παραμέτρων, όπως το μέγεθος της βάσης δεδομένων που

επρόκειτο να δημιουργηθεί, το ποσοστό των διπλοτύπων και τον αριθμό και τον τύπο λαθών που παρουσιάζονταν. Στο Febrl το πρόγραμμα που επιτελεί αυτή τη διαδικασία είναι το 'generate.py'. Μπορεί να δημιουργήσει σύνολα δεδομένων που περιέχουν στα πεδία τους ονόματα, διευθύνσεις, ημερομηνίες, αριθμούς τηλεφώνων, και αριθμούς αναγνώρισης, όπως το Α.Φ.Μ. Όλα αυτά είναι διαθέσιμα μέσω κάποιων πινάκων που διαθέτει το Febrl και προέρχονται από την ανάμιξη πραγματικών δεδομένων από τηλεφωνικούς καταλόγους. Επίσης, για έναν μεγάλο αριθμό ονομάτων και διευθύνσεων υπάρχουν διαθέσιμες πολλές διαφορετικές εκδοχές προφοράς τους, όπως επίσης και συνήθη τυπογραφικά λάθη που μπορούν να συμβούν.

Οι παράμετροι που θέτει ο χρήστης κατά τη δημιουργία των δεδομένων περιλαμβάνουν τον αριθμό των πρωτότυπων και διπλότυπων (num_records,num_duplicates) εγγραφών, τον μέγιστο αριθμό των διπλότυπων για μια αυθεντική εγγραφή (max_duplicate_per_record) καθώς επίσης και το πιθανοτικό μοντέλο που θα χρησιμοποιηθεί για τα λάθη που θα εισαχθούν στις διπλότυπες εγγραφές (distribution). Ένα παράδειγμα δημιουργίας ενός συνόλου δεδομένων αποτελεί η εντολή 'python generate.py mydata.csv 10 10 4 2 2 poisson' [1]. Έτσι δημιουργείται μια ομάδα δεδομένων που εμφανίζεται στο αρχείο εξόδου mydata με 10 αυθεντικές εγγραφές και 10 διπλότυπες, με 4 διπλότυπες ανά μια πρωτότυπη και 2 αλλαγές ανά πεδίο και ανά εγγραφή. Το πιθανοτικό μοντέλο που χρησιμοποιείται είναι το Poisson. Ο χρήστης μπορεί επίσης να ελέγξει το μέγιστο αριθμό λαθών που παρουσιάζονται ανά πεδίο και ανά εγγραφή. Οι θέσεις όπου παρουσιάζονται τα λάθη καθώς επίσης και ο τύπος των λαθών που εμφανίζονται, μοντελοποιούνται σύμφωνα με μελέτες που ήδη υπάρχουν και σχετίζονται με παρόμοια λάθη. Τα αρχεία συνόλων δεδομένων που παράγονται είναι σε μορφή αρχείου '.csv'. Είναι αρχεία κειμένου που οι τιμές τους διαχωρίζονται με ένα κόμμα και είναι προσβάσιμα τόσο από έναν επεξεργαστή κειμένου όπως είναι η εφαρμογή 'WordPad' όσο και από την εφαρμογή 'Excel' με την οποία φαίνεται πιο καθαρά ο διαχωρισμός των εγγραφών και των πεδίων. Αυτός ο generator ομάδων δεδομένων και όλα τα συσχετιζόμενα αρχεία αποτελούν μέλος τους συστήματος Febrl και είναι διαθέσιμα στον κατάλογο dsgen.

Η δημιουργία των συνόλων δεδομένων επιτυγχάνεται σε δυο στάδια. Στο πρώτο δημιουργούνται οι πρωτότυπες εγγραφές, ενώ στο δεύτερο δημιουργούνται οι διπλότυπες τους με βάση τυχαίες αλλαγές που υφίστανται οι πρωτότυπες. Κάθε εγγραφή που δημιουργείται χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό, από το οποίο φαίνεται αν είναι πρωτότυπη ή όχι. Αν είναι διπλότυπη φαίνεται ότι είναι το αντίγραφο μιας συγκεκριμένης πρωτότυπης εγγραφής, με βάση το γεγονός ότι έχουν τον ίδιο κωδικό, απλά η πρωτότυπη έχει την κατάληξη 'org' από 'original', ενώ η διπλότυπη έχει κατάληξη 'dup' από 'duplicate'.

Διπλότυπα εγγραφών δημιουργούνται από τυχαία εμφανιζόμενες αλλαγές των παρακάτω μορφών :

- Αν για το αυθεντικό αλφαριθμητικό υπάρχει κάποια άλλη εκδοχή διαφορετικής προφοράς, τότε χρησιμοποιείται αυτή.
- Εισάγεται ένας νέος χαρακτήρας σε τυχαίο σημείο του αλφαριθμητικού.
- Διαγράφεται ένας τυχαίος χαρακτήρας.
- Αντικαθίσταται ένας τυχαίος χαρακτήρας με κάποιον άλλο.
- Αντικαθίσταται ολόκληρο το αλφαριθμητικό από κάποιο άλλο.
- Γίνεται ανταλλαγή θέσεων μεταξύ δύο τυχαίων χαρακτήρων.
- Εισάγεται ένα κενό στο πεδίο, ώστε να διασπαστεί το αλφαριθμητικό.

- Αντίστοιχα διαγράφεται ένα κενό αν υπάρχει.
- Ορίζεται ένα πεδίο της εγγραφής να είναι κενό.
- Αν ένα πεδίο είναι κενό, τότε εισάγεται μια νέα τιμή σε αυτό.
- Ανταλλάσσονται οι τιμές δυο διαφορετικών πεδίων στην ίδια εγγραφή.

4.3 Εφαρμογές Καθαρισμού και Τυποποίησης

Σκοπός των διαδικασιών του καθαρισμού δεδομένων και της τυποποίησης, είναι η μετατροπή της πληροφορίας που βρίσκεται αποθηκευμένη στα πρωτότυπα δεδομένα σε μια καλώς ορισμένη και συνεπής μορφή. Το λογισμικό Febrl παρέχει την δυνατότητα του καθαρισμού και της τυποποίησης των εγγραφών μέσω της εφαρμογής 'standardization.py' [16] η οποία χρησιμοποιεί και ορισμένες ρουτίνες των εφαρμογών 'address.py', 'date.py', 'name.py' και 'phonenum.py'. Οι ρουτίνες αυτών των εφαρμογών περιέχουν συναρτήσεις, απαραίτητες στον καθαρισμό και την τυποποίηση των αλφαριθμητικών στα οποία αντιστοιχούν, δηλαδή για παράδειγμα η 'address.py' για τις διευθύνσεις και η 'phonenum.py' για τους τηλεφωνικούς αριθμούς. Όπως φαίνεται και από την ονομασία των εφαρμογών, το Febrl είναι σε θέση να επεξεργαστεί ονόματα, διευθύνσεις, ημερομηνίες και τηλεφωνικούς αριθμούς.

Τα βήματα της διαδικασίας του καθαρισμού και της τυποποίησης για ονόματα και διευθύνσεις είναι τα εξής [1] :

- Βήμα 1 : Καθαρισμός
- Βήμα 2 : Ανάθεση Ετικετών
- Βήμα 3 : Τμηματοποίηση
- Βήμα 4 : Word Spilling (γίνεται έλεγχος για το εάν σε ένα πεδίο οι λέξεις είναι κομμένες εξαιτίας περιορισμένου μήκους του πεδίου)

4.3.1 Βήμα 1 : Καθαρισμός

Το πρώτο βήμα της διαδικασίας τυποποίησης των δεδομένων είναι ο καθαρισμός των δεδομένων. Η είσοδος στη ρουτίνα καθαρισμού δεδομένων είναι μια ακολουθία αλφαριθμητικών, που μπορεί να είναι είτε ένα όνομα είτε μια διεύθυνση. Πρώτα, όλα τα γράμματα στην ακολουθία μετατρέπονται σε πεζά. Στη συνέχεια, μια λίστα επιδιόρθωσης χρησιμοποιείται, με την οποία ελέγχονται ξεχωριστά τα αλφαριθμητικά της ακολουθίας, ώστε ορισμένα από αυτά να αντικατασταθούν με άλλα που είναι σε κανονική μορφή. Για παράδειγμα, δεδομένης της λίστας επιδιόρθωσης που φαίνεται στον πιο κάτω πίνακα, Πίνακας 2, παραλλαγές του 'known as' όπως είναι 'a.k.a.' ή 'aka' όλες αντικαθίστανται από το αλφαριθμητικό 'known as'.

Πρωτότυπο	Αντικατάσταση
'knownas'	'known as'
'a.k.a'	'known as'
'aka'	'known as'
'babyof'	'baby of'
'b/o'	'baby of'
'b.o.'	'baby of'
'n/a'	' '
'na'	' '
'['	'-'
'('	'-'
'.'	' '

Πίνακας 2 – Λίστα επιδιορθώσεως

Μια λίστα επιδιόρθωσης φορτώνεται από ένα αρχείο λίστας επιδιόρθωσης. Κάθε είσοδος σε μια τέτοια λίστα αποτελείται από ένα αλφαριθμητικό και ένα αντίστοιχο αλφαριθμητικό αντικατάστασης. Γίνεται κατανοητό ότι επιδιορθώνονται λέξεις και εκφράσεις, που είναι δυνατόν να εντοπιστούν οι σωστές εκδοχές τους στην λίστα επιδιορθώσεως.

Κάθε λίστα επιδιορθώσεως ταξινομείται μειώνοντας το μήκος του αρχικού αλφαριθμητικού. Στο παράδειγμά μας, η είσοδος ' knownas ' θα αναζητούνταν πρώτα και αν βρισκόταν θα αντικαθίσταται από το ' known as '. Σημειώνουμε και τα κενά γύρω από κάποιες εισόδους. Είναι σημαντικά, κυρίως για μικρές λέξεις, όπως ' na '. Το όνομα 'bernadette' θα μετατραπεί σε 'ber dette'.

Η έξοδος της ρουτίνας καθαρισμού είναι μια τελείως νέα ακολουθία με αλφαριθμητικά, όπου κάθε αλφαριθμητικό που βρέθηκε στη λίστα επιδιορθώσεως έχει αντικατασταθεί με το αντίστοιχο αλφαριθμητικό αντικατάστασης. Ας σημειωθεί ότι το μήκος του αλφαριθμητικού εξόδου μπορεί να είναι διαφορετικό από αυτό της εισόδου.

4.3.2 Βήμα 2 : Ανάθεση Ετικετών

Αφότου γίνει ο καθαρισμός των αλφαριθμητικών που περιλαμβάνονται σε ένα πεδίο, το επόμενο βήμα είναι να διαχωριστούν σε μια λίστα που περιέχει ονόματα, αριθμούς και πιθανά διαχωριστικά. Το όνομα εισόδου 'doctor peter paul miller' για παράδειγμα, χωρίζεται μέσα σε μια λίστα που περιέχει τέσσερις λέξεις ['doctor', 'peter', 'paul', 'miller']. Όλα τα κενά που προηγούνται και ακολουθούν αφαιρούνται από τα στοιχεία της λίστας.

Χρησιμοποιώντας ποικίλους βοηθητικούς πίνακες, σε κάθε στοιχείο αυτής της λίστας ανατίθεται μία ή περισσότερες ετικέτες ανάλογα με το τι αντιπροσωπεύει. Δηλαδή ορίζεται ότι το συγκεκριμένο στοιχείο είναι κόμμα, τελεία ή παύλα. Στην περίπτωση των λέξεων ανατίθεται ετικέτα που ορίζει αν είναι όνομα, επώνυμο, τίτλος, ταχυδρομικός κώδικας, χώρα

κ.τ.λ.. Στον πιο κάτω πίνακα, Πίνακας 3, φαίνεται μια λίστα μερικών πιθανών ετικετών που μπορούν να υπάρξουν.

Τύπος Στοιχείου	Ετικέτα
Επώνυμο	SN
Αντρικό όνομα	GM
Γυναικείο όνομα	GF
Τίτλος	TI
Χώρα	CR
Περιοχή	LN
Ταχυδρομικός Κώδικας	PC
Αριθμός	NU
Στοιχείο με γράμματα και αριθμούς	AN
Άγνωστο στοιχείο	UN
Κόμμα	CO

Πίνακας 3 – Λίστα πιθανών ετικετών

Εάν μια λέξη βρεθεί σε ένα βοηθητικό πίνακα, δεν της ανατίθεται απλά μια ετικέτα αλλά αντικαθίσταται από την αντίστοιχη διορθωμένη στον βοηθητικό πίνακα.

Είναι πιθανό μια λέξη να βρίσκεται σε παραπάνω από έναν βοηθητικό πίνακα. Συνεπώς, θα της ανατεθούν παραπάνω από μία ετικέτες (ας δούμε για παράδειγμα η λέξη ονόματος 'peter' παρακάτω). Λέξεις που δεν βρίσκονται σε κανένα βοηθητικό πίνακα και που δεν ταιριάζουν με κανένα από τους κανόνες, τους ανατίθεται η ετικέτα 'UN' (unknown). Μια λέξη τίτλου όπως 'doctor' για παράδειγμα παίρνει την ετικέτα 'TI' και θα αντικατασταθεί με την λέξη 'dr', όπως είναι και οι λέξεις 'md' και 'phd'(χρησιμοποιώντας τον παρακάτω βοηθητικό πίνακα, Πίνακας 4).

Πρωτότυπο	Αντικατάσταση
'doctor'	'dr'
'doc'	'dr'
'md'	'dr'
'phd'	'dr'
'miss'	'ms'
'misses'	'ms'
'mister'	'mr'

Πίνακας 4 – Βοηθητικός Πίνακας

Οι βοηθητικοί πίνακες σαρώνονται χρησιμοποιώντας έναν αλγόριθμο ταιριάσματος, ο οποίος ψάχνει για το μακρύτερο συνδυασμό στοιχείων που ταιριάζει με μια είσοδο σε ένα βοηθητικό πίνακα. Για παράδειγμα, ο συνδυασμός των λέξεων ('macquarie' , 'fields') θα ταιριάξει με μια είσοδο σε ένα βοηθητικό πίνακα με την τοπικότητα 'macquarie fields', περισσότερο από την μικρότερη είσοδο 'macquarie' από τον ίδιο πίνακα.

Καθώς η είσοδος σε μια ρουτίνα όπου ανατίθεται ετικέτα είναι ένα καθαρό αλφαριθμητικό, η έξοδος είναι μια λίστα στοιχείων και η αντίστοιχη λίστα ετικετών. Για παράδειγμα εάν ως είσοδο έχουμε το αλφαριθμητικό ονόματος 'doctor peter paul miller' μια πιθανή έξοδος είναι η εξής :

Λίστα στοιχείων: ['dr', 'peter', 'paul', 'miller']
Λίστα ετικετών: ['TI', 'GM/SN', 'GM', 'SN']

υποθέτοντας ότι το 'peter' έχει δύο ετικέτες διότι ενδέχεται να είναι είτε αντρικό όνομα είτε επώνυμο.

4.3.3 Βήμα 3 : Τμηματοποίηση

Αν μια λίστα λέξεων και ετικετών είναι διαθέσιμη, οι ετικέτες χρησιμοποιούνται για να διαχωρίσουν τα στοιχεία εισόδου σε σωστά πεδία εξόδου. Οι απαιτούμενες είσοδοι για αυτήν την ρουτίνα είναι μια λίστα από λέξεις και μια λίστα από τις ετικέτες τους. Ελέγχει την ετικέτα κάθε στοιχείου και το τοποθετεί στο αντίστοιχο πεδίο, ώστε κάθε εγγραφή να πάρει την τυποποιημένη μορφή.

4.3.4 Βήμα 4 : Word Spilling

Γίνεται έλεγχος για το εάν σε ένα πεδίο οι λέξεις είναι κομμένες εξαιτίας περιορισμένου μήκους του πεδίου. Αυτό το βήμα συμβαίνει, όταν δεδομένα εισάγονται σε πεδία που έχουν καθορισμένο μήκος και αυτόματα όταν ένα πεδίο γεμίσει, συνεχίζει στο επόμενο χωρίς να σταματά.

4.4 Υλοποίηση της Διασύνδεσης Εγγραφών στο Febrl

Η διαδικασία της διασύνδεσης των εγγραφών αποτελείται από τρία βήματα όπως έχει ήδη αναφερθεί στην παράγραφο 2.3. Απαιτεί αρχικά την ομαδοποίηση των εγγραφών σε ευρετήρια, μετά την σύγκριση τους ώστε να παραχθεί ένα διάνυσμα βάρους, και τέλος την εξέταση του διανύσματος από έναν κατηγοριοποιητή ο οποίος θα αποφανθεί για το αν ταιριάζουν οι εγγραφές. Το λογισμικό Febrl έχει υλοποιημένες κάποιες διαδικασίες προκειμένου να φέρει σε πέρας την παραπάνω διαδικασία.

4.4.1 Μέθοδοι Δημιουργίας Ευρετηρίων

Το Febrl διαθέτει αρκετές μεθόδους δημιουργίας ευρετηρίων, οι οποίες είναι υλοποιημένες στην εφαρμογή 'indexing.py' [16]. Όλες οι μέθοδοι ευρετηριοποίησης έχουν

τα ακόλουθα χαρακτηριστικά που χρειάζονται όταν αρχικοποιείται μια μέθοδος: όνομα, περιγραφή, ομάδα δεδομένων, ορισμός_block, skip_missing (= μια σημαία που ελέγχει αν οι εγγραφές έχουν κενά οπότε και τις προσπερνά).

Η πιο απλή μέθοδος ευρετηριοποίησης είναι η 'direct' [16] η οποία θεωρεί τις τιμές των πεδίων ως τις μεταβλητές του ευρετηρίου. Η 'truncate' [16], με βάση έναν αριθμό που ορίζεται ως παράμετρος, οργανώνει τα αλφαριθμητικά των πεδίων σε ευρετήρια, σπάζοντας αυτά που ξεπερνούν σε μήκος την παράμετρο αυτή. Άλλες μέθοδοι βασίζονται στην κωδικοποίηση των τιμών των πεδίων με έναν φωνητικό αλγόριθμο, ο οποίος αφού κωδικοποιήσει τις τιμές μετά τις οργανώνει σε ευρετήρια. Το Febrl διαθέτει αρκετούς τέτοιους αλγορίθμους, οι οποίοι είναι υλοποιημένοι στην εφαρμογή 'encode.py' [16] και περιγράφηκαν αναλυτικά στην παράγραφο 3.3. Φαίνεται παρακάτω, Σχήμα 4, και ένα παράδειγμα κωδικοποίησης κάποιων blocking κλειδιών με βάση τεχνικές φωνητικής ομοιότητας.

Original names:

Name	Phonex	Soundex	ModSoundex	NYSIIS	D-Metaphone
peter	b360	p360	p690	pata	ptr
christen	c623	c623	c936	chra	krst
ole	a400	o400	o700	ol	al
nielsen	n250	n425	n738	nals	nlsn
markus	m200	m622	m930	marc	mrks
hegland	a245	h245	h478	hagl	hkln
stephen	s315	s315	s618	staf	stfn
steve	s310	s310	s620	staf	stf
roberts	r100	r163	r196	raba	rprt
tim	t500	t500	t800	tan	tm
churches	c200	c622	c930	carc	xrxs
xiong	x500	x520	x848	xang	snk
ng	n000	n200	n400	ng	nk
miller	m460	m460	m790	mala	mlr
millar	m460	m460	m798	mala	mlr
foccachio	f200	f220	f300	faca	fkx
van de hooch	f532	v532	v863	vand	fntk
xiao ching	x250	x252	x384	xaca	sxnk
asavakun	a250	a225	a380	asac	askn
prapasri	b612	p612	p913	prap	prps
von der felde	f531	v536	v869	vand	fntr
vest	f200	v230	v360	vast	fst
vest	v200	v230	v360	vast	ast
oioi	a000	o000	o000	o	a
ohio	a000	o000	o000	o	ah
oihcca	a200	o200	o300	oc	ak
nielsen	n250	n425	n738	nals	nlsn
kin	c500	k500	k800	can	kn
lin	l500	l500	l800	lan	ln
computer	c513	c513	c816	canp	knpt
record	r200	r263	r396	raca	rkrt
linkage	l520	l522	l834	lanc	lnkj
probabilistic	b614	p611	p917	prab	prpp

Σχήμα 4 – Φωνητική Κωδικοποίηση

Το Febrl χρησιμοποιεί αλγορίθμους ευρετηριοποίησης για την υλοποίηση των μεθόδων Blocking που εφαρμόζει. Οι μέθοδοι που χρησιμοποιούνται από το Febrl είναι αυτή του Τυποποιημένου Blocking που περιγράφηκε αναλυτικά στην παράγραφο 3.2.1, αυτή της Ταξινομημένης Γειτονιάς που μελετήθηκε στην παράγραφο 3.2.2 και η Ευρετηριοποίηση Διγραμμάτων για την οποία μιλήσαμε στην παράγραφο 3.2.3. Για να γίνουν πιο κατανοητές οι δύο τελευταίες μέθοδοι θα αναφέρουμε παρακάτω ένα παράδειγμα για την κάθε μία αντίστοιχα.

Δεδομένου ότι χρησιμοποιούμε τη μέθοδο της Ταξινομημένης Γειτονιάς, υποθέτουμε ότι υπάρχουν συνολικά έξι blocks όπου μέσα στα blocks φαίνεται ο υποτιθέμενος κωδικός της εγγραφής. Αυτά τα blocks ταξινομούνται ως εξής:
a123: [4, 12, 89, 99]
a129: [6, 32, 54, 84, 91]
a245: [1, 39]
a689: [3, 17, 21, 35, 49, 76, 87, 93]
a911: [2, 42, 66]
b111: [8]
Αν ορισθεί το μέγεθος του παραθύρου ως τρία, τότε θα γίνει η σύγκριση των εγγραφών που ανήκουν στα μπλοκ:
[a123, a129, a245]: [1, 4, 6, 12, 32, 39, 54, 84, 89, 91, 99]
[a129, a245, a689]: [1, 3, 6, 17, 21, 32, 35, 39, 49, 54, 76, 84, 87, 91, 93]
[a245, a689, a911]: [1, 2, 3, 17, 21, 35, 39, 42, 49, 66, 76, 87, 93]
[a689, a911, b111]: [2, 3, 8, 17, 21, 35, 42, 49, 66, 76, 87, 93]
Αν το μέγεθος του παραθύρου είναι 1 τότε το ευρετήριο ταξινόμησης είναι το ίδιο με το blocking ευρετήριο.

Χρησιμοποιώντας την Ευρετηριοποίηση Δι-γραμμάτων, θεωρούμε ότι ένα block ορίζεται ως εξής:

block_definition = [[('givname','direct')], ...]

και το κατώφλι παίρνει την τιμή 0.8. Εάν η τιμή 'peter' δοθεί στο πεδίο givname, η αντίστοιχη λίστα bigrams αποτελείται από τα εξής τέσσερα στοιχεία ['pe', 'et', 'te', 'er']. Η λίστα ταξινομείται ως εξής ['er', 'et', 'pe', 'te'], και χρησιμοποιώντας τιμή κατωφλίου 0.8 καταλήγουμε στο 4*0.8=3.2 που στρογγυλοποιείται στο 3, και το οποίο σημαίνει ότι υπολογίζονται συνδυασμοί μήκους 3. Για το συγκεκριμένο παράδειγμα έχουμε:

['et', 'pe', 'te']
['er', 'pe', 'te']
['er', 'et', 'te']
['er', 'et', 'pe']

Έτσι, ο αντίστοιχος αριθμός εγγραφής θα εισέλθει στο ανεστραμμένο ευρετήριο blocks με κλειδιά 'etpete', 'erpete', 'erette', και 'eretpe'.

4.4.2 Οι Συναρτήσεις Σύγκρισης Πεδίων του Febrl

Το Febrl διαθέτει αρκετές συναρτήσεις σύγκρισης των εγγράφων οι οποίες είναι υλοποιημένες στην εφαρμογή 'comparison.py'[16]. Αυτές ελέγχουν ξεχωριστά τα πεδία εγγραφών και ανάλογα με την συνάρτηση, μπορούν να συγκρίνουν αλφαριθμητικά, αριθμούς, ημερομηνίες, ηλικίες και χρονικές περιόδους.

Κάθε συνάρτηση συγκρίνει τις εγγραφές και παράγει τα διανύσματα βάρους τα οποία ουσιαστικά περιλαμβάνουν τιμές που δείχνουν κατά πόσο δύο εγγραφές ταιριάζουν ή όχι. Τα διανύσματα βάρους παράγονται από τη χρήση ορισμένων πιθανοτικών μοντέλων τα οποία δίνουν κάποια ποσοστά (τιμές από 0 ως 1) ομοιότητας των εγγραφών. Παρακάτω αναφέρονται ενδεικτικά ορισμένες από τις συναρτήσεις που είναι υλοποιημένες στο λογισμικό :

- Υπολογισμός βάρους βασισμένος στην συχνότητα του πεδίου.

- Ακριβής σύγκριση ολόκληρης της τιμής του πεδίου ή τμημάτων της.
- Προσεγγιστική σύγκριση της τιμής του πεδίου.
- Σύγκριση κωδικοποιημένης τιμής πεδίου.
- Αριθμητική σύγκριση με ποσοστιαία ανθεκτικότητα.
- Αριθμητική σύγκριση με απόλυτη ανθεκτικότητα.

4.4.3 Οι Κατηγοριοποιητές του Febrl

Το τελευταίο στάδιο της διασύνδεσης εγγραφών είναι η εξέταση των διανυσμάτων βάρους από έναν κατηγοριοποιητή ο οποίος θα αποφανθεί για την ομοιότητα τους. Το λογισμικό Febrl χρησιμοποιεί δύο είδη κατηγοριοποιητών για να επιτύχει αυτόν τον σκοπό. Ακολουθεί μια μικρή αναφορά αυτών των δύο κατηγοριοποιητών, των Fellegi και Sunter και του Ευέλικτου Κατηγοριοποιητή.

- **Κατηγοριοποιητής Fellegi και Sunter [17] :** Αποτελεί έναν ιδιαίτερα απλό κατηγοριοποιητή, ο οποίος αποτιμά όλα τα βάρη που απαρτίζουν ένα διάνυσμα βάρους ενός ζεύγους εγγραφών και αφού υπολογίσει το άθροισμά τους, και χρησιμοποιώντας δύο κατώφλια, αποφασίζει αν οι εγγραφές ταιριάζουν, δεν ταιριάζουν ή πρέπει να παρεμβληθεί ο ανθρώπινος παράγοντας για να αποφασίσει. Όταν η αποτίμηση των βαρών είναι κάτω από το μικρότερο κατώφλι τότε οι εγγραφές δεν ταιριάζουν. Όταν είναι πάνω από το μεγαλύτερο κατώφλι τότε ταιριάζουν, ενώ αν είναι ανάμεσα στα δυο κατώφλια τότε δεν μπορεί να αποφανθεί οπότε χρειάζεται η ανθρώπινη παρέμβαση.
- **Ευέλικτος Κατηγοριοποιητής [1] :** Η διαφορά από τον προηγούμενο είναι ότι χρησιμοποιεί διάφορες μεθόδους για την αποτίμηση του συνολικού βάρους ενός ζεύγους εγγραφών. Ενδέχεται από ένα διάνυσμα βάρους να εξεταστεί μόνο το βάρος με την ελάχιστη ή τη μέγιστη τιμή, το άθροισμα όλων ή μερικών βαρών, το γινόμενο τους, ή ο μέσος όρος τους. Παράλληλα είναι δυνατόν να γίνει συνδυασμός των παραπάνω μεθόδων. Όπως και ο προηγούμενος, βασίζεται στη χρήση δύο κατωφλίων για να πάρει την τελική απόφαση ομοιότητας.

Κεφάλαιο 5

Περιγραφή, Αξιολόγηση και Αποτίμηση των Πειραμάτων

5.1 Γενικά

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση όλων των πειραμάτων που υλοποιήθηκαν στα πλαίσια αυτής της εργασίας, προκειμένου να βγουν κάποια συμπεράσματα σχετικά με τη σύγκριση των Blocking μεθόδων που βρίσκουν εφαρμογή στο σύστημα Febrl. Θα γίνει μια πλήρης επεξήγηση του προγράμματος που χρησιμοποιήθηκε και είναι υλοποιημένο στο σύστημα Febrl και όλων των απαραίτητων τροποποιήσεων που έγιναν, προκειμένου να επιτευχθεί μια λεπτομερής μελέτη των μεθόδων. Θα αναφερθεί η μεθοδολογία που ακολουθήθηκε, για να αποφασιστεί ποιες θα ήταν οι κατάλληλες παράμετροι που θα έπρεπε να υπολογιστούν σε κάθε μέθοδο, ώστε να προκύψουν πιο αντιπροσωπευτικά αποτελέσματα.

Στη συνέχεια, θα γίνει η αξιολόγηση και η αποτίμηση αυτών των πειραμάτων. Η συγκεκριμένη αξιολόγηση αφορά τόσο το χρόνο εκτέλεσης των πειραμάτων, όσο και την ποιότητα των αποτελεσμάτων που προσφέρουν. Έτσι, όσον αφορά το χρόνο εκτέλεσης των πειραμάτων, ουσιαστικά εννοείται ο χρόνος που χρειάζεται να τρέξει το πρόγραμμα που υλοποιεί τη διαδικασία της αναζήτησης και απαλοιφής των διπλοτύπων για την κάθε μέθοδο ξεχωριστά. Όσον αφορά την ποιότητα, θα δοθούν κάποια στοιχεία ώστε να φανεί ποια από τις τρεις μεθόδους που συγκρίνονται, μπορεί να φέρει σε πέρας την διαδικασία της απαλοιφής διπλοτύπων με μεγαλύτερη επιτυχία.

Πριν προχωρήσουμε με λεπτομέρειες στην περιγραφή, αξιολόγηση και αποτίμηση των πειραμάτων θα πρέπει να αναφερθεί το υπολογιστικό σύστημα που χρησιμοποιήθηκε για τις ανάγκες της έρευνας. Συγκεκριμένα, χρησιμοποιήθηκε ένας προσωπικός υπολογιστής με επεξεργαστή AMD Athlon 3200+ στα 2.01GHz, με μνήμη RAM 512 MB ενώ το λειτουργικό σύστημα του υπολογιστή είναι Windows XP.

5.2 Περιγραφή Πειραμάτων

5.2.1 Τεχνητά Σύνολα Δεδομένων

Όπως έχει ήδη αναφερθεί στην παράγραφο 4.2, το Febrl παρέχει στους χρήστες την δυνατότητα παραγωγής τεχνητών συνόλων δεδομένων ανάλογα με τις απαιτήσεις τους. Το πρόγραμμα που επιτελεί αυτή τη διαδικασία είναι το 'generate.py'. Στη συγκεκριμένη εργασία για τις ανάγκες των πειραμάτων, χρειάστηκε να δημιουργηθούν έξι σύνολα. Δημιουργήθηκαν τέσσερα σύνολα με σχεδόν ανάλογες παραμέτρους το κάθε ένα, ώστε η κύρια διαφορά τους να έγκειται στο πλήθος των εγγραφών που αποτελούν το κάθε ένα. Τα υπόλοιπα δύο σύνολα δεδομένων δημιουργήθηκαν έχοντας διαφορά στο πλήθος των λαθών ανά εγγραφή.

Τα σύνολα δημιουργήθηκαν έτσι ώστε οι μισές από τις εγγραφές που περιέχουν να είναι οι διπλότυπες και οι υπόλοιπες μισές οι πρωτότυπες. Τα τέσσερα πρώτα σύνολα που δημιουργήθηκαν αποτελούνται συνολικά από 10000, 5000, 2000 και 1000 εγγραφές αντίστοιχα. Για κάθε πρωτότυπη εγγραφή δημιουργήθηκε ένα διπλότυπο στο οποίο είχε γίνει μία μόνο τροποποίηση σε σχέση με την πρωτότυπη. Το πιθανοτικό μοντέλο που

χρησιμοποιήθηκε σε όλα τα σύνολα δεδομένων, για τη δημιουργία των διπλότυπων εγγραφών είναι το Poisson [1].

Στη συνέχεια, δημιουργήθηκαν δύο ακόμα σύνολα δεδομένων των 5000 εγγραφών συνολικά το κάθε ένα, από τις οποίες οι 2500 είναι διπλότυπες. Για κάθε πρωτότυπη εγγραφή, αυτών των δύο συνόλων, δημιουργήθηκε μόνο ένα διπλότυπο, στο οποίο μπορούν να υπάρξουν μέχρι δύο τροποποιήσεις σε κάθε πεδίο του. Αυτό που διαφοροποιεί τα δύο αυτά σύνολα μεταξύ τους, είναι ότι σε κάθε εγγραφή μπορεί να συμβεί διαφορετικός αριθμός τροποποιήσεων και επομένως λαθών. Έτσι, στο ένα είναι πιθανό να υπάρχουν μέχρι τέσσερα λάθη ανά εγγραφή, ενώ στο άλλο μέχρι οκτώ λάθη.

Τα παραπάνω σύνολα δημιουργήθηκαν με αυτές τις παραμέτρους, ώστε η σύγκριση των Blocking μεθόδων να στηριχθεί στο πως συμπεριφέρεται η κάθε μια, για διαφορετικό πλήθος εγγραφών και για διαφορετικό πλήθος λαθών ανά εγγραφή. Επομένως, τα αποτελέσματα της σύγκρισης δεν θα έχουν να κάνουν, ούτε με τον αριθμό των διπλότυπων εγγραφών, εφόσον για όλα τα σύνολα δεδομένων ορίστηκε να είναι οι μισές εγγραφές, ούτε και με το πιθανοτικό μοντέλο δημιουργίας τους, εφόσον παντού χρησιμοποιήθηκε το Poisson. Στα πλαίσια της εργασίας έγινε προσπάθεια μελέτης μεγαλύτερων συνόλων δεδομένων, όπως είναι σύνολα με 20000 εγγραφές ακόμα και με 100000 εγγραφές. Προέκυψαν όμως προβλήματα, που αφορούσαν την μνήμη του υπολογιστή που έπρεπε να χρησιμοποιηθεί, με αποτέλεσμα να μην είναι εφικτή η μελέτη τους.

5.2.2 Τροποποιήσεις στο Πρόγραμμα Απαλοιφής Διπλοτύπων

Η σύγκριση των τριών Blocking μεθόδων που χρησιμοποιεί το Febrl (Τυποποιημένο Blocking, Ταξινομημένη Γειτονιά, Ευρετηριοποίηση Δι-γραμμάτων), έγινε με βάση το πρόγραμμα project-deduplicate.py, το οποίο αναγνωρίζει τα διπλότυπα σε ένα σύνολο δεδομένων. Πάνω σε αυτό το πρόγραμμα έγιναν οι απαραίτητες τροποποιήσεις, ώστε να προκύψουν κάποια συμπεράσματα για τις μεθόδους. Οι τροποποιήσεις που έγιναν, αφορούν τα σύνολα δεδομένων που παίρνει σαν είσοδο το πρόγραμμα, τις παραμέτρους των Blocking μεθόδων που χρησιμοποιεί και το ανώτερο και κατώτερο κατώφλι του κατηγοριοποιητή.

Βασικό κομμάτι του προγράμματος, αποτελεί ο ορισμός των παραμέτρων των Blocking μεθόδων καθώς και του κλειδιού που θα χρησιμοποιήσουν. Τα blocks καθορίζονται από μια λίστα που περιέχει υπολίστες με την εξής μορφή: (όνομα πεδίου, μέθοδος κωδικοποίησης, παράμετροι). Έτσι, για τα πειράματα αυτής της εργασίας, χρησιμοποιήθηκε και για τις τρεις μεθόδους το ακόλουθο block. Αποτελείται από τρεις υπολίστες, από τις οποίες η πρώτη βασίζεται στο συνδυασμό του επωνύμου, κωδικοποιημένο με τον αλγόριθμο Dmetaphone (μέγιστο μήκος κώδικα τεσσάρων χαρακτήρων), και του έτους γέννησης. Η δεύτερη συνδυάζει τους τρεις χαρακτήρες του ονόματος και τον κωδικό της περιοχής, ενώ η τρίτη το όνομα της περιοχής, κωδικοποιημένο με τον φωνητικό αλγόριθμο NYSIIS, και το μήνα γέννησης.

Εκτελώντας το πρόγραμμα του Febrl που κάνει απαλοιφή των διπλότυπων εγγραφών από ένα σύνολο δεδομένων, δημιουργούνται τρία αρχεία εξόδου. Στο πρώτο εμφανίζονται κάποια από τα ζευγάρια των εγγραφών που συγκρίνονται, με βάση την εκάστοτε Blocking μέθοδο, με τη μορφή τριών στηλών. Η πρώτη στήλη περιέχει τα ονόματα των πεδίων, η μεσαία τις τιμές των πεδίων αυτών της πρώτης εγγραφής ενώ η τρίτη τις τιμές των πεδίων αυτών για τη δεύτερη εγγραφή. Πόσα ζευγάρια θα εμφανιστούν, εξαρτάται από ένα κατώφλι που ορίζει ο χρήστης. Στο δεύτερο αρχείο εμφανίζεται ένα ιστόγραμμα, στο οποίο φαίνεται

πόσα ζευγάρια εγγραφών από αυτά που συγκρίθηκαν με την εκάστοτε μέθοδο, έχουν ένα συγκεκριμένο βάρος π.χ. 5 ζευγάρια έχουν βάρος 27. Και πάλι όμως, τα ζευγάρια που εμφανίζονται είναι αυτά που έχουν βάρος μεγαλύτερο από το κατώφλι που έχει θέσει ο χρήστης. Στο τρίτο αρχείο εμφανίζονται οι αριθμοί ταυτοποίησης των εγγραφών που συγκρίθηκαν και τα αντίστοιχα βάρη τους.

Αυτό που γίνεται ουσιαστικά με τα παραπάνω τρία αρχεία είναι ότι, αφού κατηγοριοποιηθούν τα ζευγάρια εγγραφών που συγκρίθηκαν και τα οποία έχουν προκύψει από κάποια από τις μεθόδους Blocking, αποθηκεύονται σε αυτά μόνο οι συγκρίσεις που έχουν ένα τελικό βάρος υψηλότερο από το κατώφλι που θέτει ο χρήστης. Έτσι αν το κατώφλι πάρει την τιμή του κατώτερου κατωφλίου του κατηγοριοποιητή, εμφανίζονται μόνο τα ζευγάρια που ταιριάζουν (αυτά δηλαδή, με βάρος μεγαλύτερο από το μεγαλύτερο κατώφλι) και τα ζευγάρια που είναι πιθανόν να ταιριάζουν (αυτά που το βάρος τους είναι ανάμεσα στο μικρότερο και το μεγαλύτερο κατώφλι). Τα ζευγάρια που δεν ταιριάζουν, ενώ ο αριθμός τους υπολογίζεται στις συνολικές συγκρίσεις, απορρίπτονται από το σύστημα. Επομένως, δεν είναι δυνατόν να γνωρίζουμε ούτε τον ακριβή αριθμό των ζευγαριών εγγραφών που δεν ταιριάζουν και έχουν κατηγοριοποιηθεί σαν ζευγάρια που δεν ταιριάζουν (true non-matches), ούτε τα ζευγάρια που ταιριάζουν και έχουν κατηγοριοποιηθεί σαν ζευγάρια εγγραφών που δεν ταιριάζουν (false non-matches). Γνωρίζουμε μόνο το συνολικό αριθμό των ζευγαριών εγγραφών που δεν ταιριάζουν. Αν θέσουμε στην τιμή του κατωφλίου την τιμή του μεγαλύτερου κατωφλίου του κατηγοριοποιητή, τότε στα αρχεία εξόδου εμφανίζονται μόνο τα ζευγάρια των εγγραφών που θεωρούνται ότι ταιριάζουν (matches).

Για κάθε ένα από τα έξι σύνολα δεδομένων που περιγράφηκαν στην παράγραφο 5.2.1, αναλύονται οι τροποποιήσεις που έγιναν στο πρόγραμμα. Για όλα τα σύνολα δεδομένων χρησιμοποιήθηκε ο ίδιος κατηγοριοποιητής, αυτός των Fellegi και Sunter, ενώ σαν κατώτερο κατώφλι χρησιμοποιήθηκε η τιμή 10.0 και σαν ανώτερο η τιμή 30.0. Αλλάζοντας τον κατηγοριοποιητή και χρησιμοποιώντας εναλλακτικά τον Ευέλικτο Κατηγοριοποιητή δεν υπήρξαν σημαντικές διαφορές στα αποτελέσματα. Οι τιμές αυτές επιλέχθηκαν έτσι, επειδή παρατηρήθηκε ότι το βάρος των περισσοτέρων ζευγαριών εγγραφών που συγκρίνονται, κυμαίνονταν ανάμεσα σε αυτές τις τιμές και επομένως, δεν θα χάνονταν πολλά ζευγάρια. Η τιμή του κατωφλίου που ορίζει ποια ζευγάρια θα εμφανιστούν στα αρχεία εξόδου τέθηκε να είναι ίση με το κατώτερο κατώφλι του κατηγοριοποιητή. Έτσι, αποθηκεύονται σε αυτά, τα ζευγάρια που είναι πιθανόν να ταιριάζουν (possible matches) και τα ζευγάρια που ταιριάζουν.

Στο Τυποποιημένο Blocking ο αριθμός των ζευγών εγγραφών, που πρόκειται να συγκριθούν, εξαρτάται από τον αριθμό των blocks και το μέγεθός τους, γι' αυτό και οι τροποποιήσεις που έγιναν αφορούν το μέγεθος του block. Έτσι, εκτός από το block που αναφέρθηκε παραπάνω και χρησιμοποιήθηκε για όλες τις Blocking μεθόδους, δημιουργήθηκε και ένα νέο μικρότερο block. Αυτό αποτελείται από μια υπολίστα η οποία βασίζεται στο συνδυασμό του επωνύμου, κωδικοποιημένο με τον αλγόριθμο Dmetaphone (μέγιστο μήκος κώδικα έξι χαρακτήρων), και του έτους γέννησης. Τα αποτελέσματα που προέκυψαν για την μέθοδο του Τυποποιημένου Blocking στηρίχθηκαν σε αυτήν την παράμετρο. Στη μέθοδο της Ταξινομημένης Γειτονιάς, το πλήθος των συγκρίσεων καθορίζεται από το μέγεθος του παραθύρου. Για τα πειράματα της συγκεκριμένης εργασίας δοκιμάστηκαν διαφορετικές τιμές του μεγέθους του παραθύρου ($w=5$, $w=10$, $w=20$), ώστε να φανεί πως συμπεριφέρεται η μέθοδος στα διαφορετικά παράθυρα και να καθοριστεί ποιο μέγεθος παραθύρου οδηγεί στα καλύτερα αποτελέσματα. Όταν το μέγεθος του παραθύρου

πάρει την τιμή ένα τότε συμπεριφέρεται όπως η μέθοδος του Τυποποιημένου Blocking. Στην Ευρετηριοποίηση Δι-γραμμάτων οι τροποποιήσεις αφορούν την τιμή του κατωφλίου (*th=0.3, th=0.6, th=0.9*). Το κατώφλι μπορεί να πάρει τιμές από μηδέν, αλλά όχι ακριβώς μηδέν, μέχρι και ένα όπου συμπεριφέρεται όπως η μέθοδος του Τυποποιημένου Blocking.

5.2.3 Μετρήσεις των Κριτηρίων Αποδοτικότητας των Blocking Μεθόδων

Όπως είδαμε στην παράγραφο 3.4, τρία είναι τα βασικά κριτήρια πολυπλοκότητας που καθορίζουν την αποδοτικότητα και την ποιότητα των Blocking μεθόδων. Η αναλογία μείωσης (*RR*) ορίζεται ως $RR=1-s/N$, όπου s είναι ο αριθμός των ζευγαριών εγγραφών που παράγονται από μια Blocking μέθοδο, και N είναι ο συνολικός αριθμός των πιθανών ζευγαριών εγγραφών που πρόκειται να συγκριθούν. Η πληρότητα ζευγαριών εγγραφών (*PC*) ορίζεται ως $PC=s_M/N_M$, όπου s_M είναι ο αριθμός των αληθινά ταιριασμένων ζευγαριών εγγραφών που παράγονται για σύγκριση από την Blocking μέθοδο, και N_M είναι ο αριθμός των αληθινά ταιριασμένων ζευγών εγγραφών σε όλο το σύνολο δεδομένων. Το *Fscore* συνδυάζει το *RR* και το *PC* μέσω της σχέσης $Fscore=2\times PC\times RR/(PC+RR)$.

Το σύστημα όπως ήδη αναφέρθηκε, εμφανίζει μόνο τα ζευγάρια των εγγραφών που ταιριάζουν και τα ζευγάρια εγγραφών που είναι πιθανόν να ταιριάζουν. Από αυτά τα ζευγάρια πρέπει ο χρήστης να εξετάσει πόσα ταιριάζουν πραγματικά ώστε να υπολογιστεί το *PC*. Ο αριθμός τους εξαρτάται από το κατώφλι του κατηγοριοποιητή που θέτει ο χρήστης.

Παρακάτω παρουσιάζονται σε πίνακες οι τιμές των κριτηρίων πολυπλοκότητας και το σύνολο των συγκρίσεων για κάθε ένα από τα τέσσερα σύνολα δεδομένων (διαφορετικό πλήθος εγγραφών) και ξεχωριστά για κάθε μια από τις τρεις μεθόδους, δηλαδή για το Τυποποιημένο Blocking (Sb), την Ταξινομημένη Γειτονιά (SN), και την Ευρετηριοποίηση Δι-γραμμάτων (BI). Πρέπει να σημειωθεί ότι όλοι οι υπολογισμοί έγιναν με ακρίβεια τεσσάρων δεκαδικών ψηφίων. Αρχικά, στους τρεις ακόλουθους πίνακες, Πίνακας 5, Πίνακας 6 και Πίνακας 7, φαίνονται οι υπολογισμοί για το σύνολο δεδομένων που αποτελείται από 1000 εγγραφές και πως διαμορφώνονται με την κάθε μέθοδο. Επίσης, στους πίνακες φαίνεται και πως επηρεάζεται κάθε μέθοδος από τις διαφορετικές τιμές των παραμέτρων της.

	Σύνολο Συγκρίσεων	**RR**	**PC**	**Fscore**
1° Blocking κλειδί	710	0.9985	1.0	0.9992
2° Blocking κλειδί	454	0.9990	0.8280	0.9054

Πίνακας 5 – Μετρήσεις με τη μέθοδο Sb για 1000 εγγραφές

	Σύνολο Συγκρίσεων	**RR**	**PC**	**Fscore**
w=5	21770	0.9564	1.0	0.9777
w=10	47064	0.9057	1.0	0.9505
w=20	94436	0.8109	1.0	0.8955

Πίνακας 6 - Μετρήσεις με τη μέθοδο SN για 1000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	130365	0.7390	0.9980	0.8491
th=0.6	10235	0.9795	0.9980	0.9886
th=0.9	849	0.9983	0.9980	0.9981

Πίνακας 7 - Μετρήσεις με τη μέθοδο BI για 1000 εγγραφές

Ακολουθούν οι αντίστοιχοι τρεις πίνακες για το σύνολο δεδομένων με 2000 εγγραφές, Πίνακας 8, Πίνακας 9 και Πίνακας 10.

	Σύνολο Συγκρίσεων	RR	PC	Fscore
1° Blocking κλειδί	1788	0.9991	1.0	0.9995
2° Blocking κλειδί	1005	0.9994	0.8100	0.8947

Πίνακας 8 - Μετρήσεις με τη μέθοδο Sb για 2000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
w=5	46007	0.9769	1.0	0.9883
w=10	100124	0.9499	1.0	0.9743
w=20	204124	0.8978	1.0	0.9461

Πίνακας 9 - Μετρήσεις με τη μέθοδο SN για 2000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	542193	0.7287	1.0	0.8430
th=0.6	48384	0.9757	1.0	0.9877
th=0.9	2469	0.9987	1.0	0.9993

Πίνακας 10 - Μετρήσεις με τη μέθοδο BI για 2000 εγγραφές

Όμοια οι αντίστοιχοι τρεις πίνακες, Πίνακας 11, Πίνακας 12, και Πίνακας 13 για το σύνολο δεδομένων με 5000 εγγραφές.

	Σύνολο Συγκρίσεων	RR	PC	Fscore
1° Blocking κλειδί	8056	0.9993	0.9992	0.9992
2° Blocking κλειδί	3467	0.9997	0.8148	0.8978

Πίνακας 11 - Μετρήσεις με τη μέθοδο Sb για 5000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
w=5	133721	0.9893	0.9996	0.9944
w=10	286535	0.9770	0.9996	0.9881
w=20	581424	0.9534	0.9996	0.9759

Πίνακας 12 - Μετρήσεις με τη μέθοδο SN για 5000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	3183318	0.7452	0.9996	0.8538
th=0.6	282090	0.9774	0.9996	0.9883
th=0.9	11635	0.9990	0.9992	0.9990

Πίνακας 13 - Μετρήσεις με τη μέθοδο BI για 5000 εγγραφές

Τέλος, ακολουθούν οι μετρήσεις στους τρεις παρακάτω πίνακες, Πίνακας 14, Πίνακας 15 και Πίνακας 16, για τις 10000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
1° Blocking κλειδί	26972	0.9994	0.9994	0.9994
2° Blocking κλειδί	10173	0.9997	0.8182	0.8998

Πίνακας 14 - Μετρήσεις με τη μέθοδο Sb για 10000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
w=5	327079	0.9934	0.9994	0.9963
w=10	683387	0.9863	0.9994	0.9928
w=20	1354832	0.9729	0.9994	0.9859

Πίνακας 15 - Μετρήσεις με τη μέθοδο SN για 10000 εγγραφές

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	13089956	0.7381	0.9996	0.8491
th=0.6	1136752	0.9772	0.9996	0.9882
th=0.9	41326	0.9991	0.9996	0.9993

Πίνακας 16 - Μετρήσεις με τη μέθοδο BI για 10000 εγγραφές

Στη συνέχεια θα δούμε, πως επηρεάζονται οι τιμές των κριτηρίων καθώς και το σύνολο των ζευγαριών εγγραφών που πρόκειται να συγκριθούν, από το πλήθος των λαθών ανά εγγραφή. Οι παρακάτω πίνακες, Πίνακας 17, Πίνακας 18, και Πίνακας 19, αφορούν ένα σύνολο δεδομένων 5000 εγγραφών με τέσσερα λάθη.

	Σύνολο Συγκρίσεων	RR	PC	Fscore
1° Blocking κλειδί	8975	0.9992	0.9200	0.9579
2° Blocking κλειδί	3382	0.9997	0.4972	0.6641

Πίνακας 17 - Μετρήσεις με τη μέθοδο Sb για εγγραφές με 4 λάθη

	Σύνολο Συγκρίσεων	RR	PC	Fscore
w=5	115445	0.9907	0.9580	0.9740
w=10	245530	0.9803	0.9656	0.9728
w=20	494570	0.9604	0.9684	0.9643

Πίνακας 18 - Μετρήσεις με τη μέθοδο SN για εγγραφές με 4 λάθη

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	2879335	0.7696	0.9656	0.8565
th=0.6	233584	0.9813	0.9488	0.9647
th=0.9	11453	0.9990	0.9168	0.9561

Πίνακας 19 - Μετρήσεις με τη μέθοδο BI για εγγραφές με 4 λάθη

Οι παρακάτω πίνακες, Πίνακας 20, Πίνακας 21 και Πίνακας 22, αφορούν ένα σύνολο δεδομένων 5000 εγγραφών με οκτώ λάθη.

	Σύνολο Συγκρίσεων	RR	PC	Fscore
1° Blocking κλειδί	10732	0,9991	0.6212	0.7660
2° Blocking κλειδί	2851	0,9997	0.2292	0.3729

Πίνακας 20 - Μετρήσεις με τη μέθοδο Sb για εγγραφές με 8 λάθη

	Σύνολο Συγκρίσεων	RR	PC	Fscore
w=5	111340	0.9910	0.7632	0.8623
w=10	229794	0.9816	0.7840	0.8174
w=20	448498	0.9641	0.7948	0.8713

Πίνακας 21 - Μετρήσεις με τη μέθοδο SN για εγγραφές με 8 λάθη

	Σύνολο Συγκρίσεων	RR	PC	Fscore
th=0.3	2470992	0.8022	0.7816	0.7917
th=0.6	216026	0.9827	0.7344	0.8405
th=0.9	12657	0.9989	0.6464	0.7848

Πίνακας 22 - Μετρήσεις με τη μέθοδο BI για εγγραφές με 8 λάθη

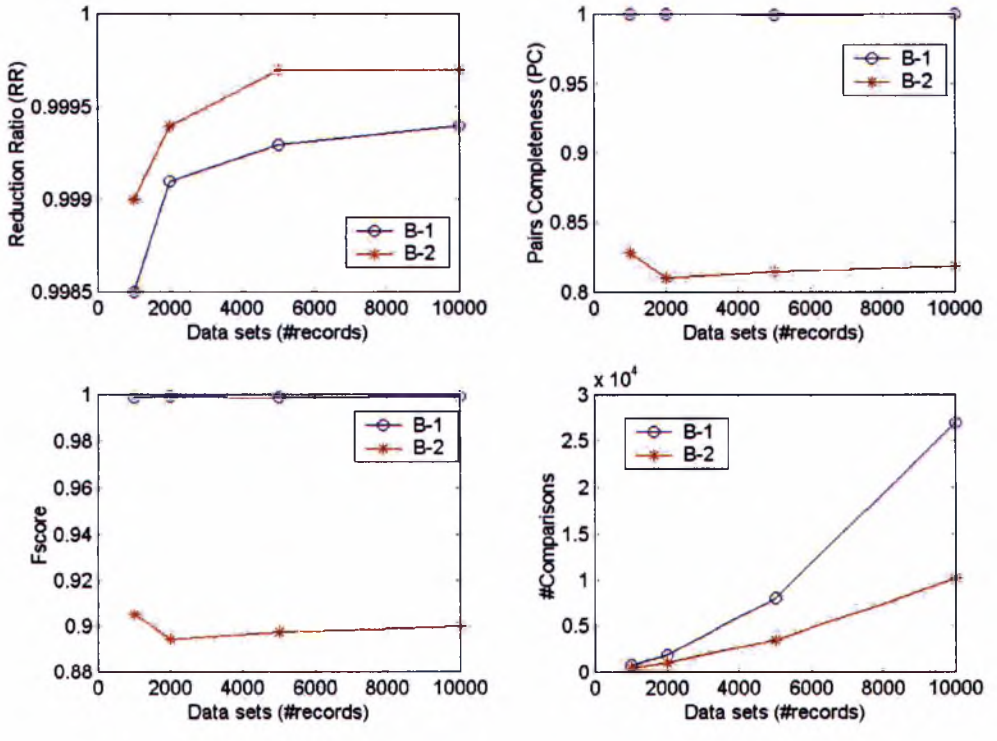
Στα παραπάνω πειράματα, διαπιστώνουμε πως για την περίπτωση όπου έχουμε μόνο ένα λάθος ανά εγγραφή η πληρότητα των ζευγαριών εγγραφών φτάνει έως και την μέγιστη τιμή της, δηλαδή *PC = 1*. Τώρα, όσον αφορά την *RR*, η τιμή της εξαρτάται από το πλήθος των συγκρίσεων που προκύπτει από την κάθε Blocking μέθοδο. Τέλος, όσον αφορά το *Fscore* οι τιμές του ποικίλουν εφόσον εξαρτώνται από το *PC* και το *RR*.

5.3 Αποτίμηση και Αξιολόγηση των Πειραμάτων

5.3.1 Αξιολόγηση της Ποιότητας των Πειραμάτων

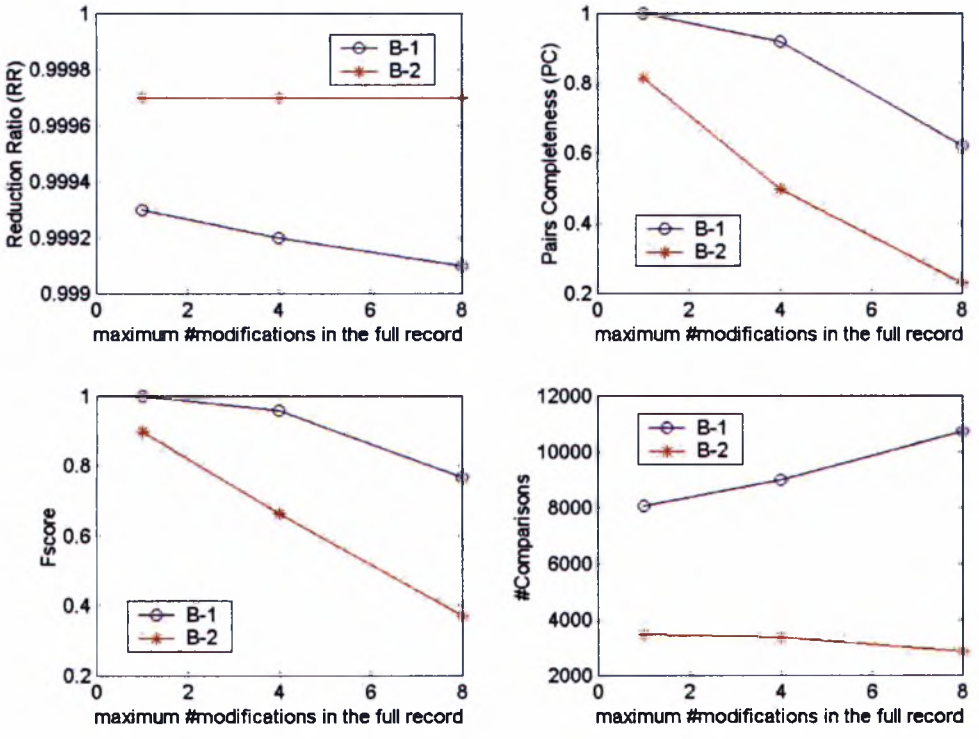
Σ' αυτήν την παράγραφο θα σχολιάσουμε τα αποτελέσματα που προκύπτουν από τις παραπάνω μετρήσεις, ώστε να μπορέσουμε να συμπεράνουμε κάποια στοιχεία για τις τρεις Blocking μεθόδους. Προκειμένου να έχουμε μια εικόνα της συμπεριφοράς των μεθόδων, με διαφορετικές παραμέτρους κάθε φορά, σε σχέση με το μέγεθος των συνόλων δεδομένων και σε σχέση με το πλήθος των λαθών ανά εγγραφή, κατασκευάσαμε κάποιες γραφικές παραστάσεις.

Αρχικά στο Σχήμα 5, παρατηρούμε πως κυμαίνονται τα τρία κριτήρια και το πλήθος των συγκρίσεων, χρησιμοποιώντας τη μέθοδο του Τυποποιημένου Blocking (Sb), καθώς αυξάνεται το μέγεθος των συνόλων δεδομένων (δηλαδή το πλήθος των εγγραφών). Προκειμένου να φανεί πότε η μέθοδος συμπεριφέρεται καλύτερα (δηλαδή σε ποια περίπτωση το *PC* και το *RR* παίρνουν μεγαλύτερες τιμές), χρησιμοποιήθηκαν τα δύο διαφορετικά μεγέθη block που περιγράφηκαν στην παράγραφο 5.2.2.



Σχήμα 5 – Συμπεριφορά της Sb για διαφορετικό πλήθος εγγραφών

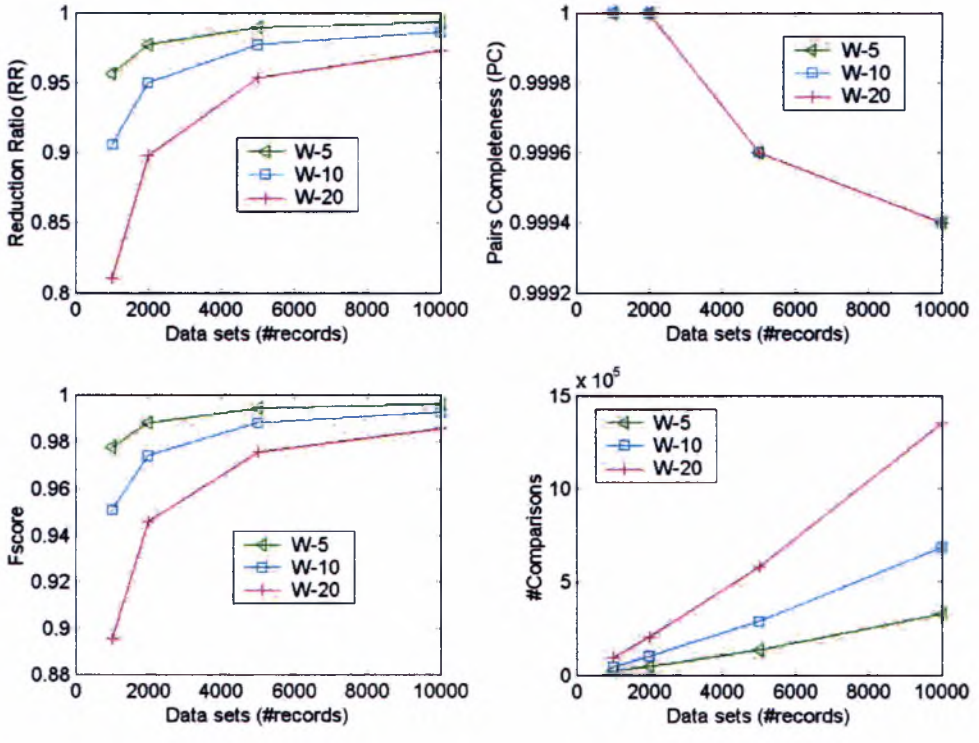
Στο Σχήμα 6 φαίνεται πως κυμαίνονται τα τρία κριτήρια και το πλήθος των συγκρίσεων, καθώς αυξάνεται το πλήθος των λαθών (ένα λάθος ανά εγγραφή, τέσσερα λάθη ανά εγγραφή και 8 λάθη ανά εγγραφή).



Σχήμα 6 – Συμπεριφορά της Sb για διαφορετικό πλήθος λαθών ανά εγγραφή

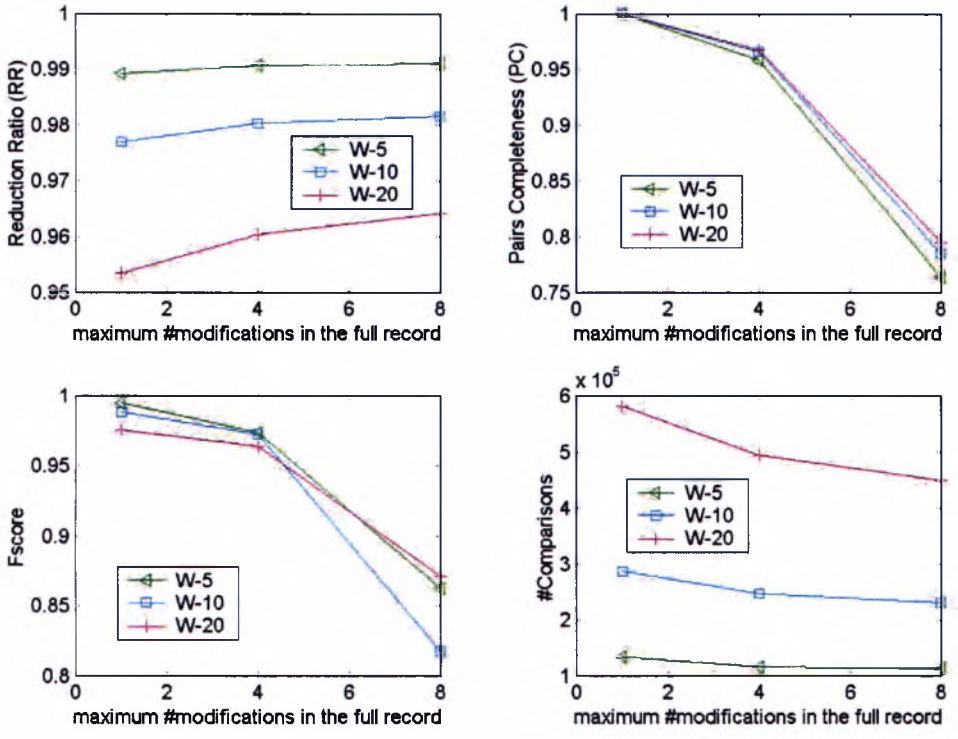
Αυτό που παρατηρείται είναι πως χρησιμοποιώντας μεγαλύτερο μέγεθος block (*B-1*) το *PC* παίρνει υψηλότερες τιμές. Απ' την άλλη, το *PC* μειώνεται όσο πληθαίνουν τα λάθη στα σύνολα των δεδομένων. Διαπιστώνεται τέλος, πως για την περίπτωση όπου έχουμε μόνο ένα λάθος ανά εγγραφή η πληρότητα των ζευγαριών εγγραφών φτάνει έως και την μέγιστη τιμή της, δηλαδή *PC=1*, για block μεγάλου μεγέθους. Όσον αφορά το *RR*, από τις γραφικές παραστάσεις φαίνεται, πως για μεγαλύτερο block, το *RR* μειώνεται. Επιπλέον το *RR* μειώνεται, καθώς αυξάνεται ο αριθμός των λαθών ανά εγγραφή ενώ αυξάνεται καθώς μεγαλώνει το μέγεθος των συνόλων δεδομένων. Το *Fscore* παίρνει τιμές που ποικίλουν εφόσον εξαρτώνται από το *PC* και το *RR*. Αυτό που μπορεί μόνο να σημειωθεί είναι ότι μειώνεται καθώς αυξάνεται το πλήθος των λαθών ανά εγγραφή. Τέλος, για το πλήθος των συγκρίσεων φαίνεται πως είναι μεγαλύτερο όσο μεγαλύτερο είναι το block.

Στη συνέχεια στο Σχήμα 7, φαίνονται οι γραφικές παραστάσεις των τριών κριτηρίων και του πλήθους των συγκρίσεων σε σχέση με το μέγεθος των συνόλων δεδομένων, χρησιμοποιώντας τη μέθοδο της Ταξινομημένης Γειτονιάς (SN). Για να παρατηρηθεί η καλύτερη συμπεριφορά της μεθόδου χρησιμοποιήθηκαν τρία διαφορετικά μεγέθη παραθύρου (*w=5, w=10, w=20*).



Σχήμα 7 – Συμπεριφορά της SN για διαφορετικό πλήθος εγγραφών

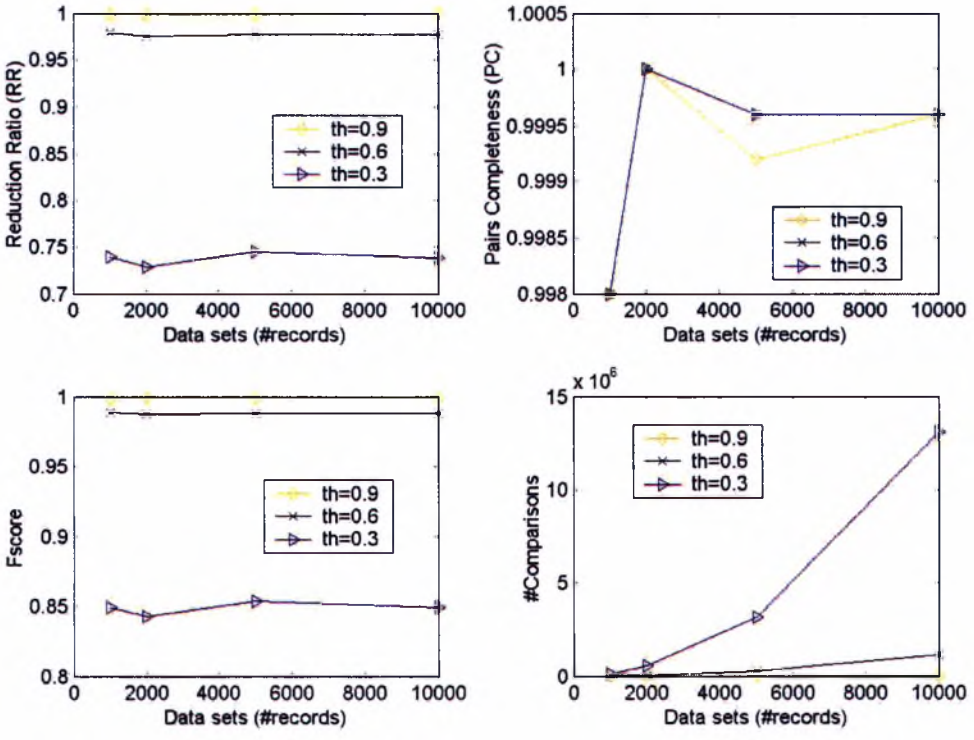
Ακολουθεί το Σχήμα 8, όπου φαίνεται πως κυμαίνονται οι τιμές των *PC*, *RR* και *Fscore* καθώς και το πλήθος των συγκρίσεων που προκύπτει, για τα τρία διαφορετικά μεγέθη παραθύρου, καθώς αυξάνεται ο αριθμός των λαθών ανά εγγραφή.



Σχήμα 8 – Συμπεριφορά της SN για διαφορετικό πλήθος λαθών ανά εγγραφή

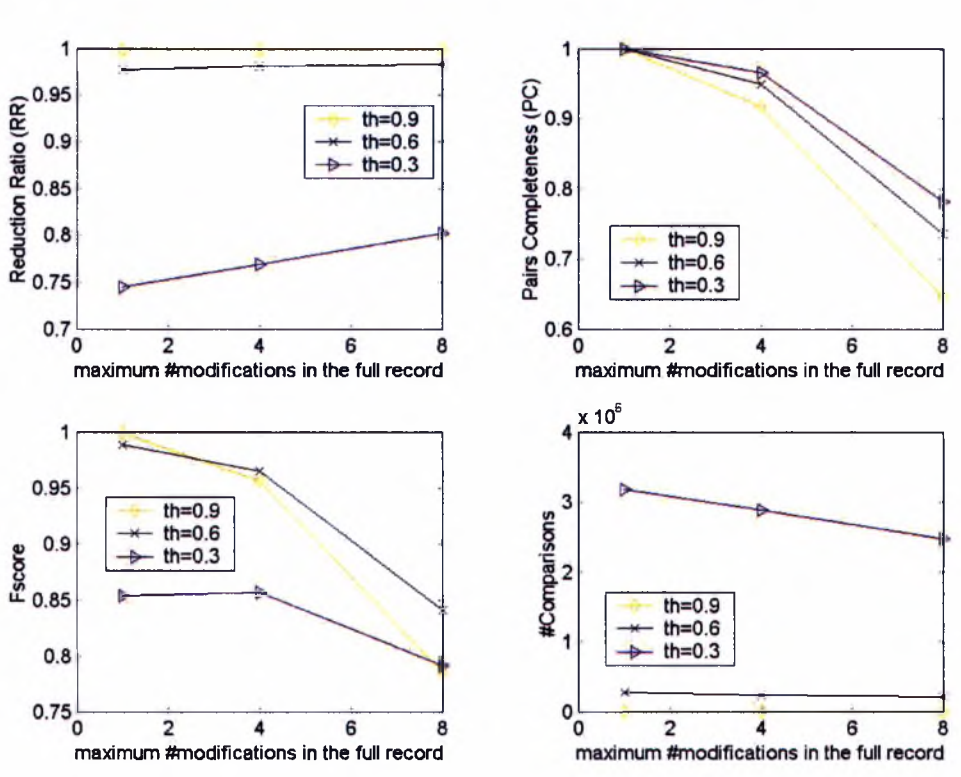
Φαίνεται λοιπόν, πως καθώς αυξάνεται το μέγεθος του παραθύρου το *PC* είτε θα αυξάνεται είτε θα παραμένει σταθερό. Συγκεκριμένα, αν υπάρχει μεγάλος αριθμός λαθών ανά εγγραφή, για μεγαλύτερο μέγεθος παραθύρου το *PC* παίρνει υψηλότερες τιμές, ενώ παραμένει σταθερό όταν το πλήθος των λαθών είναι μικρότερο. Επίσης, παρατηρείται πώς το *PC* μειώνεται καθώς αυξάνεται το πλήθος των εγγραφών και ο αριθμός των λαθών ανά εγγραφή. Όσον αφορά το *RR*, παίρνει χαμηλότερες τιμές όταν αυξάνεται το μέγεθος του παραθύρου ενώ αυξάνεται καθώς αυξάνεται το πλήθος των εγγραφών και το πλήθος των λαθών. Αυτό που μπορεί να σημειωθεί για το *Fscore* είναι ότι αυξάνεται καθώς αυξάνεται το πλήθος των εγγραφών ενώ παίρνει μεγαλύτερες τιμές για μικρότερο μέγεθος παραθύρου, όταν ο αριθμός των λαθών είναι μικρός. Τέλος όσον αφορά το πλήθος των συγκρίσεων, είναι μεγαλύτερο καθώς αυξάνεται το μέγεθος του παραθύρου.

Ακολουθούν οι γραφικές παραστάσεις για την Ευρετηριοποίηση Δι-γραμμάτων (BI). Έτσι, στο Σχήμα 9 παρατηρούμε τι κάνουν τα τρία κριτήρια καθώς και το πλήθος των συγκρίσεων, με διαφορετική τιμή κατωφλίου (*th=0.3, th=0.6, th=0.9*), καθώς αυξάνεται το πλήθος των εγγραφών.



Σχήμα 9 – Συμπεριφορά της BI για διαφορετικό πλήθος εγγραφών

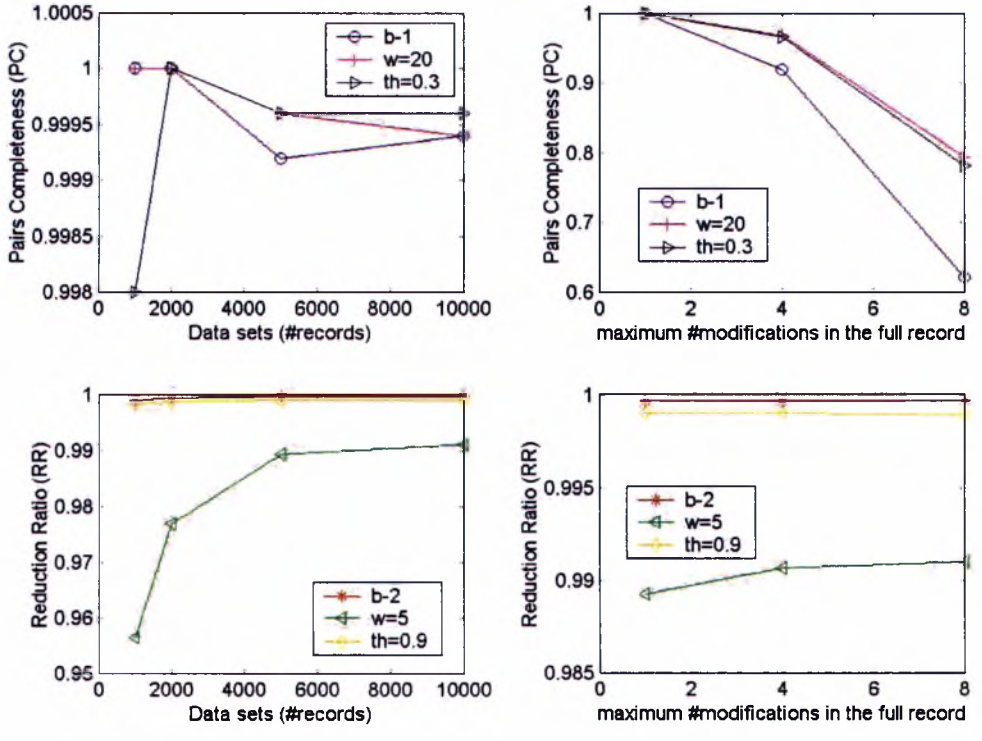
Στις παρακάτω γραφικές παραστάσεις, Σχήμα 10, παρατηρείται πως διαμορφώνονται οι τιμές των *PC*, *RR* και *Fscore* καθώς και το πλήθος των συγκρίσεων, για τα τρία διαφορετικά μεγέθη παραθύρου, καθώς αυξάνεται το πλήθος των λαθών ανά εγγραφή.



Σχήμα 10 – Συμπεριφορά της BI για διαφορετικό πλήθος λαθών ανά εγγραφή

Αυτό λοιπόν, που παρατηρείται είναι ότι για μεγαλύτερες τιμές του κατωφλίου το *PC* μειώνεται ή σε κάποιες περιπτώσεις παραμένει σταθερό. Συγκεκριμένα, παραμένει σταθερό όταν υπάρχει μικρό ποσοστό λαθών ανά εγγραφή. Επιπλέον, καθώς αυξάνεται το πλήθος των αλλαγών ανά εγγραφή, το *PC* μειώνεται. Τώρα σε σχέση με το *RR*, όσο αυξάνεται το κατώφλι αυξάνεται το *RR* και το ίδιο συμβαίνει καθώς αυξάνονται τα λάθη ανά εγγραφή. Το *Fscore* παίρνει τιμές ανάλογα με τις τιμές των *PC* και *RR*. Τέλος, το πλήθος των συγκρίσεων που προκύπτει με τη Ευρετηριοποίηση Δι-γραμμάτων αυξάνεται καθώς αυξάνεται το πλήθος των εγγραφών ενώ υψηλότερες τιμές παίρνει για μικρότερο κατώφλι.

Προκειμένου να βγουν τα τελικά συμπεράσματα για την σύγκριση των τριών μεθόδων Blocking που μελετήθηκαν μέχρι τώρα, επιλέχθηκαν οι παράμετροι για την κάθε μέθοδο που οδηγούν στην καλύτερη συμπεριφορά της, δηλαδή στην μέγιστη τιμή των *PC* και *RR*. Έτσι για τη μέθοδο του Τυποποιημένου Blocking, παρατηρήθηκε ότι μεγαλύτερές τιμές του *PC* πήραμε όταν το block ήταν μεγαλύτερο ενώ μεγαλύτερες τιμές για το *RR* πήραμε όταν το block ήταν μικρότερο. Για τη μέθοδο της Ταξινομημένης Γειτονιάς, το *PC* είχε υψηλότερες τιμές όταν το μέγεθος του παραθύρου πήρε την τιμή *w=20* ενώ το *RR* όταν *w=5*. Τέλος, για την Ευρετηριοποίηση Δι-γραμμάτων παρατηρήθηκε ότι υψηλότερες τιμές του *PC* σημειώθηκαν όταν το κατώφλι πήρε την τιμή *th=0.3* και του *RR* όταν *th=0.9*. Στο Σχήμα 11 φαίνονται οι γραφικές παραστάσεις των *PC* και *RR*, σε σχέση με το μέγεθος των συνόλων δεδομένων και του πλήθους των λαθών ανά εγγραφή, για τις τρεις μεθόδους.



Σχήμα 11 – Σύγκριση των μεθόδων Sb, SN, BI ως προς τα *PC* και *RR*

Αυτό που συμπεραίνουμε με μια πρώτη ματιά από τις παραπάνω γραφικές παραστάσεις είναι ότι προκειμένου να πετύχουμε μέγιστη πληρότητα ζευγαριών εγγραφών (*PC*) η καλύτερη μέθοδος είναι αυτή της Ταξινομημένης Γειτονιάς με μέγεθος παραθύρου *w=20* ενώ μέγιστη αναλογία μείωσης (*RR*) επιτυγχάνεται με τη μέθοδο του Τυποποιημένου Blocking με μικρό μέγεθος block, εφόσον το πλήθος των συγκρίσεων είναι μικρότερο.

5.3.2 Αξιολόγηση του Χρόνου Εκτέλεσης των Πειραμάτων

Όσον αφορά τον χρόνο εκτέλεσης των πειραμάτων θα παρουσιαστούν κάποια στατιστικά στοιχεία. Αυτά αφορούν τους χρόνους που χρειάζεται για να τρέξει το πρόγραμμα που υλοποιεί την διαδικασία της εξάλειψης των διπλοτύπων, για κάθε μέθοδο Blocking ξεχωριστά και για κάθε ένα από τα σύνολα δεδομένων που χρησιμοποιήθηκαν στα πειράματα. Έτσι οι χρόνοι εκτέλεσης συνοψίζονται στον παρακάτω πίνακα, Πίνακας 23.

		1000 εγγραφές (1 λάθος)	**2000 εγγραφές (1 λάθος)**	**5000 εγγραφές (1 λάθος)**	**5000 εγγραφές (4 λάθη)**	**5000 εγγραφές (8 λάθη)**	**10000 εγγραφές (1 λάθος)**
Sb	**1^{o} blocking κλειδί**	375 milli sec	968 milli sec	4.45 sec	5.08 sec	6.08 sec	15 sec
	2^{o} blocking κλειδί	202 milli sec	484 milli sec	1.72 sec	1.78 sec	1.53 sec	5.17 sec
SN	**w=5**	12 sec	26 sec	1 min και 18 sec	1 min και 8 sec	1 min και 4 sec	3 min και 13 sec
	w=10	28 sec	59 sec	2 min και 50 sec	2 min και 26 sec	2 min και 13 sec	6 min και 40 sec
	w=20	58 sec	2 min και 4 sec	5 min και 49 sec	4 min και 56 sec	4 min και 22 sec	13 min και 21 sec
BI	**th=0.3**	1 min και 22 sec	5 min και 39 sec	33 min και 18 sec	30 min και 10 sec	25 min και 30 sec	2 hrs και 23 min και 55 sec
	th=0.6	6.5 sec	30 sec	2 min και 55 sec	2 min και 25 sec	2 min και 11 sec	11 min και 36 sec
	th=0.9	483 milli sec	1.42 sec	6.88 sec	6.86 sec	7.5 sec	24 sec

Πίνακας 23 – Χρόνοι εκτέλεσης της διαδικασίας απαλοιφής διπλοτύπων

Από τον παραπάνω πίνακα παρατηρούμε πως ο χρόνος εκτέλεσης της διαδικασίας απαλοιφής διπλοτύπων αυξάνεται καθώς αυξάνεται το πλήθος των εγγραφών. Αυτό είναι λογικό εφόσον, καθώς αυξάνεται το μέγεθος των συνόλων δεδομένων αυξάνονται και οι συγκρίσεις που προκύπτουν από την κάθε μέθοδο. Για τη μέθοδο του Τυποποιημένου Blocking παρατηρούμε, πως ο χρόνος είναι μικρότερος όταν μικραίνει το block. Επίσης, για τη μέθοδο της Ταξινομημένης Γειτονιάς παρατηρούμε, πως ο χρόνος αυξάνεται καθώς μεγαλώνει το μέγεθος του παραθύρου ενώ μειώνεται όταν για ένα συγκεκριμένο αριθμό εγγραφών αυξάνονται τα λάθη ανά εγγραφή. Τέλος, για την Ευρετηριοποίηση Δι-γραμμάτων ο χρόνος εκτέλεσης μειώνεται, καθώς αυξάνεται το κατώφλι, εφόσον αυξάνεται το πλήθος των συγκρίσεων.

5.4 Συμπεράσματα

Με βάση την προηγούμενη αξιολόγηση των πειραμάτων οδηγούμαστε σε κάποια συμπεράσματα για την αποτελεσματικότητα της κάθε μεθόδου. Γενικά, αυτό που μπορεί να παρατηρηθεί, είναι πως το σύνολο των συγκρίσεων που προκύπτει με την εκάστοτε μέθοδο αυξάνεται καθώς αυξάνεται το πλήθος το δεδομένων, γεγονός αναμενόμενο. Επίσης, λογικό είναι πως για μεγαλύτερο αριθμό λαθών ανά εγγραφή, η ακρίβεια της μεθόδου, που στα πειράματα μας εκφράζεται μέσω της πληρότητας των ζευγαριών (*PC*), να μειώνεται. Στη συνέχεια, σ' αυτήν την παράγραφο θα αναφερθούν πιο αναλυτικά κάποια στοιχεία που παρατηρήθηκαν παραπάνω για την κάθε μέθοδο ξεχωριστά, και θα γίνει μια προσπάθεια επεξήγησή τους.

Αρχικά, στην παράγραφο 5.3.1 μελετήθηκε η συμπεριφορά του Τυποποιημένου Blocking. Ο αριθμός των συγκρίσεων των ζευγαριών εγγραφών, που δημιουργούνται με την μέθοδο αυτή, εξαρτάται από το μέγεθος των blocks που χρησιμοποιούνται. Έτσι, χρησιμοποιήθηκαν δύο διαφορετικά μεγέθη block, ώστε να διαπιστωθεί πότε η μέθοδος συμπεριφέρεται καλύτερα, πότε δηλαδή μεγιστοποιούνται οι τιμές των μέτρων που κρίνουν την απόδοσή της. Με το μικρό μέγεθος block (*B-2*), οι εγγραφές που περιέχονται σε κάθε block είναι λιγότερες, με αποτέλεσμα να μειώνεται η ακρίβεια της μεθόδου. Αυτό παρατηρήθηκε στα πειράματα μέσω του μέτρου της πληρότητας των ζευγαριών εγγραφών (*PC*). Συγκεκριμένα, το *PC* είχε πολύ μεγαλύτερες τιμές για το μεγάλο block (*B-1*) συγκριτικά με τις τιμές που προέκυψαν με το μικρό block (*B-2*). Οι λιγότερες συγκρίσεις εγγραφών που προκύπτουν με μικρά block, έχουν σαν αποτέλεσμα πολλά ζευγάρια εγγραφών που πραγματικά ταιριάζουν, να χάνονται. Έτσι μειώνεται και το *PC*. Εφόσον οι εγγραφές μέσα σε ένα block (*B-2*) μικρού μεγέθους είναι λίγες, ο αριθμός των συγκρίσεων που θα προκύψουν θα είναι κατά συνέπεια μικρός. Η αναλογία μείωσης (*RR*) εξαρτάται από το σύνολο των συγκρίσεων (όσο περισσότερες οι συγκρίσεις, τόσο μειώνεται το *RR*), οπότε μικρότερες τιμές παίρνει με τη χρησιμοποίηση του μεγάλου μεγέθους block (*B-1*). Τα αποτελέσματα, δηλαδή που προέκυψαν με την μέθοδο του Τυποποιημένου Blocking, ήταν αναμενόμενα.

Στη συνέχεια, μελετήθηκε η συμπεριφορά της μεθόδου της Ταξινομημένης Γειτονιάς. Είναι γνωστό ότι το μέγεθος του block εξαρτάται από το κλειδί και από το μέγεθος του παραθύρου. Οι συγκρίσεις που προκύπτουν από την μέθοδο, εξαρτώνται από το μέγεθος του block. Επομένως, εφόσον για τα πειράματά μας το κλειδί που χρησιμοποιείται είναι σταθερό, ο αριθμός των ζευγαριών εγγραφών που συγκρίνονται, καθορίζονται από το παράθυρο. Η χρήση του παραθύρου περιορίζει τον αριθμό των πιθανών συγκρίσεων για κάθε εγγραφή σε $2w-1$. Έτσι, για μεγάλο w, οι συγκρίσεις που θα προκύψουν θα είναι περισσότερες, εφόσον στο παράθυρο βρίσκονται περισσότερες εγγραφές. Περισσότερες συγκρίσεις οδηγούν σε μείωση του *RR*, γι' αυτό και οι υψηλότερες τιμές του *RR* προκύπτουν στα πειράματα μας όταν $w=5$. Επιπλέον, όπως αναφέρθηκε και προηγουμένως, αφού οι συγκρίσεις αυξάνονται με την αύξηση του w, θα αυξάνεται και η πληρότητα των ζευγαριών *PC*, εφόσον θα χάνονται λιγότερα σωστά ταιριασμένα ζευγάρια. Ωστόσο, στα πειράματα μας παρατηρήσαμε, ότι για κάποια σύνολα δεδομένων το *PC* παραμένει σταθερό καθώς αυξάνεται το w. Αυτό συμβαίνει είτε γιατί ακόμα και με μικρές τιμές του w, η πληρότητα των ζευγαριών για το συγκεκριμένο σύνολο έχει φτάσει την μέγιστη τιμή της, είτε γιατί ο αριθμός των λαθών ανά εγγραφή είναι μικρός.

Μετά ακολούθησαν οι γραφικές παραστάσεις που περιέγραψαν τη συμπεριφορά της Ευρετηριοποίησης Δι-γραμμάτων. Οι συγκρίσεις που προκύπτουν εξαρτώνται από το μέγεθος του block αλλά και από τον αριθμό τους. Το μέγεθος των blocks καθώς και ο αριθμός τους καθορίζεται από το κλειδί και από το κατώφλι. Επομένως, εφόσον για τα πειράματά μας το κλειδί που χρησιμοποιείται είναι σταθερό, ο αριθμός των ζευγαριών εγγραφών που συγκρίνονται, καθορίζονται μόνο από το κατώφλι. Ξέρουμε ότι όσο χαμηλότερο είναι το κατώφλι, τόσο κοντύτερες είναι οι υπο-λίστες, καταλήγοντας σε μικρότερα αλλά και περισσότερα blocks. Άρα, αφού μικρή τιμή του κατωφλίου οδηγεί σε μικρά αλλά πολλά blocks, οι εγγραφές που βρίσκονται στο μικρό block θα είναι λίγες αλλά οι συνολικές συγκρίσεις που θα προκύψουν θα είναι πολλές λόγω του μεγάλου αριθμού των blocks. Το γεγονός αυτό οδηγεί στην αύξηση των ζευγαριών που πραγματικά ταιριάζουν, στη αύξηση δηλαδή του *PC*. Έτσι, το γεγονός ότι υψηλότερες τιμές για το *PC* λάβαμε με *th=0.3*, είναι λογικό. Αντίστοιχα, μεγάλες τιμές για το *RR* προκύπτουν καθώς αυξάνεται το κατώφλι, εφόσον έτσι μειώνονται οι συγκρίσεις, γεγονός που συμβαίνει με *th=0.9*. Ωστόσο, στα πειράματα μας παρατηρήσαμε ότι για κάποια σύνολα δεδομένων το *PC* παραμένει σταθερό καθώς αυξάνεται το *th*. Αυτό συμβαίνει είτε γιατί η πληρότητα των ζευγαριών για το συγκεκριμένο σύνολο έχει την μέγιστη τιμή της, είτε γιατί ο αριθμός των λαθών ανά εγγραφή είναι μικρός.

Τελικά, κάνοντας τις γραφικές παραστάσεις των δύο ενδεικτικών κριτηρίων της απόδοσης, με τις καλύτερες τιμές των παραμέτρων και για τις τρεις μεθόδους, καταλήξαμε πως καλύτερη μέθοδος ως προς την πληρότητα των ζευγαριών εγγραφών (*PC*) είναι η μέθοδος της Ταξινομημένης Γειτονιάς (με μεγάλο μέγεθος παραθύρου). Ως προς την αναλογία μείωσης (*RR*), η αποδοτικότερη μέθοδος είναι το Τυποποιημένο Blocking (με μικρό μέγεθος block). Για την Ευρετηριοποίηση Δι-γραμμάτων οι μετρήσεις έδειξαν πως για μικρό κατώφλι, η πληρότητα των ζευγαριών εγγραφών έχει σχετικά υψηλές τιμές, καλύτερες από αυτές του Τυποποιημένου Blocking (με μεγάλο μέγεθος block). Για μεγάλο κατώφλι, η αναλογία μείωσης (*RR*) παρουσιάζει πολύ υψηλότερες τιμές σε σχέση με αυτές της Ταξινομημένης Γειτονιάς (με μικρό μέγεθος παραθύρου).

Κεφάλαιο 6

Επίλογος - Μελλοντική Εργασία

Η μελέτη που έγινε σ' αυτήν την εργασία αφορά ένα αντικείμενο που βρίσκεται σε συνεχής εξέλιξη. Η διασύνδεση των εγγραφών που βρίσκονται σε διαφορετικές βάσεις δεδομένων, καθώς και η αναζήτηση διπλοτύπων, είναι τομείς που παρουσιάζουν ιδιαίτερο ενδιαφέρον. Η παρούσα εργασία ασχολήθηκε σχεδόν αποκλειστικά με την μελέτη των Blocking μεθόδων που χρησιμοποιούνται στην διαδικασία ανίχνευσης διπλοτύπων, προκειμένου να μειωθεί ο αριθμός των ζευγαριών εγγραφών που συγκρίνονται. Δόθηκε ιδιαίτερη προσοχή στις Blocking μεθόδους που χρησιμοποιεί το Febrl ενώ μελετήθηκε η συμπεριφορά τους προκειμένου να βγουν κάποια συμπεράσματα για την αποτελεσματικότητα τους.

Πιο συγκεκριμένα στην εργασία αυτή αρχικά, έγινε μια αναφορά στα προβλήματα της διασύνδεσης εγγραφών, στις τεχνικές που χρησιμοποιούνται για την απαλοιφή διπλοτύπων, καθώς και στα κριτήρια απόδοσης τους. Στη συνέχεια, ενώ παρουσιάστηκαν κάποιες υπάρχουσες μέθοδοι Blocking, ιδιαίτερη σημασία δόθηκε στις Blocking μεθόδους που χρησιμοποιεί το σύστημα Febrl. Μελετήθηκε η λειτουργία του συστήματος Febrl πιο γενικά, ενώ ουσιαστικά το κύριο τμήμα της εργασίας αποτέλεσε η διεξαγωγή κάποιων πειραματικών δοκιμών σχετικά με τις τρεις Blocking μεθόδους (Τυποποιημένο Blocking, Ταξινομημένη Γειτονιά, Ευρετηριοποίηση Δι-γραμμάτων) που αυτό χρησιμοποιεί. Δημιουργήθηκαν κάποια σύνολα δεδομένων με συγκεκριμένες ιδιότητες και με συγκεκριμένο αριθμό διπλοτύπων στο κάθε ένα. Εκτελέσθηκε η διαδικασία της αναζήτησης των διπλοτύπων χρησιμοποιώντας μια από τις τρεις Blocking μεθόδους κάθε φορά. Μελετήθηκε η συμπεριφορά τους για διαφορετικές παραμέτρους, καταλήγοντας στην επιλογή εκείνης της παραμέτρου, που προσέφερε τη μεγιστοποίηση των τιμών των κριτηρίων αποτελεσματικότητας των μεθόδων.

Ωστόσο, κατά τη διάρκεια της έρευνας παρουσιάστηκαν και κάποια προβλήματα. Αυτό που κυρίως διαπιστώθηκε, είναι ότι δεν υπήρχαν παρόμοιες μελέτες με το ίδιο αντικείμενο. Το Febrl είναι ένα σύστημα που δεν βρίσκεται στην τελική του μορφή, αλλά δοκιμάζεται ακόμα, οπότε είναι λογικό να μην είναι ευρέως γνωστό και να μην έχει μελετηθεί λεπτομερώς. Έτσι, υπήρξαν αρχικά κάποια προβλήματα σχετικά με τη λειτουργία του. Επίσης, ένα πρόβλημα που διαπιστώθηκε είναι ότι, εφόσον το σύστημα κατηγοριοποιεί κάποια ζευγάρια εγγραφών ως πιθανώς ταιριασμένα, έγκειται στο χρήστη να μετρήσει ποια από τα πιθανά ζευγάρια ταιριάσματος είναι όντως σωστά, μια διαδικασία αρκετά επίπονη και χρονοβόρα για μεγάλα σύνολα δεδομένων. Έτσι, τα σύνολα δεδομένων που μελετήθηκαν έφταναν το μέγεθος των 10000 εγγραφών.

Όπως αναφέρθηκε παραπάνω, η σύγκριση περιορίστηκε στις μεθόδους Blocking που χρησιμοποιεί το Febrl μέχρι τώρα. Θα ήταν πολύ ενδιαφέρον να μελετηθούν κάποια στιγμή και άλλες μέθοδοι, αφού πρώτα υλοποιούνταν στη γλώσσα Python ώστε να είναι συμβατές με το σύστημα Febrl. Επιπλέον, κάτι που θα είχε εξαιρετική σημασία θα ήταν οι πειραματικές δοκιμές που έγιναν, να αφορούσαν και μη τεχνητά σύνολα δεδομένων, ώστε να μελετηθεί η συμπεριφορά των μεθόδων και με δεδομένα που δεν γνωρίζουμε το είδος ή τον αριθμό των λαθών που περιέχουν. Θα μπορούσε τέλος κάποιος, να υπολογίσει κάποια άλλα κριτήρια απόδοσης των μεθόδων και να στηρίξει τη σύγκριση τους σ' αυτά.

Καταλήγοντας, η κύρια συνεισφορά αυτής της μελέτης, αφορά την σύγκριση των τριών μεθόδων Blocking του Febrl βασισμένη ουσιαστικά, σε δύο σημαντικά μέτρα της απόδοσης τους. Αυτά τα μέτρα είναι η αναλογία μείωσης και η πληρότητα των ζευγαριών εγγραφών, δύο κριτήρια ενδεικτικά της συμπεριφοράς των μεθόδων που θελήσαμε να συγκρίνουμε. Στα κριτήρια αυτά στηρίχθηκε η σύγκριση τους και η διεξαγωγή συμπερασμάτων. Η σύγκριση των μεθόδων τέλος, μελετήθηκε σε σχέση με το διαφορετικό μέγεθος των συνόλων δεδομένων, που χρησιμοποιήθηκαν στα πειράματα που έγιναν, με το διαφορετικό πλήθος των λαθών ανά εγγραφή, που υπήρξαν στα σύνολα δεδομένων, και με τις διαφορετικές παραμέτρους που επηρεάζουν την κάθε μια από αυτές τις μεθόδους.

Βιβλιογραφία

1. P. Christen, and T. Churches, "Febrl – Freely extensible biomedical record linkage: Release 0.3", Documentation, Australian National University, April 6 2005.

2. R. Baxter, P. Christen, and T. Churches, "A Comparison of Fast Blocking Methods for Record Linkage", in Proc. of ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington, DC, USA, August 2003.

3. A. Elmagarmid, P. G. Ipeirotis, V. Verykios, "Duplicate Record Detection: A Survey", in Knowledge and Data Engineering, IEEE Transactions on, Jan. 2007

4. L. Gu, R. Baxter, "Adaptive Filtering for Efficient Record Linkage", 2004 SIAM Int. Conf. on Data Mining, April 22-24, Orlando, Florida.

5. L. Gu, R. Baxter, D. Vickers and C. Rainsford, "Record Linkage: Current Practice and Future Directions", Technical Report 03/83, April 2003, CSIRO Mathematical and Information Sciences, GPO Box 664, Canberra 2601, Australia

6. M. Cochinwala, S. Dalal, A. Elmagarmid, and V. Verykios, "Record Matching: Past, Present and Future", Available as Technical Report CSD-TR #01-013, Department of Computer Sciences, Purdue University, July 2001.

7. M. Elfeky, V. Verykios, and A. Elmagarmid, "TAILOR: A Record Linkage Toolbox", in Proceedings of the 18th International Conference on Data Engineering, San Jose, California, February 2002.

8. A.J Lait, and B. Randell, "An Assessment of Name Matching Algorithms", Technical Report, Department of Computing Science, University of Newcastle upon Tyne, UK 1993.

9. A. McCallum, K. Nigram and L. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching", in Proc. of the sixth ACM SIGKDD Int. Conf. on KDD, 2000.

10. M. A. Hernandez, S. J. Stolfo, "The Merge/Purge Problem for Large Databases", Department of Computer Science, Columbia University, New York, NY 10027

11. A. E. Monge and C.P. Elkan, "An efficient domain-independent algorithm for detecting approximately duplicate database records." In Proc. of the ACM-SIGMOD Workshop on Research Issues in on Knowledge Discovery and Data Mining, 1997

12. M. Neiling and R. M. Muller, "The good into the Pot, and the bad into the Crop." Preselection of Record Pairs for Database, Documents, and Information Fusion, Magdeburg, Germany, 2001.

13. Leicester E., "Gill. OX-LINK: The Oxford medical record linkage system." In Proceedings of the International Record Linkage Workshop and Exposition, 1997.

14. M. Elfeky, V. Verykios, "On Search Enhancement of the Record Linkage Process", in Proceedings of the KDD 2003 Workshop on Data Cleaning, Record Linkage and Object Consolidation, August 2003, Washington, DC, USA.

15. M. Elfeky, V. Verykios, and A. Elmagarmid, "Record Linkage: A Machine Learning Approach, A Toolbox, and A Digital Government Web Service", Department of Computer Sciences, Purdue University, Technical Report CSD-TR 03-024.

16. The Febrl Project Web site http://datamining.anu.edu.au/projects/linkage.html

17. I. Fellegi, and A. Sunter, "A Theory for Record Linkage", in Journal of the American Statistical Society, 1969.

18. The Python programming language Web site http://www.python.org

19. Mark A. Cameron, Kelly L. Taylor and Rohan Baxter, "Web Service Composition and Record Linking", in Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

20. M. Michelson and C. A. Knoblock, "Learning Blocking Schemes for Record Linkage", in Proceedings of AAAI-2006

21. Peter Christen and Tim Churches, "A Probabilistic Deduplication, Record Linkage and Geocoding System", Proceedings of the ARC Health Data Mining workshop, University of South Australia, April 2005

22. Peter Christen and Karl Goiser, "Assessing Deduplication and Data Linkage Quality: What to Measure?", in Proceedings of the 4th Australasian Data Mining Conference (AusDM 2005), Sydney, December 2005.

23. Hernanndez, M. A., and Stolfo, S. J., "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem". Data Mining and Knowledge Discovery 2(1):9-37, 1998

24. Peter Christen, "Probabilistic Data Generation for Deduplication and Data Linkage", Proceedings of the 6th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL '05), Brisbane, July 2005

25. V.S. Verykios, M.G. Elfeky, A.K. Elmagarmid, M. Cochinwala, and S. Dalal, "On the Accuracy and Completeness of the Record Matching Process", in Proc. Of the 2000 Conf. on Information Quality, Boston, Massachusetts, October 2000.

26. Peter Christen and Karl Goiser, "Quality and Complexity Measures for Data Linkage and Deduplication", Springler, March 2007

27. Winkler, W. E., "Matching and Record Linkage", in B. G. Cox et al. (ed.) Business Survey Methods, New York: J. Wiley, 355-384, 1995

28. Winkler, W. E., "Approximate String Comparator Search Strategies for Very Large Administrative Lists". Technical report, Statistical Research Report Series (Statistics 2005-02) U.S. Census Bureau, 2005

29. Winkler, W. E. (1999c), "Record Linkage Software and Methods for Merging Administrative Lists", Eurostat, Proceedings of the Exchange of Technology and Know-How '99

30. Tony Blakely and Clare Salmond, "Probabilistic record linkage and a method to calculate the positive predictive value", International Journal of Epidemiology 2002

31. Patrick Lehti and Peter Fankhauser, "A Precise Blocking Method for Record Linkage", DaWaK 2005, LNCS 3589, Springer-Verlag Berlin Heidelberg 2005

32. Peter Christen, Tim Churches and Markus Hegland, "Febrl – A Parallel Open Source Data Linkage System", PAKDD 2004, LNAI 3056, Springer-Verlag Berlin Heidelberg 2004

33. Karl Goiser, Peter Christen, "Towards Automated Record Linkage", Australasian Data Mining Conference (AusDM 2006), Sydney, December 2006.

34. Β. Μητρογιάννης, "Διατήρηση της εμπιστευτικότητας κατά την ενοποίηση των δεδομένων σε κατανεμημένες βάσεις δεδομένων", Σεπτέμβριος 2005.