ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ

Υλοποίηση Ευφυούς Πράκτορα για Ομαδοποίηση Εγγράφων που Ανακτώνται από το Διαδίκτυο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχάλης Κ. Νεοφύτου

Επιβλέπων : Ανδρέας-Γεώργιος Ν. Σταφυλοπάτης

Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2004

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ

Υλοποίηση Ευφυούς Πράκτορα για Ομαδοποίηση Εγγράφων που Ανακτώνται από το Διαδίκτυο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχάλης Κ. Νεοφύτου

Επιβλέπων : Ανδρέας-Γεώργιος Ν. Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Ιουνίου 2004.

……	……	……
Σταφυλοπάτης Ανδρέας-Γεώργιος	Κόλλιας Στέφανος	Τσανάκας Παναγιώτης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

……

Μιχάλης Κ. Νεοφύτου

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

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε στον Τομέα Τεχνολογίας Πληροφορικής & Υπολογιστών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Θέμα της διπλωματικής είναι οι τεχνικές ομαδοποίησης εγγράφων και η παράλληλη υλοποίηση ενός ευφυούς πράκτορα ο οποίος σε συνεργασία με μια μηχανή αναζήτησης στο Διαδίκτυο θα εφαρμόζει τις τεχνικές αυτές στα αποτελέσματα που ανακτώνται μετά από έρευνα του χρήστη.

Το μεγάλο πλήθος εγγράφων που σήμερα είναι διαθέσιμα στο Διαδίκτυο, σε τοπικά δίκτυα αλλά και σε ψηφιακές βιβλιοθήκες καθιστά πολύ χρήσιμη την ύπαρξη μιας μεθοδολογίας για την αποδοτική διαχείριση και ανάκτηση τους. Μια τέτοια προοπτική προσφέρουν οι διάφορες μέθοδοι ομαδοποίησης εγγράφων με βάση το θεματικό περιεχόμενο κάθε εγγράφου. Η ανάπτυξη τέτοιων τεχνικών ήταν ραγδαία τα τελευταία χρόνια και χρήση τους εκτείνεται σε ένα ευρύ φάσμα εφαρμογών. Ο πυρήνας αυτών των τεχνικών βασίζεται σε μεθόδους ανάκτησης πληροφορίας και μάθησης (επιβλεπόμενης ή μη) μηχανής, κάτι που σημαίνει ότι ένα σύστημα για να μπορεί να διεκπεραιώσει ομαδοποίηση εγγράφων σε ικανοποιητικά πλαίσια απόδοσης πρέπει προηγουμένως να έχει εκπαιδευτεί με χρήση κατάλληλων προτύπων εκπαίδευσης.

Εφαρμόζοντας μια τέτοια μέθοδο ομαδοποίησης κειμένων υλοποιήθηκε ένας ευφυής πράκτορας, ο SCAgent ο οποίος πραγματοποιεί αναζήτηση εγγράφων από το Διαδίκτυο με βάση τις σχετικές αιτήσεις του χρήστη. Ο πράκτορας λαμβάνει τα αποτελέσματα της αναζήτησης και αφού εκτελέσει ομαδοποίηση με βάση το θεματικό περιεχόμενο τους τα επιστρέφει στον χρήστη σε μια μορφή που να τον διευκολύνει στην πιο αποτελεσματική και γρήγορη αξιοποίηση τους.

Η παρούσα διπλωματική εργασία αποτελείται από τρία κεφάλαια και δύο παραρτήματα. Στο πρώτο κεφάλαιο περιγράφονται θέματα που σχετίζονται με το πρόβλημα της ομαδοποίησης κειμένων και αναπτύσσονται τεχνικές που αφορούν στην βελτιστοποίηση της όλης διαδικασίας. Στο δεύτερο κεφάλαιο περιγράφεται η υλοποίηση και η λειτουργία του ευφυούς πράκτορα που αναπτύχθηκε, του SCAgent. Στο τρίτο και τελευταίο κεφάλαιο περιγράφεται η πειραματική μεθοδολογία που εφαρμόστηκε με στόχο την αξιολόγηση της επίδοσης μιας συγκεκριμένης μεθόδου ομαδοποίησης σε συνδυασμό με τις διάφορες παραμέτρους που υπεισέρχονται σε αυτή. Τέλος, στο παράρτημα Α σημειώνεται η σχετική βιβλιογραφία από την οποία αντλήθηκαν χρήσιμες πληροφορίες, ενώ στο παράρτημα Β παρατίθεται μέρος του κώδικα της εφαρμογής SCAgent που αφορά στα σημαντικότερα σημεία υλοποίησης του.

Θα επιθυμούσα να εκφράσω τις θερμές μου ευχαριστίες στον υπεύθυνο για την εκπόνηση της εργασίας καθηγητή κ. Ανδρέα-Γεώργιο Σταφυλοπάτη για την πολύτιμη καθοδήγηση που προσέφερε καθώς και στον υποψήφιο διδάκτορα κ. Σπύρο Βρεττό για την σημαντική βοήθεια και συνεργασία για την όσο το δυνατό καλύτερη προσέγγιση του αντικειμένου που πραγματεύεται η παρούσα εργασία.

Μιχάλης Νεοφύτου
Αθήνα, Ιούνιος 2004

Λέξεις κλειδιά: ομαδοποίηση εγγράφων, μέθοδοι ομαδοποίησης, ανάκτηση πληροφορίας, πειραματική αξιολόγηση, επίδοση, υλοποίηση πράκτορα

Abstract

This thesis was developed at the Computer Science Division of the School of Electrical and Computer Engineering of the National Technical University of Athens.

The subject of the thesis is document clustering methods alongside with the implementation of an intelligent agent for the application of these methods over results acquired post to a user defined Web query.

The vast array of documents available today at the Web, local networks and digital libraries renders very useful the existence of a certain methodology for efficient management and retrieval. Various document clustering methods based on the contents of each document offer this perspective. The development of these methods has been rapid over the past years and their enforcement extends to a wide range of applications. Information retrieval techniques and machine learning (supervised or not) constitute the core of these methods, which means that a system in order to transact document clustering effectively has to be trained using the appropriate training data.

SCAgent, an intelligent agent application, has been implemented by applying one of these clustering methods to Web documents retrieved after user defined queries. The agent receives query results and after performing content-based clustering, presents the user with a view that facilitates the efficient and fast utilization.

The thesis is composed by three chapters and two appendixes. In chapter one, topics relevant to document clustering and process optimization are discussed. In chapter two, the implementation details and functionality of the intelligent agent developed are presented. Chapter three describes the experimental methodology that was applied with the scope of evaluating the performance of a certain clustering method in conjunction with the various parameters used. Finally, appendix A presents the relevant bibliography from which useful information was drawn while appendix B lists part of the source code of the SCAgent application pertaining to the most important implementation details.

I wish to express my warmest thanks to professor Andreas Stafylopatis who held the responsibility of supervising the present thesis for his valuable guidance and to PhD student Spyros Vrettos for the important support and cooperation during the approach of the subject presented in this thesis.

Michael Neophytou
Athens, June 2004

Keywords: document clustering, clustering methods, information retrieval, experimental evaluation, performance, agent implementation

Πίνακας περιεχομένων

Ευρετήριο σχημάτων και πινάκων

Σχήματα

Πίνακες

1

Ομαδοποίηση εγγράφων

1.1 Εισαγωγή

Κατά τα πρόσφατα χρόνια έχει παρατηρηθεί μια εξαιρετικά αυξανόμενη τάση διάθεσης μεγάλου όγκου εγγράφων στο Διαδίκτυο, σε ηλεκτρονικές βιβλιοθήκες αλλά και σε μεγάλα τοπικά δίκτυα. Το γεγονός αυτό έχει οδηγήσει στην ανάπτυξη αρκετών μεθόδων οι οποίες να βοηθούν τον ενδιαφερόμενο στην γρήγορη πλοήγηση, αποδοτική σύνοψη και αποτελεσματική οργάνωση αυτών των διαθέσιμων πόρων με απώτερο στόχο την ευκολότερη ανεύρεση της πληροφορίας την οποία αναζητά. Οι ταχείς και ποιοτικοί αλγόριθμοι ομαδοποίησης εγγράφων διαδραματίζουν ένα σημαντικό ρόλο στην επίτευξη αυτού του στόχου καθώς έχει αποδειχτεί ότι παρέχουν ένα εύχρηστο μηχανισμό πλοήγησης οργανώνοντας μεγάλο όγκο πληροφορίας σε θεματικές ομάδες σαφώς μικρότερου πλήθους. Αυτή η συνεχώς αυξανόμενη σημασία της ομαδοποίησης εγγράφων και το επεκταμένο φάσμα των εφαρμογών της, οδήγησε στην ανάπτυξη ενός σημαντικού αριθμού καινοφανών αλγόριθμων με ποικίλες αντισταθμίσεις ανάμεσα στην πολυπλοκότητα και την ποιότητα. Ανάμεσα σε αυτούς τους αλγόριθμους, μια κλάση αλγόριθμων ομαδοποίησης οι οποίοι έχουν σχετικά χαμηλές υπολογιστικές απαιτήσεις είναι εκείνοι αυτοί που αντιμετωπίζουν το πρόβλημα της ομαδοποίησης ως μια διαδικασία βελτιστοποίησης που στοχεύει στην μεγιστοποίηση ή ελαχιστοποίηση ενός συγκεκριμένου κριτηρίου που ορίζεται επί της συνολικής λύσης ομαδοποίησης.

1.2 Ανάκτηση πληροφορίας

Η τεράστια αύξηση κατά τα πρόσφατα χρόνια της ποσότητας της διαθέσιμης πληροφορίας με ηλεκτρονικά μέσα σίγουρα παρέχει αυξημένες δυνατότητες και πληθώρα χρήσιμων πόρων στους σημερινούς χρήστες. Εντούτοις, ένας χρήστης ο οποίος αντιμετωπίζει την ανάγκη ανάκτησης και χρήσης πληροφορίας οποιουδήποτε είδους συχνά συνθλίβεται από την αυξημένη ποσότητα πληροφορίας και δυσκολεύεται αρκετά στην διάκριση της εκείνης η οποία είναι πραγματικά χρήσιμη για την διεκπεραίωση των στόχων του. Αδιαμφισβήτητα η πλέον επιτυχής προσέγγιση όσον αφορά στην οργάνωση αυτής της μάζας πληροφορίας ούτως ώστε ο χρήστης να έχει καλύτερη αντίληψη της είναι η δομημένη παροχή της πληροφορίας με χρήση συγκεκριμένων τεχνικών.

Η γνωστή εγκυκλοπαίδεια Brittanica αποτελεί ένα παράδειγμα επιτυχούς οργάνωσης της πληροφορίας η οποία παρέχει τόσο εξερευνητικές δυνατότητες από απόψεως μάθησης όσο

δυνατότητες έρευνας για συγκεκριμένες αναφορές. Το τμήμα προ-παίδειας (*Propaedia*) περιέχει μια ιεραρχική, δομημένη σκιαγράφηση διαφόρων γνώσεων και ένα οδηγό χρήσης της εγκυκλοπαίδειας. Το τμήμα μικρο-παιδείας (*Micropaedia*) προσφέρει ένα κατάλογο άρθρων τα οποία καλύπτουν θέματα που αφορούν στην ανθρώπινη γνώση και είναι ταξινομημένα αλφαβητικά με βάση την λέξη-κλειδί η οποία αντιπροσωπεύει το θέμα. Αντικείμενα για τα οποία απαιτείται αναλυτικότερη κάλυψη παρατίθενται στο τμήμα μακρο-παιδείας (*Macropaedia*) ενώ το ευρετήριο υποστηρίζει την αναζήτηση θεμάτων με βάση μια λέξη-κλειδί [Mur02].

Αυτή η οργάνωση πληροφορίας, αποδεδειγμένα αποτελεσματική στο πέρασμα των χρόνων, μπορεί να εξομοιωθεί από τα συστήματα ανάκτησης πληροφορίας. Μια ιεραρχική οργάνωση των θεμάτων μπορεί να υποστηρίζει είτε την διαδικασία μάθησης για κάποιον χρήστη που δεν είναι εξοικειωμένος με το αντικείμενο είτε την καλύτερη πλοήγηση για κάποιον χρήστη που γνωρίζει την σημασιολογική δομή ενός συγκεκριμένου πεδίου. Για την ολοκλήρωση της γενικότερης εικόνας, οι σύνδεσμοι αναφοράς (*referential links*) παρέχουν συνδέσεις σε εναλλακτικές περιοχές κάλυψης του αντικειμένου και οι μηχανές αναζήτησης διαδραματίζουν τον ρόλο του ευρετηρίου. Ενώ οι σύνδεσμοι αναφοράς έχουν αποδειχτεί χρήσιμοι σε εφαρμογές πολυμέσων και ιδιαίτερα στο Διαδίκτυο, το ενδιαφέρον επικεντρώνεται στις μεθόδους ιεραρχικής οργάνωσης πληροφορίας για την υποστήριξη της διαδικασίας μάθησης, ανάκτησης ή πλοήγησης.

Παραδοσιακά η δομή του χώρου της πληροφορίας χρησιμοποιούταν για την βελτίωση της αποτελεσματικότητας των αλγόριθμων ανάκτησης ή για την αυτόματη επέκταση ερωτημάτων αναζήτησης. Η πρόσφατη εμφάνιση διαδραστικών περιβαλλόντων αναζήτησης πληροφορίας έχει αυξήσει την σημασία μεθόδων δόμησης και κατηγοριοποίησης.

Τόσο χειροκίνητες όσο και αυτόματες μέθοδοι έχουν προταθεί και διερευνηθεί για την κατηγοριοποίηση-ομαδοποίηση συλλογών εγγράφων με την προοπτική υποστήριξης συστημάτων ανάκτησης πληροφορίας. Η χειροκίνητη κατηγοριοποίηση τυπικά βελτιστοποιείται για ένα εξειδικευμένο πεδίο με χρήση της γνώσης ειδικών στο συγκεκριμένο αντικείμενο και είναι περισσότερο πιθανό να αποκαλύπτει την σημασιολογική δομή του πεδίου και των υποπεριοχών του. Εναλλακτικά, η ομαδοποίηση (*clustering*) διαθέτει το πλεονέκτημα του αυτοματισμού και κατά συνέπεια είναι μια διαδικασία γρηγορότερη. Είναι ανεξάρτητη του θεματικού πεδίου, οδηγούμενη από δεδομένα (βασίζεται στο περιεχόμενο των εγγράφων παρά στην γνώση ειδικών σχετικά με το πεδίο) και συνήθως επιτυχής στην αναγνώριση θεμάτων με σημασία σε σχετικά ετερογενείς συλλογές [Mur02].

Η παρατήρηση της συμπεριφοράς ενός ερευνητή σε μια βιβλιοθήκη αποκαλύπτει χωρίς αμφιβολία την σημασία της ύπαρξης δομής κατά την μάθηση, την εξερεύνηση ενός πεδίου ή την διενέργεια κάποιας έρευνας. Αν μια εξειδικευμένη συλλογή εμφανίζει κάποια ταξινόμηση ή έχει

ταξινομηθεί χειρονακτικά, τότε αυτή η δομή μπορεί να χρησιμοποιηθεί για την αποτελεσματική υποστήριξη της πλοήγησης ανάμεσα στο περιεχόμενο της συλλογής. Παρόλα αυτά, δεν εμφανίζουν όλα τα θεματικά πεδία κάποιου είδους ταξινόμηση ή δεν σχετίζονται με κάποια ταξινομημένη συλλογή. Τα περιεχόμενα τέτοιων συλλογών μπορούν να ομαδοποιηθούν και το λαμβανόμενο αποτέλεσμα μπορεί να χρησιμοποιηθεί για την καθοδήγηση της έρευνας.

Συνεπώς, η διαδικασία της ομαδοποίησης εγγράφων εξετάζεται ως ένα εργαλείο κατασκευής μιας δομής της συλλογής η οποία να παρέχει αυξημένες δυνατότητες στον ερευνητή.

1.3 Ομαδοποίηση και η υπόθεση ομαδοποίησης

Η ανάλυση ομάδων ή διαδικασία ομαδοποίησης (*clustering process*) είναι μια τεχνική πολυμεταβλητής ανάλυσης η οποία αναθέτει αντικείμενα σε αυτόματα δημιουργούμενες ομάδες βασισμένη στον υπολογισμό του βαθμού συσχέτισης ή ομοιότητας μεταξύ των αντικειμένων και των ομάδων [Mur02]. Η ομαδοποίηση εμφανίζεται σε μια πληθώρα εφαρμογών σχετικές με την ανάκτηση πληροφορίας όπως την συμπαράταξη όρων (*term clustering*) βασισμένη στις ιδιωματικές εκφράσεις τους ή την συμπαράταξη πηγών πληροφορίας (*grouping of information sources*). Η πιο κοινή της εφαρμογή όμως εντοπίζεται στην ομαδοποίηση εγγράφων-κειμένων.

Η εισαγωγή της έννοιας της ομαδοποίησης εγγράφων στο πεδίο της ανάκτησης πληροφορίας έγινε με στόχο την αύξηση της αποδοτικότητας της ανάκτησης. Μετά από κάποια αρχική επιβάρυνση οφειλόμενη στην διαδικασία οργάνωσης των εγγράφων σε ομάδες, η έρευνα μπορεί να εξελιχθεί σε αναζήτηση των ομάδων που ανταποκρίνονται καλύτερα στους όρους ενός ερωτήματος σε αντίθεση με την εξαντλητική εξέταση όλων των επιμέρους εγγράφων.

Η υπόθεση ομαδοποίησης (*cluster hypothesis*) που προτάθηκε από τους Jardine και van Rijsbergen βασίστηκε στην πρόταση ότι οι συσχετίσεις μεταξύ εγγράφων παρέχουν πληροφορία για την σχετικότητα των εγγράφων ως ένα συγκεκριμένο αίτημα. Οι πειραματικές εργασίες τους έδειξαν ακριβώς ότι «στενά συνδεδεμένα έγγραφα τείνουν να ανήκουν στην ίδια ομάδα και να είναι σχετικά ως προς την ίδια αίτηση» [JR71].

1.4 Τεχνικές μάθησης

Η ομαδοποίηση είναι μια διαδικασία η οποία μπορεί να εκτελεστεί αποδοτικότερα με την ενσωμάτωση και χρησιμοποίηση πρότερης ή εξωτερικής γνώσης. Η γνώση που ενσωματώνεται αντικατοπτρίζει τους συγκεκριμένους στόχους και την συλλογιστική του χρήστη και σαφώς συνεισφέρει στην αποκόμιση καλύτερων αποτελεσμάτων. Σε μια τέτοια περίπτωση η οργάνωση της πληροφορίας γίνεται κατά τρόπο που αντικατοπτρίζει πιο στενά τα ενδιαφέροντα του χρήστη

σε σχέση με τις σταθερές προσεγγίσεις που παρέχουν οι συμβατικές μέθοδοι ομαδοποίησης. Ιδιαίτερα, μια επιβλεπόμενη τεχνική ομαδοποίησης είναι απαραίτητη όταν οι ομάδες που προκύπτουν με εφαρμογή μη επιβλεπόμενων τεχνικών δεν είναι συμπαγείς και καλά διαχωρισμένες.

Αρκετοί ερευνητές έχουν ασχοληθεί με το πρόβλημα της ενσωμάτωσης εξωτερικής γνώσης σε ένα σύστημα ομαδοποίησης. Μια σχετική πρόταση αποτελεί η «ομαδοποίηση προσανατολισμένη στον χρήστη» (*user-oriented clustering*) κατά την οποία η ταυτοποίηση των ομάδων γίνεται στην βάση της ανάδρασης σχετικότητας (*relevance feedback*) χωρίς την εξάρτηση από κάποιο όρο-δείκτη. Αυτή η προσέγγιση απαιτεί όπως το σύστημα να έχει συσσωρεύσει μια μακροπρόθεσμη πληροφορία αποτελεσμάτων από προηγούμενες έρευνες. Για την αναγνώριση των ομάδων απαιτείται η επίλυση του προβλήματος επιλογής συνόρων ενώ επιπλέον ενδέχεται να προκύψουν δυσκολίες στις περιπτώσεις αποτελεσμάτων που επιστρέφονται από διαφορετικές έρευνες.

Μια άλλη προτεινόμενη μέθοδος κατά την οποία παρέχεται εξωτερική γνώση με την εφαρμογή λογικών κανόνων ταξινόμησης, δεν κρίνεται ιδιαίτερα ικανοποιητική για κείμενα καθώς το μεγάλο πλήθος όρων καθιστά απαγορευτική την έκφραση αυτής της γνώσης από μέρους του χρήστη σε βαθμό ικανό να βελτιώσει την διαδικασία ομαδοποίησης.

Η προσέγγιση της ημι-επιβλεπόμενης μάθησης αποτελεί ίσως την καλύτερη τομή μεταξύ των επιβλεπόμενων και μη επιβλεπόμενων τεχνικών. Ο ημι-επιβλεπόμενος χαρακτήρας εκφράζεται κυρίως με την αξιοποίηση της διαθέσιμης γνώσης για την καλύτερη αρχικοποίηση του αλγόριθμου ομαδοποίησης και την εφαρμογή μερικών απλών περιορισμών (*constraints*) οι οποίοι πρέπει να ικανοποιούνται από την κατανομή των προτύπων στις ομάδες. Ιδιαίτερα, οι κατάλληλες αρχικοποιήσεις βασισμένες σε εξωτερική γνώση κατευθύνουν την αναζήτηση λύσης στο πρόβλημα της ομαδοποίησης σε καλύτερες περιοχές του χώρου αναζήτησης και συνεπώς αποτρέπουν τον εγκλωβισμό σε τοπικές λύσεις ενώ παράλληλα παράγουν ομαδοποιήσεις που συνάδουν με τις οριζόμενες από τον χρήστη προδιαγραφές.

1.5 Μέθοδοι ομαδοποίησης

Το πρόβλημα της ομαδοποίησης έχει μελετηθεί σε μεγάλο βαθμό από αρκετές επιστημονικές οπτικές γωνίες και διάφοροι σχετικοί αλγόριθμοι έχουν αναπτυχθεί κατά καιρούς. Αυτοί οι αλγόριθμοι μπορούν να κατηγοριοποιηθούν βασιζόμενοι σε κριτήρια που αφορούν είτε στην θεμελιώδη μεθοδολογία που εφαρμόζουν οδηγώντας σε συσσωρευτικές (*agglomerative*) ή κατατμητικές (*partitional*) προσεγγίσεις είτε στην δομή της τελικής λύσης την οποία παράγουν καταλήγοντας σε ιεραρχικές και μη ιεραρχικές προσεγγίσεις.

Οι ιεραρχικοί αλγόριθμοι παράγουν μια λύση ομαδοποίησης η οποία παρουσιάζει μια δενδρική δομή με μια μοναδική ομάδα ως υπερσύνολο στην κορυφή και μονοσύνολες ομάδες στα άκρα. Ειδικότερα, οι συσσωρευτικοί αλγόριθμοι προσδιορίζουν τις ζητούμενες ομάδες αναθέτοντας αρχικά κάθε αντικείμενο σε μια ξεχωριστή ομάδα και ακολούθως εκτελώντας επαναληπτικές συνενώσεις ομάδων μέχρι την ικανοποίηση κάποιου κριτηρίου τερματισμού. Έχουν προταθεί αρκετές μέθοδοι για τον προσδιορισμό των ομάδων που πρέπει να συνενώνονται σε κάθε επανάληψη όπως οι μέσοι ομάδων (*group average*), απλής σύνδεσης (*single-link*), πλήρους σύνδεσης (*complete link*), CURE, ROCK, CHAMELEON.

Οι κατατμητικοί (*partitional*) αλγόριθμοι όπως ο K-Means, K-medoids ή Autoclass προσδιορίζουν τις ομάδες με κατάτμηση ολόκληρου του συνόλου των προτύπων σε ένα προκαθορισμένο ή αυτόματα παραγόμενο πλήθος ομάδων. Σε εξάρτηση με τον συγκεκριμένο αλγόριθμο που χρησιμοποιείται, ένα σχήμα ομαδοποίησης μπορεί να ληφθεί άμεσα ή έμμεσα με επαναλαμβανόμενες διχοτομήσεις. Στην πρώτη περίπτωση δεν υπάρχει εν γένει κάποια σχέση μεταξύ των σχημάτων ομαδοποίησης που λαμβάνονται στα διαφορετικά επίπεδα ανάλυσης (*granularity levels*) ενώ στην δεύτερη δημιουργούνται λύσεις ιεραρχικής δομής.

Οι μη ιεραρχικές μέθοδοι διαιρούν μια συλλογή N εγγράφων σε M ομάδες και χρησιμοποιούν ευριστικές μεθόδους για την ανάθεση των εγγράφων ούτως ώστε να επιτύχουν καλές υπολογιστικές επιδόσεις. Με αναδρομική χρήση μπορούν επίσης να χρησιμοποιηθούν για την παραγωγή ιεραρχικών δομών.

Στις περισσότερες περιπτώσεις η ομαδοποίηση που λαμβάνεται εξαρτάται από την σειρά επεξεργασίας των εγγράφων και από τις ευριστικές παραμέτρους που χρησιμοποιούνται. Σε πειράματα ανάκτησης και αξιολόγησης πληροφορίας βασισμένης σε ομάδες εντοπίστηκε μια σχετική μείωση της επίδοσης και για τον λόγο αυτό οι μη ιεραρχικές τεχνικές κρίθηκαν ακατάλληλες για χρήση στο γενικότερο πλαίσιο της ανάκτησης πληροφορίας.

Εντούτοις, η πρόσφατη αναγνώριση της χρησιμότητας τεχνικών οπτικοποίησης και εργαλείων πλοήγησης ιδιαίτερα στα πλαίσια της ανάκτησης πληροφορίας από το Διαδίκτυο προσέδωσε εκ νέου σημασία στις μη ιεραρχικές μεθόδους ομαδοποίησης λόγω της αποδοτικότητας τους από άποψης ταχύτητας οργάνωσης των αποτελεσμάτων έρευνας. Παρόλο που θεωρούνται λιγότερο αποδοτικές μέθοδοι σε σχέση με τις αντίστοιχες ιεραρχικές, έχουν αποδειχτεί επαρκείς για χρήση από διάφορα εργαλεία ανάκτησης και απεικόνισης. Τέτοια εργαλεία μπορούν να υποστηρίξουν τον χρήστη στην εξερεύνηση των αποτελεσμάτων της έρευνας με την αναγνώριση της δομής του ανακτώμενου συνόλου εγγράφων.

Στα πρόσφατα χρόνια, αρκετοί ερευνητές έχουν αναγνωρίσει το γεγονός ότι οι κατατμητικοί αλγόριθμοι είναι κατάλληλοι στις περιπτώσεις ομαδοποίησης μεγάλων συνόλων δεδομένων λόγω των σχετικά χαμηλών υπολογιστικών απαιτήσεων τους. Ένα βασικό

χαρακτηριστικό αρκετών αλγόριθμων ομαδοποίησης είναι ότι χρησιμοποιούν μια συνολική συνάρτηση κριτηρίου, η βελτιστοποίηση της οποίας καθοδηγεί την όλη διαδικασία ομαδοποίησης. Οι αλγόριθμοι για τους οποίους η συνάρτηση κριτηρίου είναι σαφής και διατυπώνεται άμεσα, μπορεί να θεωρηθεί ότι αποτελούνται από δύο (ανεξάρτητα μεταξύ τους) συστατικά μέρη. Το πρώτο είναι η συνάρτηση κριτηρίου η οποία πρέπει να βελτιστοποιείται από το σχήμα ομαδοποίησης και το δεύτερο ο αλγόριθμος που επιτυγχάνει αυτή την βελτιστοποίηση.

1.6 Αλγόριθμοι ομαδοποίησης

Η διαφορά ανάμεσα στις μεθόδους ομαδοποίησης και στους αλγόριθμους ομαδοποίησης είναι αρκετά λεπτή και πολλές φορές μη διακρίσιμη. Κατά γενική αποδοχή οι μέθοδοι περιγράφουν αλγοριθμικά βήματα κατά γενικό τρόπο και χωρίς αναφορές σε λεπτομέρειες υλοποίησης όπως για παράδειγμα τις δομές δεδομένων που χρησιμοποιούνται στην πράξη. Κάθε μέθοδος μπορεί να υλοποιηθεί με μια ποικιλία αλγόριθμων οι οποίοι διαφέρουν στις εξής απόψεις:

▪ την δόμηση των δεδομένων - μια εξειδικευμένη δομή δεδομένων μπορεί να βελτιώσει την ταχύτητα πρόσβασης αλλά και να αυξήσει την πολυπλοκότητα του κώδικα μειώνοντας έτσι την ευελιξία και προσαρμοστικότητα του αλγόριθμου σε διαφορετικές συλλογές δεδομένων

▪ την χρήση της μνήμης - αν τα δεδομένα και τα ενδιάμεσα αποτελέσματα αποθηκεύονται στη μνήμη τότε μπορεί να επιτευχθεί χαμηλότερη υπολογιστική πολυπλοκότητα και ψηλότερες ταχύτητες εκτέλεσης. Το μειονέκτημα είναι η απαίτηση για υπολογιστικά συστήματα που διαθέτουν αρκετή ποσότητα μνήμης ή εναλλακτικά ο περιορισμός του μεγέθους της συλλογής που πρόκειται να ομαδοποιηθεί

▪ την χρήση των αποθηκευτικών μέσων - η χρήση των αποθηκευτικών μέσω συνήθως εξισορροπείται με την χρήση της μνήμης. Εντούτοις, για πολύ μεγάλες συλλογές είναι ανέφικτη η αποθήκευση των δεδομένων αποκλειστικά στην μνήμη και συνεπώς η γρήγορη πρόσβαση σε μέσα μόνιμης αποθήκευσης είναι ουσιαστικής σημασίας. Στην περίπτωση αυτή η χρήση των κατάλληλων δομών αποθήκευσης και η συμπίεση μπορούν να αποδειχτούν εξαιρετικά χρήσιμες

Ακόμα και αν διαφέρουν σε πολυπλοκότητες χρόνου και αποθήκευσης, οι αλγόριθμοι που υλοποιούν την ίδια ακριβώς μέθοδο ομαδοποίησης αναμένεται ότι θα παράγουν το ίδιο αποτέλεσμα με δεδομένη είσοδο. Αυτό δεν συμβαίνει πάντοτε καθώς οι αλγόριθμοι είναι δυνατό να χρησιμοποιούν διαφορετικές ρυθμιστικές παραμέτρους για την αύξηση της επίδοσης τους και ενδέχεται το αποτέλεσμα να τροποποιούνται έστω και σε μικρό βαθμό.

1.7 Αναπαράσταση εγγράφων

Τα αντικείμενα μιας συλλογής στην οποία πρόκειται να εφαρμοστεί η διαδικασία της ομαδοποίησης είναι αναγκαίο να περιγραφούν με γνωρίσματα ή χαρακτηριστικά κατά τέτοιο τρόπο ώστε να μπορεί να υπολογιστεί ο βαθμός ομοιότητας μεταξύ τους. Στην περίπτωση των εγγράφων-κειμένων συνήθως οι λέξεις-κλειδιά αποτελούν τα γνωρίσματα. Συνεπώς, η ομαδοποίηση στην πράξη εφαρμόζεται σε αντιπροσώπους των εγγράφων οι οποίοι λαμβάνονται με μεθόδους που περιγράφονται στη συνέχεια.

Άλλα χαρακτηριστικά όπως όμοια διαπιστευτήρια (*credentials*) ή συνύπαρξη των ίδιων συγγραφέων μπορούν να χρησιμοποιηθούν για τον υπολογισμό της συσχέτισης εγγράφων. Ηλεκτρονικά έγγραφα με ειδικό περιεχόμενο όπως εικόνες, βίντεο και ήχους μπορούν να αναπαρασταθούν με μια ποικιλία ιδιαίτερων χαρακτηριστικών του κάθε είδους. Γενικά, η ομαδοποίηση εφαρμόζεται με τον ίδιο τρόπο όπως και για τα απλά κείμενα αρκεί να είναι δυνατή η παραγωγή περιγραφών-αντιπροσώπων του εγγράφου και ο υπολογισμός κάποιου μέτρου ομοιότητας επί αυτών.

Οι αλγόριθμοι ομαδοποίησης συνήθως χρησιμοποιούν το μοντέλο διανυσματικού χώρου για την αναπαράσταση κάθε εγγράφου [KZ02]. Σύμφωνα με αυτό το μοντέλο, κάθε έγγραφο d θεωρείται ως ένα διάνυσμα ορισμένο στο πεδίο των όρων (λέξεων). Στην απλούστερη μορφή, το έγγραφο αναπαρίσταται ως διάνυσμα συχνοτήτων όρων (*term frequency vector – TF vector)*

$$\vec{d}_{tf} = \{tf_1, tf_2, ..., tf_m\}$$

όπου tf_i είναι η συχνότητα του όρου i που εμφανίζεται στο έγγραφο. Μια τροποποίηση του μοντέλου αυτού που συναντάται συχνά είναι η στάθμιση της συχνότητας κάθε όρου του εγγράφου με βάση την αντίστροφη συχνότητα κειμένου (*inverse document frequency, IDF*) επί του συνόλου των εγγράφων. Το κίνητρο για την συγκεκριμένη τροποποίηση εντοπίζεται στην διαπίστωση ότι οι όροι που εντοπίζονται συχνά σε μεγάλο αριθμό εγγράφων διαθέτουν μειωμένη ικανότητα διάκρισης των εγγράφων μεταξύ τους και συνεπώς είναι αναγκαίο να υποβαθμιστούν. Η στάθμιση πραγματοποιείται πολλαπλασιάζοντας την συχνότητα κάθε όρου i με την ποσότητα $log(N / df_i)$ όπου N είναι το πλήθος των δεδομένων εγγράφων και df_i το πλήθος των εγγράφων στα οποία εμφανίζεται ο συγκεκριμένος όρος. Αυτή η τροποποίηση οδηγεί στην αναπαράσταση συχνότητα όρου-αντίστροφη συχνότητα κειμένων (*term frequency-inverse document frequency, TF-IDF*)

$$\vec{d}_{tfidf} = \{tf_1 \cdot log(N / df_1), tf_2 \cdot log(N / df_2), ..., tf_m \cdot log(N / df_m)\}$$

Στην γενικότερη περίπτωση, τα έγγραφα εμφανίζουν διαφορετικά μήκη καθώς είναι δυνατό να διαφέρει το πλήθος των όρων που εντοπίζονται σε κάθε έγγραφο. Για να

αντισταθμιστεί η ιδιομορφία που δημιουργεί αυτός ο παράγοντας τα διανύσματα αναπαράστασης κάθε εγγράφου κανονικοποιούνται σε μοναδιαίο μήκος ($\left\|\vec{d}_{tfidf}\right\|_2 = 1$) και πλέον τα έγγραφα αναπαρίστανται ως διανύσματα στο σύστημα αναφοράς μιας μοναδιαίας υπερσφαίρας.

1.8 Στάθμιση χαρακτηριστικών

Διάφοροι μέθοδοι στάθμισης έχουν χρησιμοποιηθεί κατά καιρούς για τον υπολογισμό της συνεισφοράς των δεικτοδοτημένων όρων σε κάθε έγγραφο ενός συνόλου προς ομαδοποίηση.

Η επίδραση των μεθόδων στάθμισης στην ομαδοποίηση εγγράφων δεν είναι ιδιαίτερα σαφής. Διαισθητικά, η στάθμιση επηρεάζει την αναπαράσταση των εγγράφων και εμμέσως την ομοιότητα μεταξύ τους κατά τρόπο ώστε «καλές» μέθοδοι στάθμισης αναμένεται ότι θα αναγνωρίζουν με περισσότερη ακρίβεια «όμοια» έγγραφα και θα βελτιώνουν την επίδοση της διαδικασίας ομαδοποίησης.

Ανάμεσα στα πειραματικά αποτελέσματα που έχουν αναφερθεί για το συγκεκριμένο θέμα, ο Willett υποδεικνύει ότι η χρήση μεθόδων στάθμισης δεν οδηγεί σε σταθερές βελτιώσεις της επίδοσης σε σχέση με την χρήση μη σταθμισμένων όρων [Wil83]. Εντούτοις, η ισχύς του συμπεράσματος ότι οι εξεζητημένες μέθοδοι στάθμισης δεν συνεισφέρουν σημαντικά περιορίζεται από την επιλογή των μέτρων ομοιότητας που δοκιμάστηκαν, το είδος των συλλογών εγγράφων που χρησιμοποιήθηκαν στα πειράματα και από τα μέτρα επίδοσης που υιοθετήθηκαν. Επιπλέον, αυτά τα συμπεράσματα αμφισβητούνται από την υπόθεση του Dubin ότι η ανάθεση εγγράφων σε ομάδες εξαρτάται περισσότερο από την εν γένει διακριτική ικανότητα των όρων που απαρτίζουν τους αντιπροσώπους κάθε εγγράφου [Dub96].

Εκτός από την συζήτηση σχετικά με την επίδραση μεθόδων εσωτερικής στάθμισης (*internal weighting schemes*) βασισμένων στις συχνότητες των όρων στα έγγραφα, προτείνεται επίσης η εφαρμογή μεθόδων εξωτερικής στάθμισης οι οποίες μπορούν να οδηγήσουν σε ωφέλιμα αποτελέσματα. Αν δεχτούμε την υπόθεση ότι κάποιες συλλογές εγγράφων μπορεί να παρουσιάζουν μια φυσική δομή με βάση την οποία η θέση κάθε εγγράφου είναι σαφής, τότε η διαδικασία της ομαδοποίησης μάλλον θα έπρεπε να ανακαλύπτει αυτή την εγγενή δομή παρά να επιβάλλει κάποια άλλη. Όμως, για συλλογές που καλύπτουν μια ποικιλία πολύπλοκων και πολύπλευρων θεμάτων ο χρήστης ενδέχεται να ενδιαφέρεται για κάποια συγκεκριμένη όψη ή τοπική προβολή της συλλογής. Συνεπώς, ένα σχήμα εξωτερικής στάθμισης (για παράδειγμα εξαγόμενο από συχνά υποβαλλόμενες ερωτήσεις) μπορεί να ωθήσει προς την κατεύθυνση που επιθυμεί ο χρήστης.

1.9 Συντελεστές ομοιότητας

Οι συντελεστές ομοιότητας είναι συναρτήσεις οι οποίες αντιστοιχίζουν μια πραγματική τιμή σε ένα ζεύγος αντικειμένων μιας συλλογής βασιζόμενοι πάνω στα χαρακτηριστικά που περιγράφουν τα αντικείμενα και καταδεικνύουν με αυτό τον τρόπο τον βαθμό ομοιότητας ή ανομοιότητας μεταξύ τους.

Στο παρελθόν έχουν προταθεί δύο επικρατείς μέθοδοι για τον υπολογισμό του βαθμού ομοιότητας μεταξύ δύο εγγράφων d_i και d_j. Η πρώτη μέθοδος βασίζεται στη ευρέως χρησιμοποιούμενη συνάρτηση συνημίτονου (*cosine function*)

$$cos\left(\vec{d}_i, \vec{d}_j\right) = \frac{\vec{d}_i^t \cdot \vec{d}_j}{\left\|\vec{d}_i\right\| \cdot \left\|\vec{d}_j\right\|}$$

και με δεδομένο ότι τα d_i και d_j είναι κανονικοποιημένα με μέτρο την μονάδα η πιο πάνω σχέση απλοποιείται στην $cos\left(\vec{d}_i, \vec{d}_j\right) = \vec{d}_i^t \cdot \vec{d}_j$ [KZ02]. Η συνάρτηση αυτή μηδενίζεται όταν τα διανύσματα είναι τελείως ανόμοια (ορθογώνια) μεταξύ τους και παίρνει την μέγιστη τιμή της μονάδας όταν τα διανύσματα είναι εντελώς όμοια. Ο συντελεστής συνημίτονου (ή εσωτερικού γινομένου) είναι ιδιαίτερα χρήσιμος στην διαδικασία της ομαδοποίησης λόγω της διαισθητικής ερμηνείας αλλά και λόγω της αποδεδειγμένης αποτελεσματικότητας του. Η διεύθυνση του διανύσματος μπορεί να θεωρηθεί ως ένδειξη για το θεματικό περιεχόμενο του εγγράφου το οποίο αναπαριστά και συνεπώς μια μεγάλη τιμή της συνάρτησης ομοιότητας συνημίτονου φανερώνει έγγραφα τα οποία αναμένεται να αναφέρονται στο ίδιο (ή παραπλήσιο) θέμα.

Είναι δυνατή και μερικές φορές ευκολότερη ή διαισθητικά καλύτερη η χρήση συντελεστών απόστασης που φανερώνουν ανομοιότητα αντί των συντελεστών ομοιότητας. Το πλεονέκτημα τους είναι η απλή γεωμετρική ερμηνεία που είναι βολική για γραφικές αναπαραστάσεις και για τον λόγο αυτό είναι ιδιαίτερα δημοφιλής η χρήση τους σε εργαλεία οπτικής αναπαράστασης. Το μειονέκτημα αυτών των συντελεστών εντοπίζεται στην ανακριβή αναπαράσταση της πραγματικής σχέσης μεταξύ δύο εγγράφων που μπορεί να οδηγήσει στην θεώρηση των εγγράφων ως όμοια ακόμα και αν δεν εμφανίζουν κοινούς όρους.

Έτσι, μια δεύτερη μέθοδος εκτιμά την ομοιότητα (ή καλύτερα την ανομοιότητα) των εγγράφων υπολογίζοντας την ευκλείδεια απόσταση μεταξύ των αντίστοιχων διανυσμάτων

$$dis\left(\vec{d}_i, \vec{d}_j\right) = \sqrt{\sum_{k=1}^{m}\left(d_k^i - d_k^j\right)^2} = \left\|\vec{d}_i - \vec{d}_j\right\|_2$$

Στην περίπτωση εντελώς όμοιων εγγράφων, η απόσταση των αντίστοιχων διανυσμάτων είναι μηδενική ενώ για εντελώς ανόμοια έγγραφα η απόσταση των αντίστοιχων διανυσμάτων προκύπτει ίση με $\sqrt{2}$ [KZ02].

1.10 Συναρτήσεις κριτηρίου

Σε ένα κάπως αφαιρετικό επίπεδο το πρόβλημα της ομαδοποίησης εγγράφων διατυπώνεται ως ακολούθως: Με δεδομένο ένα σύνολο S που αποτελείται από n έγγραφα, ζητείται να προσδιοριστεί μια κατάτμηση του συνόλου σε ένα προκαθορισμένο πλήθος k υποσυνόλων $S_1, S_2, ..., S_k$ τέτοια ώστε ο βαθμός ομοιότητας των εγγράφων που ανατίθενται στο κάθε υποσύνολο να είναι αυξημένος σε σχέση με τα έγγραφα που ανατίθενται στα υπόλοιπα υποσύνολα.

Η ζητούμενη επίτευξη του αυξημένου βαθμού ομοιότητας είναι μια διαδικασία στενά συνδεδεμένη με την βελτιστοποίηση της τιμής μιας συνάρτησης κριτηρίου που ορίζεται πάνω στο συνολικό προκύπτον σχήμα ομαδοποίησης. Συνεπώς, το πρόβλημα της ομαδοποίησης μετατοπίζεται στον προσδιορισμό ενός σχήματος ομαδοποίησης που βελτιώνει στον καλύτερο δυνατό βαθμό μια συγκεκριμένη συνάρτηση κριτηρίου. Στην συνέχεια παρουσιάζονται διάφορες συναρτήσεις κριτηρίου οι οποίες είναι δυνατό να χρησιμοποιηθούν τόσο για την αξιολόγηση μιας λύσης ομαδοποίησης όσο και για την καθοδήγηση της όλης διαδικασίας.

Μια κατηγορία συναρτήσεων κριτηρίου για σχήματα ομαδοποίησης είναι οι «εσωτερικές» συναρτήσεις οι οποίες εστιάζουν στην παραγωγή μιας λύσης ομαδοποίησης η οποία βελτιστοποιεί την τιμή της συνάρτησης κριτηρίου που ορίζεται πάνω στα έγγραφα κάθε ομάδας χωρίς να λαμβάνεται υπόψη η σχέση των εγγράφων που ανατίθενται σε διαφορετικές ομάδες [KZ02].

Η πρώτη εσωτερική συνάρτηση κριτηρίου που εξετάζεται μεγιστοποιεί το άθροισμα των μέσων βαθμών ομοιότητας μεταξύ των εγγράφων που ανατίθενται σε μια ομάδα και εφαρμόζει στάθμιση που βασίζεται στο μέγεθος της κάθε ομάδας. Συγκεκριμένα, αν θεωρήσουμε ότι χρησιμοποιείται η συνάρτηση συνημίτονου για την μέτρηση του βαθμού ομοιότητας μεταξύ των εγγράφων τότε επιθυμούμε όπως μια λύση ομαδοποίησης μεγιστοποιεί την συνάρτηση κριτηρίου

$$I_1 = \sum_{r=1}^{k} n_r \cdot \left(\frac{1}{n_r^2} \cdot \sum_{\vec{d}_i, \vec{d}_j \in S_r} cos\left(\vec{d}_i, \vec{d}_j\right) \right)$$

Σημειώνεται εδώ ότι η πιο πάνω διατύπωση για την συνάρτηση κριτηρίου λαμβάνει υπόψη και τους βαθμούς ομοιότητας κάθε εγγράφου μιας ομάδας με τον εαυτό του. Δεδομένου ότι ο

συγκεκριμένος βαθμός ομοιότητας είναι πάντοτε μέγιστος για όλα τα έγγραφα και συνεπώς λαμβάνεται ως κοινός παρονομαστής για όλα τα έγγραφα όλων των ομάδων, δεν προκύπτει ζήτημα εσφαλμένης εκτίμησης της τιμής της συνάρτησης.

Η δεύτερη εσωτερική συνάρτηση κριτηρίου χρησιμοποιείται κυρίως στον δημοφιλή αλγόριθμο K-Means και στοχεύει στην μεγιστοποίηση της ομοιότητας μεταξύ του κεντροειδούς διανύσματος κάθε ομάδας με τα διανύσματα αναπαράστασης των εγγράφων που ανατίθενται στην συγκεκριμένη ομάδα. Χρησιμοποιώντας και πάλι την συνάρτηση συνημίτονου για τον προσδιορισμό του βαθμού ομοιότητας, η συνάρτηση κριτηρίου διατυπώνεται ως

$$I_2 = \sum_{r=1}^{k} \sum_{\vec{d}_i \in S_r} cos\left(\vec{d}_i, \vec{C}_r\right)$$

Τέλος, μια άλλη συνάρτηση κριτηρίου που χρησιμοποιείται ευρέως σκοπεύει στην ελαχιστοποίηση των τετραγωνικών αποστάσεων που προκύπτουν για τα έγγραφα μιας ομάδας και του αντίστοιχου κεντροειδούς διανύσματος και πιο συγκεκριμένα

$$I_3 = \sum_{r=1}^{k} \sum_{\vec{d}_i \in S_r} \left\| \vec{d}_i - \vec{C}_r \right\|^2$$

Με απλούς αλγεβρικούς χειρισμούς των εκφράσεων για τις πιο πάνω συναρτήσεις κριτηρίου μπορεί εύκολα να διαπιστωθεί ότι η συνάρτηση I_3 είναι παρόμοια με την I_1 καθώς οι ομοιότητες εκφράζονται με χρήση των τετραγωνικών αποστάσεων αντί των εσωτερικών γινομένων. Γενικά, οι τρεις διατυπώσεις είναι ισοδύναμες και η χρήση κάποιας συγκεκριμένης εξ αυτών εξαρτάται από την ευκολία με την οποία μπορεί να υλοποιηθεί.

Μια άλλη κατηγορία συναρτήσεων κριτηρίου για σχήματα ομαδοποίησης είναι οι «εξωτερικές» συναρτήσεις οι οποίες εστιάζουν στην παραγωγή μιας λύσης ομαδοποίησης η οποία βελτιστοποιεί την τιμή της συνάρτησης κριτηρίου που ορίζεται πάνω στις διάφορες ομάδες του σχήματος ομαδοποίησης.

Στην περίπτωση αυτή, η διαισθητική προφανής λύση μιας συνάρτησης η οποία να ελαχιστοποιεί τον βαθμό ομοιότητας μεταξύ των κεντροειδών διανυσμάτων των διάφορων ομάδων δεν είναι η πλέον αποδοτική. Μια τέτοια λύση με κεντροειδή διανύσματα ορθογώνια μεταξύ τους θα είχε ως αποτέλεσμα ένα σχήμα ομαδοποίησης στο οποίο οι $k-1$ ομάδες θα περιείχαν από ένα ακριβώς έγγραφο με την μέγιστη ανομοιότητα ως προς το σύνολο των υπόλοιπων εγγράφων και την k-οστή ομάδα να περιέχει όλα τα υπόλοιπα έγγραφα. Για τον λόγο αυτό, μια εξωτερική συνάρτηση που έχει προταθεί στοχεύει στον διαχωρισμό των εγγράφων κάθε ομάδας από το σύνολο των εγγράφων παρά στον διαχωρισμό των εγγράφων των διάφορων ομάδων με την ελαχιστοποίηση της ποσότητας

$$E_1 = \sum_{r=1}^{k} n_r \cdot cos\left(\vec{C}_r, \vec{C}\right)$$

όπου C είναι το κεντροειδές διάνυσμα ολόκληρης της συλλογής των εγγράφων.

Κατ' αναλογία με τις εσωτερικές συναρτήσεις κριτηρίου μπορεί να οριστεί μια επιπλέον συνάρτηση με διαφορετική διατύπωση που στην περίπτωση αυτή επιθυμούμε όπως η λύση ομαδοποίησης μεγιστοποιεί την τιμή της:

$$E_2 = \sum_{r=1}^{k} n_r \cdot \left\|\vec{C}_r - \vec{C}\right\|^2$$

Η συνάρτηση αυτή χρησιμοποιεί τις ευκλείδειες αποστάσεις μεταξύ των κεντροειδών διανυσμάτων των ομάδων ως προς το κεντροειδές διάνυσμα ολόκληρης της συλλογής και αποτελεί μια εναλλακτική διατύπωση σε σχέση με την χρήση της συνάρτησης συνημίτονου.

1.11 Βελτιστοποίηση συναρτήσεων κριτηρίου

Υπάρχουν αρκετές μέθοδοι οι οποίες μπορούν να εφαρμοστούν για την βελτιστοποίηση των συναρτήσεων κριτηρίου που αναφέρθηκαν στα προηγούμενα. Ένας συνηθισμένος τρόπος επίτευξης αυτής της βελτιστοποίησης είναι με την χρήση μιας «άπληστης» στρατηγικής (*greedy strategy*). Αυτές οι «άπληστες» προσεγγίσεις χρησιμοποιούνται συνήθως μέσα στα πλαίσια κατατμητικών αλγόριθμων ομαδοποίησης (*partitional clustering algorithms*) όπως τον γνωστό αλγόριθμο K-Means και έχει αποδειχτεί ότι αρκετές συναρτήσεις κριτηρίου συγκλίνουν τελικά σε κάποιο τοπικό ελάχιστο (ή αντίστοιχα μέγιστο).

Η βελτιστοποίηση με «άπληστη» στρατηγική αποτελείται από δύο φάσεις. Στην πρώτη φάση της «αρχικής ομαδοποίησης» κατασκευάζεται μια λύση ομαδοποίησης επιλέγοντας τυχαία k πρότυπα για να αποτελέσουν τα κέντρα των αντίστοιχων ομάδων της λύσης. Ακολούθως υπολογίζεται ο βαθμός ομοιότητα κάθε εγγράφου με τα k κέντρα χρησιμοποιώντας την συνάρτηση συνημίτονου ή την ευκλείδεια απόσταση και τα έγγραφα ανατίθενται στις ομάδες που αντιστοιχούν στα κέντρο με την μεγαλύτερη ομοιότητα. Αυτή η προσέγγιση οδηγεί σε μια αρχική λύση ομαδοποίησης.

Κατά την δεύτερη φάση της «βελτίωσης των ομάδων» (*cluster refinement*) ο στόχος είναι η σταδιακή διαδοχική βελτίωση της λύσης ομαδοποίησης. Σε κάθε στάδιο (επανάληψη) υπολογίζονται εκ νέου οι βαθμοί ομοιότητας όλων των εγγράφων σε σχέση με τα κέντρα των ομάδων όπως είχαν διαμορφωθεί από την προηγούμενη κατανομή και εκτελούνται οι αναγκαίες μετακινήσεις εγγράφων από ομάδα σε ομάδα κατά τρόπο τέτοιο ώστε πάντοτε να ικανοποιείται το κριτήριο της μέγιστης ομοιότητας εγγράφου-ομάδας. Αυτή η στρατηγική βελτίωσης χαρακτηρίζεται ως μαζική (*batch*) καθώς τα κέντρα των ομάδων ανανεώνονται μετά την νέα

κατανομή όλων των εγγράφων. Εναλλακτικά, είναι δυνατό η ανανέωση των κέντρων να εκτελείται αμέσως μετά την μετακίνηση του κάθε εγγράφου και σε αυτή την περίπτωση η στρατηγική χαρακτηρίζεται άμεση (*online*).

2

Σχεδίαση και ανάπτυξη ευφυούς πράκτορα

2.1 Εισαγωγή

Η θεωρία που αφορά στο θέμα της οργάνωσης, ανάκτησης και χρήσης πληροφορίας που αναπτύχθηκε στο προηγούμενο κεφάλαιο και ειδικότερα οι τεχνικές γύρω από την διαδικασία ομαδοποίησης εγγράφων χρησιμοποιήθηκαν στην πράξη για την ανάπτυξη μιας εφαρμογής-ευφυούς πράκτορα, του SCAgent (***S****earch and* ***C****luster* ***Agent***) με στόχο την δημιουργία ενός βοηθητικού εργαλείου το οποίο να παρέχει στον χρήστη αφενός την δυνατότητα διενέργειας μιας πιο αποτελεσματικής έρευνας και ανάκτησης εγγράφων από το Διαδίκτυο και αφετέρου την παροχή μιας πλατφόρμας για την εφαρμογή τεχνικών ομαδοποίησης για σκοπούς μελέτης του αντικειμένου της ομαδοποίησης εγγράφων.

Η υλοποίηση της εφαρμογής SCAgent έγινε με χρήση του περιβάλλοντος προγραμματισμού Visual Studio .NET 2003™ της εταιρείας Microsoft™ και πιο συγκεκριμένα ως γλώσσα προγραμματισμού στην οποία συντάχθηκε εξ' ολοκλήρου ο κώδικας χρησιμοποιήθηκε η Microsoft Visual Basic© έκδοση 7.1.

2.2 Απαιτήσεις

Για την σωστή εκτέλεση και χρήση της εφαρμογής απαιτείται όπως εγκατασταθεί προηγουμένως στο μηχάνημα του χρήστη η πλατφόρμα εκτέλεσης εφαρμογών Microsoft Framework .NET© έκδοσης 1.1. Το συγκεκριμένο περιβάλλον εκτέλεσης διανέμεται προς ελεύθερη χρήση από την σχετική ιστοσελίδα της Microsoft™ και η εγκατάσταση του αποτελεί απαραίτητη προϋπόθεση για την δυνατότητα εκτέλεσης οποιασδήποτε εφαρμογής που έχει αναπτυχθεί σε κάποια από τις γλώσσες προγραμματισμού που περιλαμβάνονται στο περιβάλλον του Visual Studio .NET 2003©. Σημειώνεται ότι στο CD που συνοδεύει τον παρόν τόμο περιέχεται το Framework .NET© έκδοση 1.1 και καθώς η σχετική εφαρμογή εγκατάστασης του SCAgent δεν το εγκαθιστά αυτόματα στο μηχάνημα του χρήστη, προτείνεται όπως η εγκατάσταση του προηγηθεί από τον ίδιο τον χρήστη.

Επιπλέον, η εφαρμογή χρησιμοποιεί για την εκτέλεση της έρευνας στο Web την δημοφιλή μηχανή αναζήτησης Google και ειδικότερα το σχετικό API που διανέμεται δωρεάν από την ιστοσελίδα http://www.google.com/apis. Η εφαρμογή εγκατάστασης του SCAgent εγκαθιστά τις απαραίτητες βιβλιοθήκες του API οι οποίες ενημερώνονται αυτόματα από τον δικτυακό τόπο του Google.

2.3 Περιγραφή υλοποίησης

Η εφαρμογή SCAgent αποτελείται από ένα αριθμό κλάσεων οι οποίες συνεργαζόμενες και αλληλεπιδρώντας συνθέτουν τον λειτουργικό ιστό της εφαρμογής. Μια σύντομη περιγραφή των κλάσεων αυτών παρατίθεται πιο κάτω:

▪ *Κλάση formMain*

Αποτελεί τον πυρήνα της εφαρμογής καθώς κατά πρώτο λόγο υλοποιεί και διαχειρίζεται την γραφική διεπαφή μεταξύ εφαρμογής-χρήστη (*Graphical User Interface, GUI*) και κατά δεύτερο λόγο συντονίζει και χρησιμοποιεί τις υπόλοιπες κλάσεις για την διαχείριση και επεξεργασία των δεδομένων της εφαρμογής. Όλες οι λειτουργίες της εφαρμογής που παρέχονται στον χρήστη ορίζονται σε αυτή την κλάση, η οποία περιορίζει τον ρόλο των υπολοίπων κλάσεων στην μοντελοποίηση και εσωτερική αναπαράσταση των οντοτήτων που εισάγονται και χρησιμοποιούνται στην εφαρμογή.

▪ *Κλάση classCluster*

Αποτελεί την μοντελοποίηση της οντότητας 'cluster'. Η κλάση παρέχει μεθόδους δημιουργίας ενός cluster, διαχείρισης του ιδίου και των εγγράφων που αυτό περιλαμβάνει και παρέχει πρόσβαση σε ένα σύνολο ιδιοτήτων του όπως αυτές διαμορφώνονται με την δυναμική εισαγωγή-διαγραφή εγγράφων κατά τον χρόνο εκτέλεσης της εφαρμογής.

▪ *Κλάση classScheme*

Αποτελεί την μοντελοποίηση της οντότητας 'scheme' η οποία εισάγεται ως περιγραφή του γενικότερου σχήματος κατάτμησης που κατασκευάζεται από την εφαρμογή. Από δομικής άποψης τα clusters που κατασκευάζει η εφαρμογή (στιγμιότυπα της κλάσης classCluster) αποτελούν συστατικά στοιχεία του scheme κατάτμησης και η διαχείριση τους πραγματοποιείται αποκλειστικά μέσω αυτού. Η κλάση παρέχει μεθόδους για την δημιουργία-διαγραφή clusters και πρόσβαση σε ιδιότητες που χαρακτηρίζουν το σχήμα κατάτμησης στην ολότητα του.

▪ *Κλάση classResult*

Αποτελεί την μοντελοποίηση της οντότητας 'έγγραφο' που προκύπτει ως αποτέλεσμα της αναζήτησης του χρήστη. Η κλάση παρέχει μεθόδους διαχείρισης του ιδίου του εγγράφου καθώς και πρόσβασης στις ιδιότητες του και γενικά παρέχει μια αναπαράσταση του εγγράφου χρήσιμη για την εφαρμογή αλλά χωρίς να αποτελεί μια ολοκληρωμένη εκδοχή του κατάλληλη για απευθείας αυτόνομη χρήση εκτός του πλαισίου της διαδικασίας ομαδοποίησης.

▪ *Κλάση classQuery*

Αποτελεί την μοντελοποίηση της οντότητας ‘query’ η οποία κατ’ αναλογία με την οντότητα scheme εισάγεται ως περιγραφή του ολοκληρωμένης συλλογής δεδομένων τα οποία ανακτώνται μετά από έρευνα του χρήστη για επεξεργασία από την εφαρμογή. Η κλάση παρέχει μεθόδους για την διαχείριση των εγγράφων (στιγμιότυπα της κλάσης classResult) και πρόσβαση σε ιδιότητες που χαρακτηρίζουν ολόκληρη την συλλογή των εγγράφων.

▪ *Κλάση classVector*

Αποτελεί την μοντελοποίηση ενός διανύσματος πραγματικών αριθμών μεταβλητού μεγέθους. Η κλάση παρέχει μεθόδους δημιουργίας, τροποποίησης και εκτέλεσης πράξεων επί του διανύσματος καθώς και πρόσβαση σε διάφορες ιδιότητες του.

▪ *Κλάση classKernel*

Αποτελεί την μοντελοποίηση μιας συλλογής διαφόρων μεταβλητών τις οποίες χρησιμοποιεί η εφαρμογή. Χρησιμοποιείται για λόγους ευχρηστίας, αποδοτικότητας, ελαχιστοποίησης και ορθής λειτουργίας του κώδικα της εφαρμογής καθώς με την μέθοδο της ‘βραχυκύκλωσης’ συμβάντων (*events*) απλοποιεί την διαχείριση όλων των μεταβλητών global εμβέλειας που χρησιμοποιούνται από διαφορετικά τμήματα της εφαρμογής.

▪ *Κλάση moduleMain*

Αποτελεί μια ειδική μορφή κλάσης η οποία παρέχει μεθόδους με λειτουργίες γενικής φύσεως. Αυτές οι μέθοδοι είναι συνήθως μικρά και συχνά χρησιμοποιούμενα τμήματα κώδικα τα οποία για λόγους αποδοτικότητας και επαναχρησιμοποίησης κώδικα (*code reusability*) ομαδοποιούνται στα πλαίσια αυτής της κλάσης για χρήση όποτε απαιτείται. Η κλάση είναι ειδικής μορφής καθώς είναι μια αποκλειστικά shared κλάση, δηλαδή δεν μπορεί να δημιουργηθεί κάποιο στιγμιότυπο αυτής ούτε και είναι δυνατό να κληρονομηθεί από κάποια άλλη κλάση. Οι μέθοδοι και τα γνωρίσματα της είναι ορατά και διαθέσιμα απ΄ όλες τις κλάσεις της εφαρμογής και μπορούν να χρησιμοποιηθούν σαν αντικείμενα global εμβέλειας. Η κλάση αυτή περιέχει την μέθοδο main() η οποία αποτελεί και το σημείο εκκίνησης της εφαρμογής.

2.4 Περιγραφή των σημαντικότερων λειτουργιών της εφαρμογής

Η κλάση formMain κληρονομείται από την κλάση System.Windows.Forms.Form του συστήματος και αποτελεί ένα container γραφικών στοιχείων διεπαφής για την εφαρμογή. Σε όλη την διάρκεια εκτέλεσης της εφαρμογής δημιουργείται και χρησιμοποιείται μόνο ένα στιγμιότυπο

αυτής της κλάσης. Κατά την φάση δημιουργίας αυτού του στιγμιότυπου δημιουργούνται και αρχικοποιούνται όλα τα γραφικά στοιχεία διεπαφής που χρησιμοποιούνται στην εφαρμογή. Η κλάση δεν διαθέτει γνωρίσματα (*attributes*) με την στενή έννοια των μεταβλητών, καθώς δεν αποτελεί μοντελοποίηση κάποιας οντότητας που χρησιμοποιείται στην εφαρμογή αλλά γνωρίσματα της κλάσης αποτελούν όλα τα γραφικά στοιχεία διεπαφής της εφαρμογής. Οι μέθοδοι της κλάσης αποτελούν κατά κύριο λόγο χειριστές συμβάντων (*event handlers*) με τα οποία οδηγείται η εφαρμογή (*event-driven application*). Επιπλέον, στην κλάση περιλαμβάνονται μέθοδοι για την εκτέλεση των βασικών λειτουργιών της εφαρμογής καθώς και ορισμένες μέθοδοι που εκτελούν βοηθητικές λειτουργίες.

Πιο κάτω σημειώνονται οι σημαντικότερες λειτουργίες της εφαρμογής που ενσωματώνονται ως μέθοδοι της κλάσης formMain:

- `ExecuteQuery()`

  Η μέθοδος αυτή πραγματοποιεί την έρευνα που ορίζει ο χρήστης χρησιμοποιώντας την μηχανή αναζήτησης Google. Λόγω κατασκευαστικών περιορισμών του API που διανέμενεται από το Google, η λήψη των αποτελεσμάτων της αναζήτησης γίνεται κατά στάδια μέχρι να ικανοποιηθεί το μέγιστο πλήθος αποτελεσμάτων που ορίζει ο χρήστης. Μετά από κάθε πετυχημένη λήψη ενός τμήματος των αποτελεσμάτων, δημιουργείται καινούρια στιγμιότυπα της κλάσης classResult (μέσω της κλάσης χειρισμού classQuery) στα οποία καταχωρούνται τα δεδομένα που αφορούν στα ληφθέντα αποτελέσματα. Στην περίπτωση ανίχνευσης σφάλματος κατά την διάρκεια εκτέλεσης ή λήψης αποτελεσμάτων της έρευνας ή στην περίπτωση που ο χρήστης επιλέξει διακοπής της διαδικασίας, όλα τα δεδομένα που ενδεχομένως είχαν ληφθεί μέχρι εκείνο το σημείο ακυρώνονται και η μέθοδος τερματίζει την εκτέλεση της. Κατά την διάρκεια της φάσης λήψης των αποτελεσμάτων, στον χώρο πληροφοριών εμφανίζονται πληροφοριακά στοιχεία που περιγράφουν την εξέλιξη της διαδικασίας. Η επιτυχής ολοκλήρωσης της διαδικασίας έρευνας και ανάκτησης των αποτελεσμάτων καθιστά δυνατή την μετάβαση στο επόμενο βήμα της εφαρμογής με την ενεργοποίηση του σχετικού πλήκτρου 'Next'.

- `ExecuteTransformation()`

  Η εκτέλεση αυτής της μεθόδου είναι δυνατή εφόσον έχει προηγηθεί επιτυχημένη λήψη των αποτελεσμάτων κάποιας έρευνας. Κατά την διαδικασία της λήψης των αποτελεσμάτων, ανακτάται το περιεχόμενο του κάθε αποτελέσματος με σκοπό την ανίχνευση των όρων-κλειδιών του (*keywords*). Την ανίχνευση των όρων-κλειδιών κάθε αποτελέσματος ακολουθεί η κατασκευή του υπερσυνόλου των όρων-κλειδιών για το σύνολο όλων των αποτελεσμάτων της έρευνας. Ο χρήστης έχει την δυνατότητα να επιλέξει τον αποκλεισμό κάποιων όρων-κλειδιών από την συνέχεια της επεξεργασίας εφόσον κρίνει

ότι ο αποκλεισμός θα βελτιώσει την διαδικασία της ομαδοποίησης. Αφού προσδιοριστούν οι όροι-κλειδιά που θα κρατηθούν για την επεξεργασία, η μέθοδος κατασκευάζει τα διανύσματα όρων με τα οποία αναπαρίστανται τα αποτελέσματα.

- `ExecuteKeywordReduction()`

  Κατά την διαδικασία της κατασκευής των διανυσμάτων αναπαράστασης των αποτελεσμάτων, υπολογίζεται παράλληλα το διάνυσμα μέσων συχνοτήτων των όρων ως προς τις αντίστροφες συχνότητες εγγράφων (*term frequencies–inverse document frequencies, TFIDF*) για λόγους επιτάχυνσης της όλης διαδικασίας. Η μέθοδος σε πρώτη φάση επεξεργάζεται αυτό το διάνυσμα και σε συνδυασμό με την τιμή κατωφλίου που έχει ορίζει ο χρήστης, εντοπίζει τους όρους που ικανοποιούν την συνθήκη κατωφλίου και πρέπει να κρατηθούν ως σημαντικοί στην συνέχεια της επεξεργασίας. Στην δεύτερη φάση, για κάθε έγγραφο κατασκευάζεται το μειωμένης διάστασης διάνυσμα που περιέχει αυτούς τους σημαντικούς όρους και το οποίο πλέον χρησιμοποιείται στα επόμενα βήματα επεξεργασίας. Η τιμή κατωφλίου που εφαρμόζεται είναι δυνατό να αποκόψει μικρό ή μεγάλο αριθμό όρων καταλήγοντας σε μικρή ή μεγάλη μείωση αντίστοιχα της διάστασης του διανυσματικού μοντέλου αναπαράστασης. Ο χρήστης εκτελώντας διαδοχικές επαναλήψεις της μεθόδου με διαφορετικές τιμές κατωφλίου μπορεί να επιτύχει μείωση της διάστασης σε επιθυμητά επίπεδα καθώς σε κάθε εκτέλεση της μεθόδου οποιοδήποτε προηγούμενο αποτέλεσμα ενδεχομένως υπήρχε αντικαθίσταται με το νέο που προκύπτει από την πιο πρόσφατη εκτέλεση. Τέλος, σημειώνεται ότι η διακοπή της επεξεργασίας πριν την ολοκλήρωση της έχει ως αποτέλεσμα την ακύρωση όλων των αποτελεσμάτων που είχαν υπολογιστεί μέχρι την στιγμή εκείνη και καθιστά αδύνατη την μετάβαση στο επόμενο βήμα της εφαρμογής.

- `ExecuteKeywordWeighting()`

  Η μέθοδος αυτή προσδιορίζει ενδεικτικά βάρη των όρων του διανυσματικού μοντέλου αναπαράστασης εγγράφων τα οποία αποτελούν μέτρο της διακριτικής ικανότητας του κάθε όρου. Κατά την εκτέλεση της μεθόδου χρησιμοποιούνται τα έγγραφα εκείνα τα οποία ο χρήστης έχει επιλέξει ως σημαντικά και τα οποία θεωρούνται ότι αντιπροσωπεύουν πιο χαρακτηριστικά το θεματικό ενδιαφέρον του χρήστη. Στα έγγραφα αυτά πρέπει να περιέχεται τουλάχιστον ένα έγγραφο το οποίο να διαθέτει πληροφορία όρων-κλειδιών καθώς σε διαφορετική περίπτωση δεν είναι δυνατή η εξαγωγή χρήσιμης πληροφορίας για την ανίχνευση των ενδιαφερόντων του χρήστη. Από τα διανύσματα αναπαράστασης των εγγράφων κατασκευάζεται το αντίστοιχο κεντροειδές ως χαρακτηριστικός αντιπρόσωπος το οποίο χρησιμοποιείται ως διάνυσμα βάρους των όρων κατά την εκτέλεση του αλγόριθμου κατάτμησης. Η εκτέλεση της μεθόδου παρακάμπτεται

στην περίπτωση που ο χρήστης επιλέξει την μη χρήση βαρών για τους όρους και η μετάβαση στο επόμενο βήμα είναι επιτρεπτή κατ' εξαίρεση του γενικής δομής της εφαρμογής. Σε περίπτωση διακοπής της επεξεργασίας πριν αυτή ολοκληρωθεί, το διάνυσμα βάρους των όρων παύει να είναι διαθέσιμο και για τον υπολογισμό του πρέπει να γίνει εκ νέου κλήση εκτέλεσης της μεθόδου.

- `ExecuteClustering()`

  Η μέθοδος εκτελεί τον αλγόριθμο ομαδοποίησης K-Means με είσοδο τα δεδομένα που ανακτήθηκαν από την έρευνα του χρήστη και έτυχαν της κατάλληλης επεξεργασίας στα προηγούμενα βήματα. Η χρήση της σταθμισμένης εκδοχής ή όχι του αλγόριθμου ανιχνεύεται αυτόματα από την ύπαρξη διανύσματος στάθμισης και κατά την διάρκεια εκτέλεσης του αλγόριθμου παρουσιάζονται οπτικές ενδείξεις για την πρόοδο της διαδικασίας και την εξέλιξη των τιμών των μέτρων επίδοσης του σχήματος ομαδοποίησης. Επιπλέον, εμφανίζονται πληροφορίες για την μορφή της λύσης σε κάθε επαναληπτικό βήμα και δίνεται η δυνατότητα παρακολούθησης της πορείας ανεύρεσης της τελικής λύσης.

- `ExecuteSummary()`

  Η μέθοδος χρησιμοποιείται για την εμφάνιση μιας συνοπτικής περιγραφής των clusters της λύσης που προέκυψε από την εκτέλεση του αλγόριθμου κατάτμησης. Για την οπτικοποίηση χρησιμοποιείται μια δενδροειδής αναπαράσταση η οποία απεικονίζει την κατανομή των εγγράφων στις διάφορες ομάδες. Για κάθε ομάδα είναι διαθέσιμες στατιστικές πληροφορίες καθώς και η μετάβαση σε ομάδες συναφούς περιεχομένου ενώ για κάθε αποτέλεσμα είναι δυνατή η άμεση μετάβαση για επισκόπηση.

2.5 Περιγραφή αρχιτεκτονικής

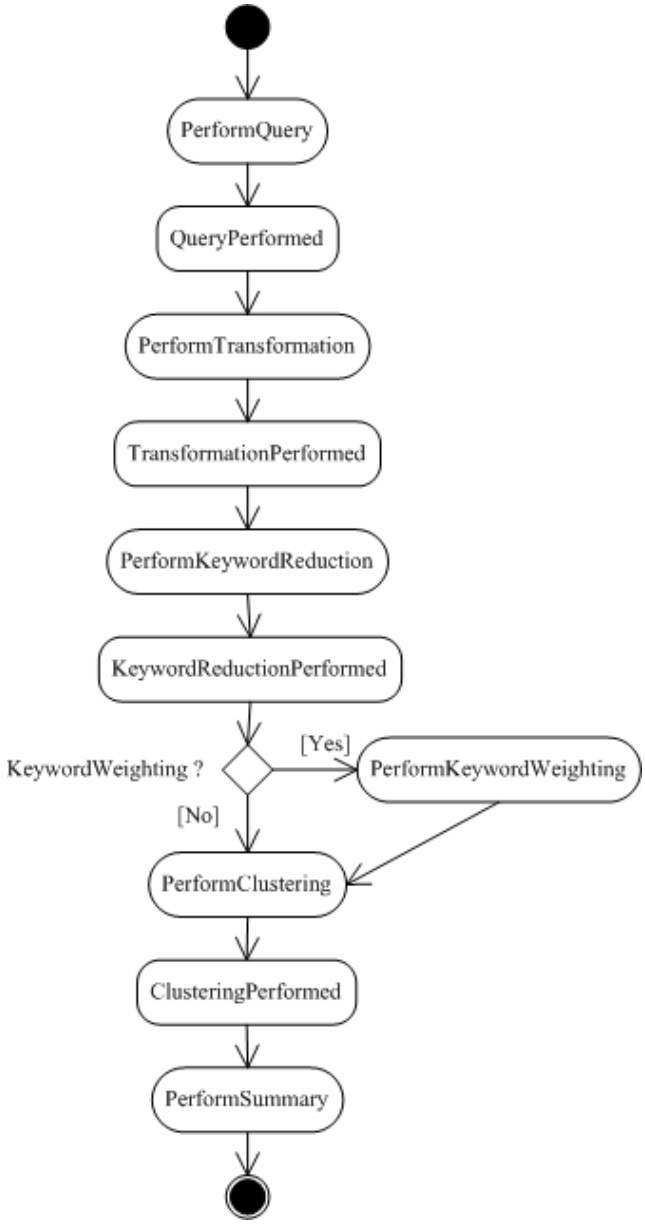
Στην συνέχεια δίνονται μερικά διαγράμματα UML που αφορούν στην αρχιτεκτονική άποψη της εφαρμογής SCAgent.

Το στατικό διάγραμμα (*static ή class diagram*) αποτελεί μια απεικόνιση της αρχιτεκτονικής δομής της εφαρμογής. Στο διάγραμμα απεικονίζονται οι σημαντικότερες κλάσεις που χρησιμοποιούνται και για την κάθε κλάση σημειώνονται τα κυριότερα χαρακτηριστικά και μέθοδοι της. Σημειώνονται επίσης οι συσχετίσεις (*associations*) όπου αυτές υπάρχουν μεταξύ των κλάσεων οι οποίες αναπαριστούν την εξάρτηση (ποσοτική και ποιοτική) των στιγμιότυπων της κάθε κλάσης.

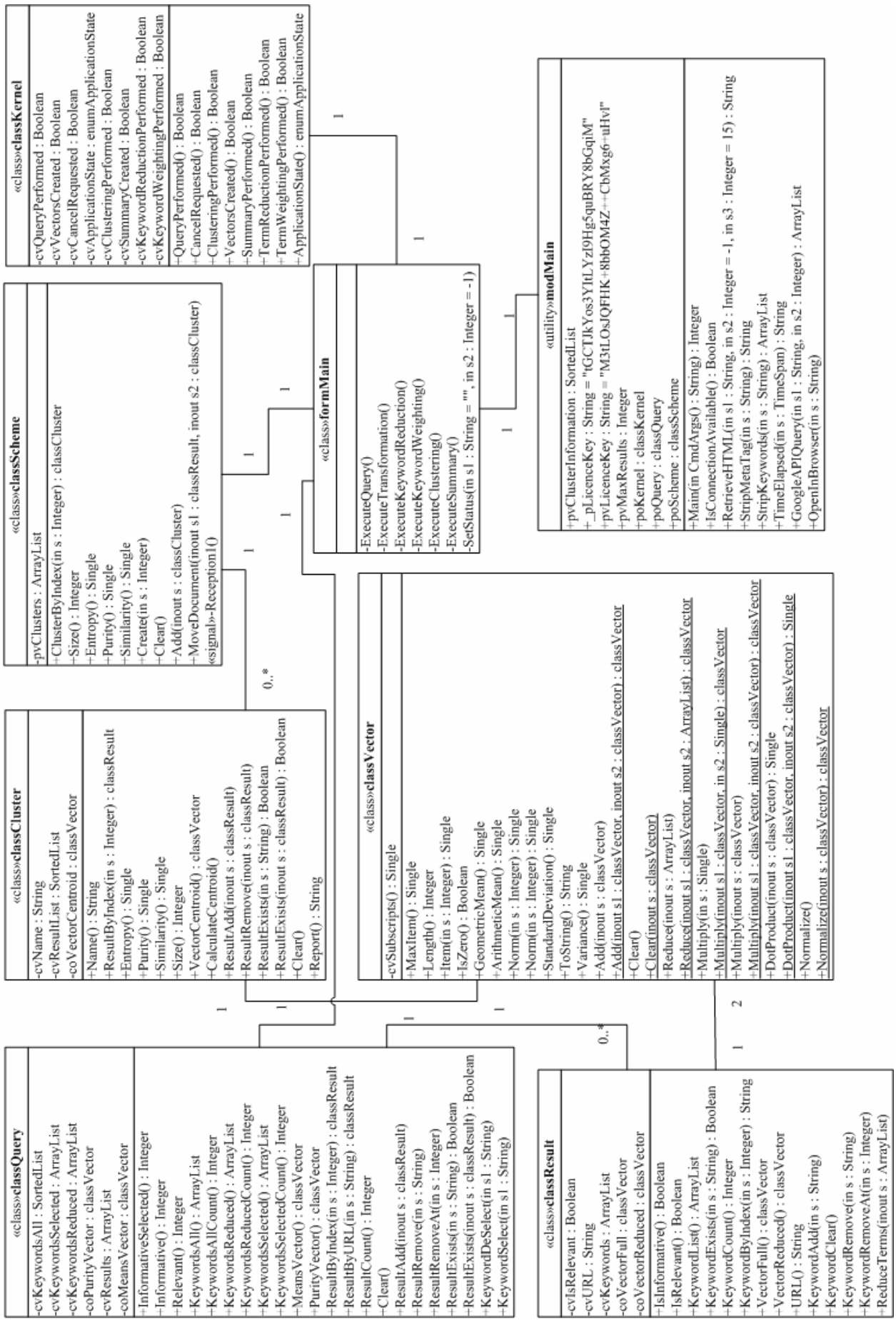
Το διάγραμμα διαδοχής (*sequence diagram*) παρουσιάζει σε γενικές γραμμές την αλληλεπίδραση μεταξύ αντικειμένων κατά χρονική σειρά που συντελείται κατά την διάρκεια μιας τυπικής εκτέλεσης της εφαρμογής. Συγκεκριμένα, αναπαριστά τα αντικείμενα που

συμμετέχουν στις διάφορες ενέργειες με την διάρκεια ζωής τους καθώς και τα μηνύματα που ανταλλάσσουν οργανωμένα κατά χρονική σειρά. Το διάγραμμα χρησιμεύει στην καλύτερη κατανόηση της λογικής σειράς με την οποία εκτελούνται τα διάφορα βήματα που συνθέτουν τον κύκλο εκτέλεσης της εφαρμογής.

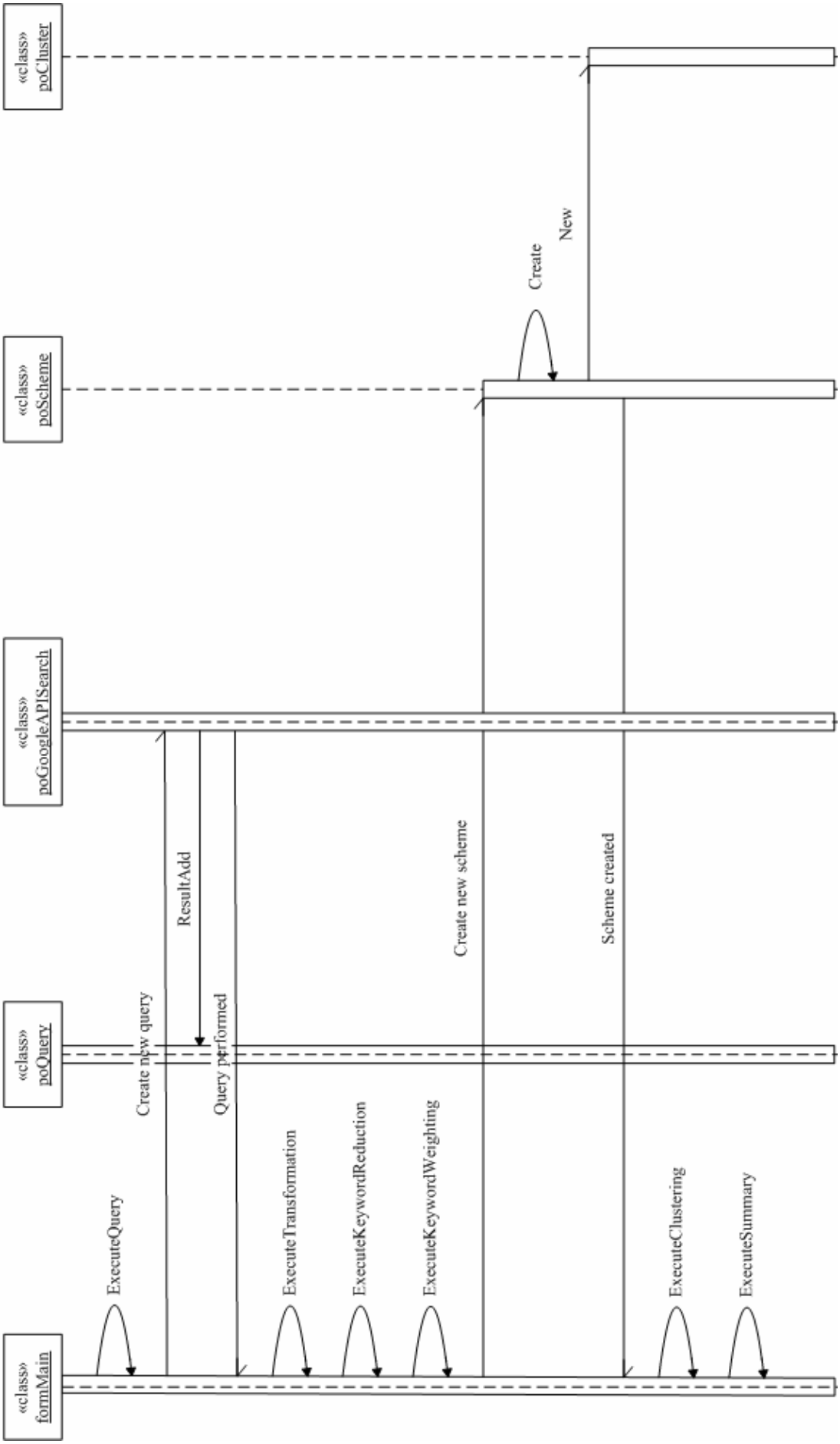
Το διάγραμμα δραστηριότητας (*activity diagram*) αποτελεί μια ειδική περίπτωση ενός διαγράμματος καταστάσεων στο οποίο όλες οι καταστάσεις είναι φάσεις στις οποίες εκτελείται κάποια ενέργεια και οι μεταβάσεις πυροδοτούνται από την συμπλήρωση αυτών των ενεργειών. Σκοπός του διαγράμματος είναι να αναπαραστήσει την ροή εκτέλεσης εσωτερικών διαδικασιών της εφαρμογής χωρίς να συμπεριλαμβάνει εξωτερικά γεγονότα που προκύπτουν από την παρέμβαση του χρήστη.



Σχήμα 2.5.1: Διάγραμμα δραστηριότητας για την εφαρμογή SCAgent



Σχήμα 2.5.2: Στατικό διάγραμμα για την εφαρμογή SCAgent



Σχήμα 2.5.3: Διάγραμμα διαδοχής για την εφαρμογή SCAgent

2.6 Περιγραφή γραφικής διεπαφής χρήστη

Η γραφική διεπαφή εφαρμογής-χρήστη έχει σχεδιαστεί με στόχο την μεγαλύτερη δυνατή απλότητα και φιλικότητα προς τον χρήστη. Για τον σκοπό αυτό, για την διάρθρωση της πορείας εκτέλεσης της εφαρμογής υιοθετήθηκε η προσέγγιση του 'μάγου' (*wizard*). Τα βήματα δηλαδή εκτέλεσης των διαφόρων λειτουργιών της εφαρμογής εκτελούνται με τέτοια σειρά, η οποία είναι εκ των προτέρων καθορισμένη και εξασφαλίζει με λογικό τρόπο την ομαλή και απρόσκοπτη εκτέλεση των λειτουργιών της εφαρμογής. Συγκεκριμένα, τα βήματα εκτέλεσης σε λογικό επίπεδο έχουν ως ακολούθως:

- Ορισμός έρευνας, διεξαγωγή και ανάκτηση των σχετικών αποτελεσμάτων
- Επιλογή των χαρακτηριστικών των αποτελεσμάτων για την συνέχεια της επεξεργασίας
- Μείωση της διάστασης του διανυσματικού μοντέλου αναπαράστασης των εγγράφων
- Προσδιορισμός διανύσματος στάθμισης των όρων του διανυσματικού μοντέλου
- Εκτέλεση αλγορίθμου ομαδοποίησης
- Εμφάνιση σχήματος ομαδοποίησης

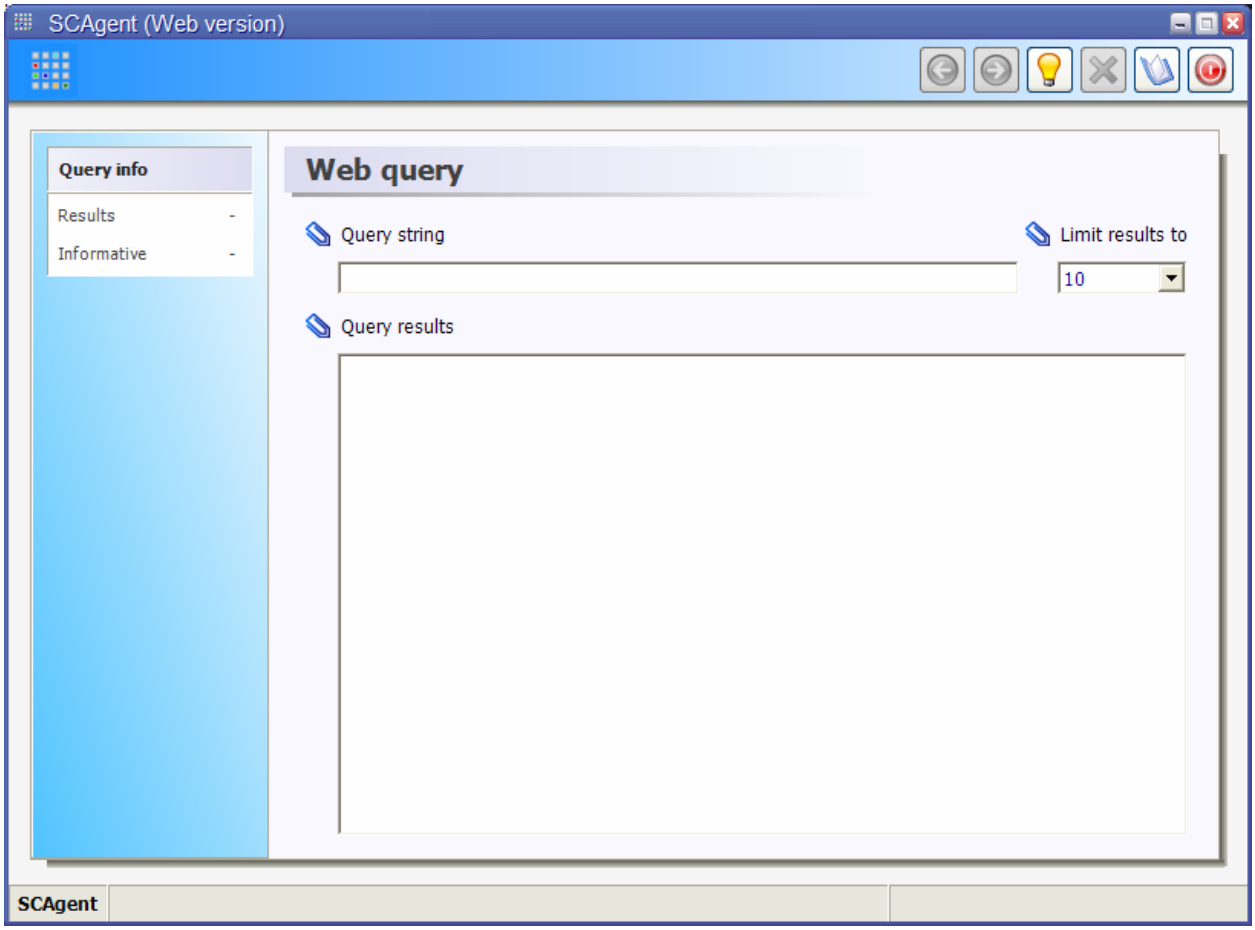
Τα πιο πάνω στάδια εκτέλεσης εφαρμόζονται στην πράξη με αντίστοιχα λογικά βήματα κατά την πορεία εκτέλεσης της εφαρμογής. Έτσι, η εφαρμογή γενικά χρησιμοποιεί έξι αλληλεξαρτημένες όψεις (*views*) που υλοποιούνται με αντίστοιχες οθόνες που παρουσιάζονται στον χρήστη. Η αλληλεξάρτηση μεταξύ αυτών των όψεων προκύπτει από το γεγονός ότι τα αποτελέσματα επεξεργασίας που εξάγονται σε κάποιο από τα πιο πάνω βήματα χρησιμοποιούνται για την μετάβαση στο επόμενο λογικό στάδιο εκτέλεσης της εφαρμογής. Κατ΄ αυτόν τον τρόπο, δεν είναι δυνατή για παράδειγμα η απευθείας μετάβαση από το βήμα της μείωσης της διάστασης του μοντέλου διανυσματικής αναπαράστασης των εγγράφων στο βήμα εκτέλεσης του αλγόριθμου κατάτμησης χωρίς ενδιάμεσα να προηγηθεί ο υπολογισμός βαρών των όρων. Στην περίπτωση που ο χρήστης επιθυμεί σκόπιμα την παράκαμψη κάποιου ενδιάμεσου βήματος, όπως για παράδειγμα όταν επιθυμεί να μην εφαρμόσει μείωση της διάστασης του μοντέλου αναπαράστασης εγγράφων, τότε η παράκαμψη του σχετικού βήματος γίνεται με κατάλληλη επιλογή τιμών για τις παραμέτρους που ζητούνται στο συγκεκριμένο βήμα. Κατά συνέπεια, η διάρθρωση εκτέλεσης της εφαρμογής που περιγράφηκε στα προηγούμενα δεν είναι δεσμευτική από λειτουργικής άποψης αλλά εξυπηρετεί την λογική οργάνωση της εφαρμογής και υποβοηθά τον χρήστη στην ορθή και αποτελεσματική χρήση της.

Η αλληλεπίδραση εφαρμογής - χρήστη έχει υλοποιηθεί επίσης με σκοπό την απλότητα και την ελαχιστοποίηση της συμμετοχής που απαιτείται από τον τελευταίο. Για τον σκοπό αυτό, για την υλοποίηση του μηχανισμού χειρισμού της εφαρμογής χρησιμοποιείται ένα μικρό σύνολο

πλήκτρων ελέγχου τα οποία διατηρούν την ίδια λειτουργικότητα σε ολόκληρη την εφαρμογή, στα πλαίσια που αυτό είναι εφικτό. Αυτό το σύνολο πλήκτρων ελέγχου αποτελείται από πλήκτρα πλοήγησης ανάμεσα στις όψεις - οθόνες της εφαρμογής που περιγράφηκαν στα προηγούμενα καθώς και πλήκτρα για την έκδοση εντολών επεξεργασίας και διακοπής όταν αυτή απαιτείται. Τα πλήκτρα ελέγχου εμφανίζονται στον χρήστη με την μορφή ράβδου εργαλείων (*toolbar buttons*).

Στην συνέχεια, παρατίθενται δείγματα των όψεων - οθονών της εφαρμογής με μια σύντομη περιγραφή της λειτουργικότητας που η κάθε μια παρέχει:

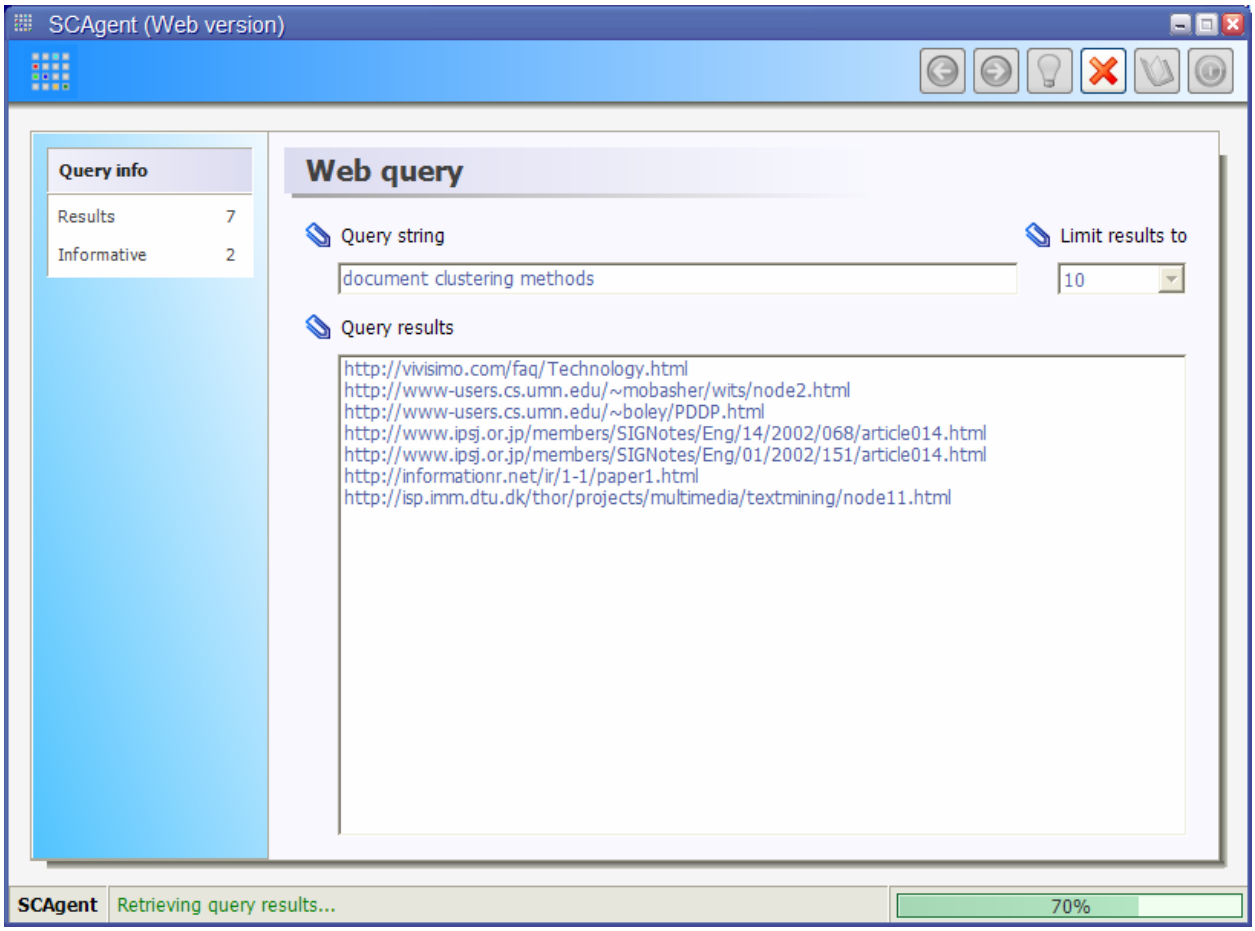
▪ *Όψη καθορισμού έρευνας και ανάκτησης αποτελεσμάτων*



Σχήμα 2.6.1: Εισαγωγική όψη εφαρμογής

Η όψη αυτή αποτελεί την αρχική οθόνη της εφαρμογής. Η διάταξη των γραφικών στοιχείων διεπαφής που απεικονίζεται στο πιο πάνω σχήμα είναι αυτή που χρησιμοποιείται εξ' ολοκλήρου στην εφαρμογή. Αποτελείται από την ράβδο εργαλείων στο επάνω δεξί άκρο, την γραμμή κατάστασης στο κάτω άκρο και τον ενδιάμεσο χώρο εργασίας. Ο ενδιάμεσος χώρος εργασίας χωρίζεται σε δύο κατακόρυφα τμήματα. Το αριστερό τμήμα αποτελεί χώρο εμφάνισης

πληροφοριών που αφορούν στις τιμές διαφόρων μεγεθών που σχετίζονται με το εκάστοτε βήμα εκτέλεσης. Για την συγκεκριμένη όψη, τα χρήσιμα μεγέθη τα οποία απεικονίζονται είναι το μέγεθος του συνόλου αποτελεσμάτων που προέκυψαν από την έρευνα και ο αριθμός αυτών στα οποία ανιχνεύτηκε πληροφορία για λέξεις - κλειδιά. Το δεξί τμήμα του χώρου εργασίας, το οποίο καταλαμβάνει και το μεγαλύτερο μέρος της οθόνης, χρησιμοποιείται για την εισαγωγή τιμών των παραμέτρων οι οποίες είναι αναγκαίες για την εκτέλεση της εργασίας του εκάστοτε βήματος. Στην συγκεκριμένη όψη παρατηρούμε τον χώρο εισαγωγής της φράσης έρευνας, τις επιλογές περιορισμού του μέγιστου πλήθους ανακτώμενων αποτελεσμάτων και τον κατάλογο των αποτελεσμάτων που επιστράφηκαν από την έρευνα.



Σχήμα 2.6.2: Στάδιο εκτέλεσης έρευνας

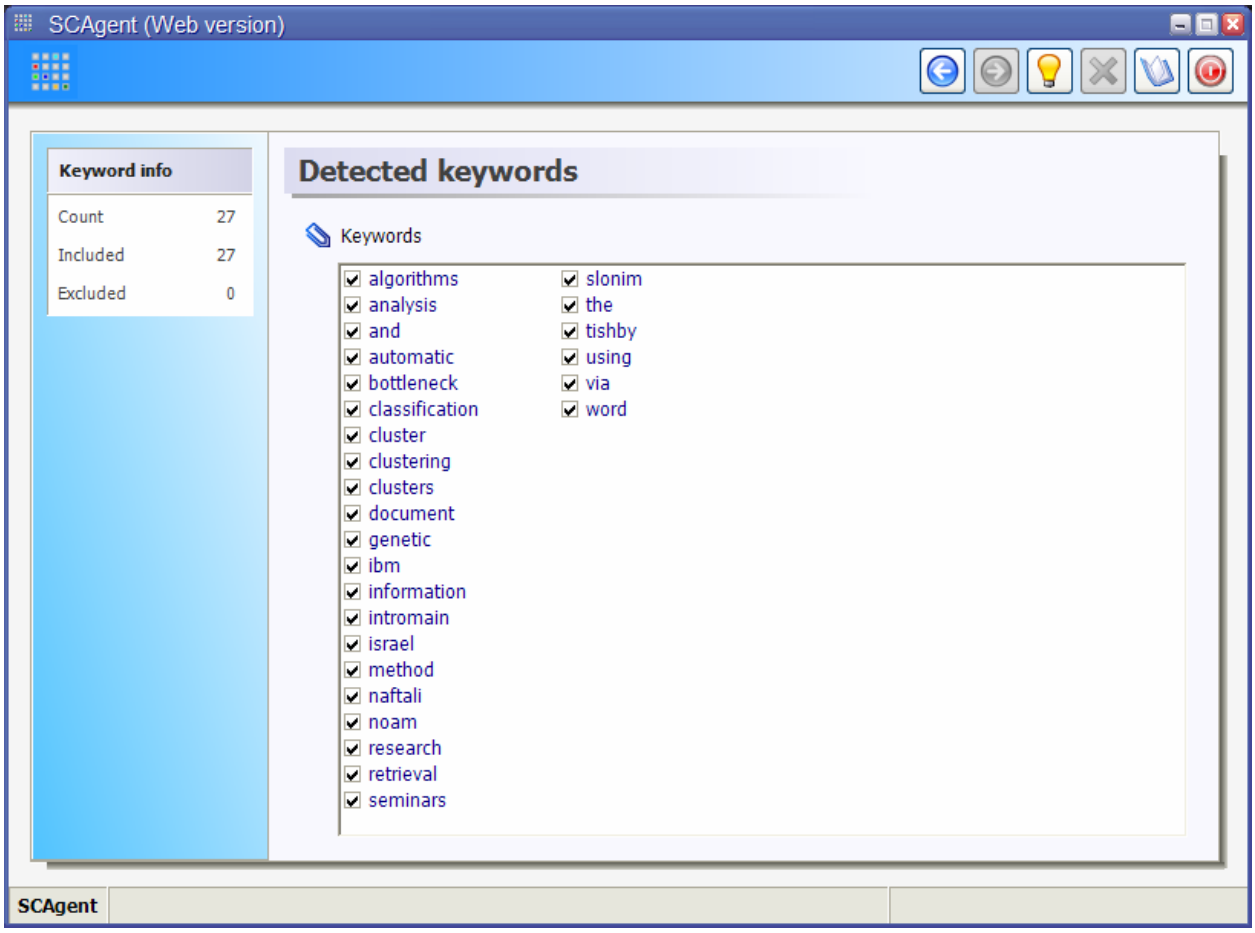
Η αλλαγή της φράσης έρευνας ή του μέγιστου πλήθους ανακτώμενων αποτελεσμάτων ενεργοποιεί το πλήκτρο 'Process' για την έναρξη της διαδικασίας έρευνας. Όπως απεικονίζεται και στα προηγούμενα σχήματα, τα πλήκτρα τα οποία αντιστοιχούν σε ενέργειες που δεν είναι διαθέσιμες κάποια συγκεκριμένη στιγμή του χρόνου εκτέλεσης της εφαρμογής, είναι απενεργοποιημένα. Για το πιο πάνω στιγμιότυπο, η αλλαγή της έρευνας ενεργοποιεί το πλήκτρο 'Process' για την σχετική επεξεργασία αλλά καθώς η επεξεργασία ακόμη δεν έχει

πραγματοποιηθεί, το πλήκτρο 'Next' για πλοήγηση στο επόμενο βήμα της εφαρμογής δεν είναι ακόμη διαθέσιμο. Αυτή η συμπεριφορά της γραφικής διεπαφής εφαρμογής-χρήστη υιοθετείται σε ολόκληρη την εφαρμογή και αποτελεί ένα οπτικό μέσο εποπτείας της εξέλιξης της επεξεργασίας που εκτελεί η εφαρμογή ανά πάσα χρονική στιγμή.

Όταν κάποια διαδικασία επεξεργασίας βρίσκεται σε εξέλιξη, στον χώρο της γραμμής κατάστασης εμφανίζονται πληροφορίες που αφορούν στην πρόοδο της επεξεργασίας που εκτελείται. Στο σχήμα 2.6.2 παρουσιάζεται ένα στιγμιότυπο από το στάδιο εκτέλεσης της έρευνας.

Ειδικότερα, στο αριστερό μέρος της γραμμής κατάστασης εμφανίζεται μια σύντομη περιγραφή της εργασίας που εκτελείται την συγκεκριμένη χρονική στιγμή και δίπλα μια γραφική αναπαράσταση της προόδου αυτής. Κατά την διάρκεια εκτέλεσης της επεξεργασίας, ενεργοποιείται το πλήκτρο 'Cancel' που παρέχει την δυνατότητα διακοπής της επεξεργασίας. Προφανώς όταν εκτελείται κάποια επεξεργασία κάθε άλλη εντολή καθίσταται μη διαθέσιμη μέχρι το πέρας ή την διακοπή της επεξεργασίας.

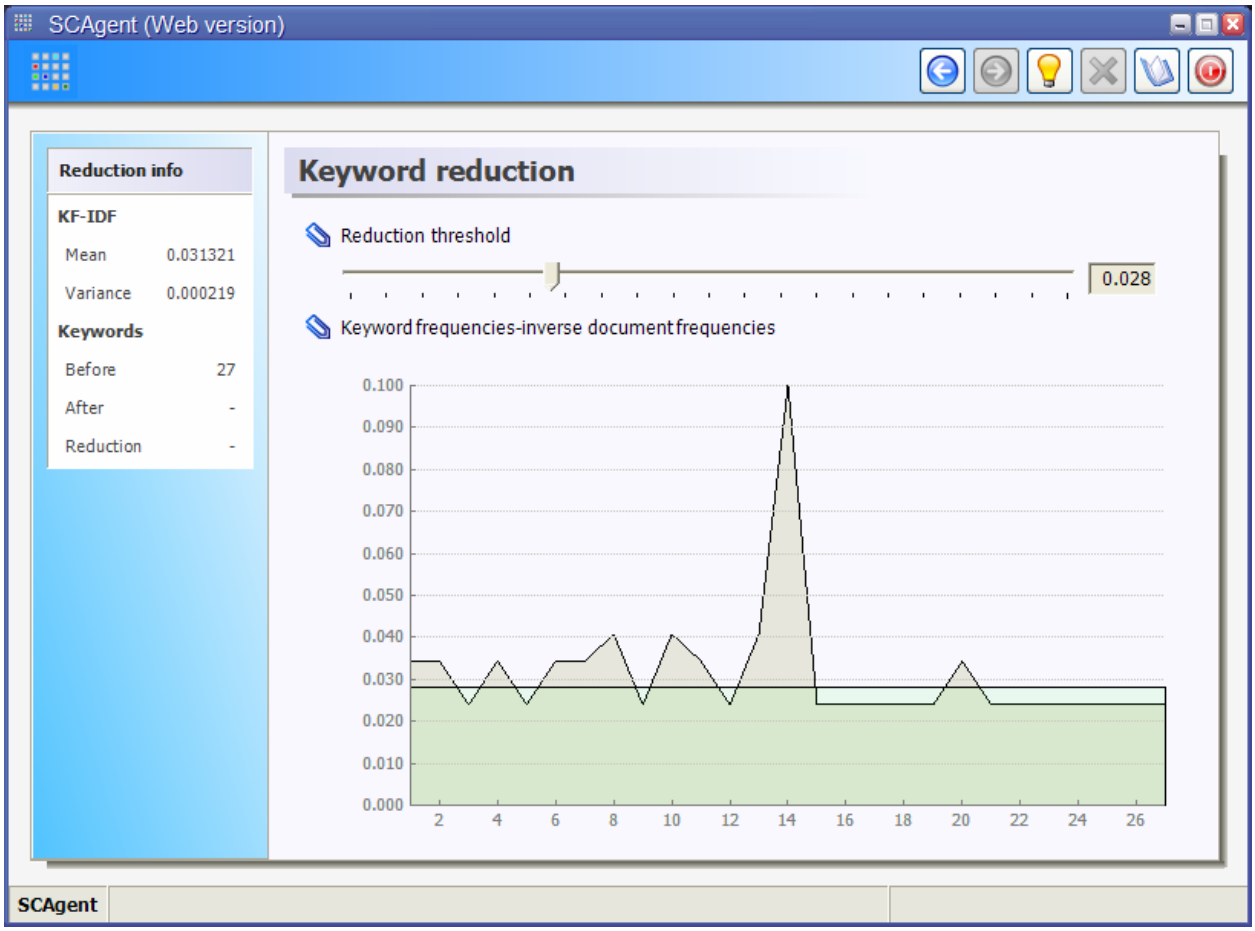
▪ *Όψη επιλογής όρων των αποτελεσμάτων*



Σχήμα 2.6.3: Όψη επιλογής όρων των αποτελεσμάτων

Με την όψη αυτή εμφανίζονται οι όροι που ανιχνεύτηκαν σε όλα τα αποτελέσματα και παρέχεται η δυνατότητα επιλογής αυτών που θα κρατηθούν ως σημαντικοί για την δημιουργία των διανυσματικών αναπαραστάσεων των εγγράφων. Ως αρχική κατάσταση προτείνεται από την εφαρμογή η συμπερίληψη όλων των όρων που ανιχνεύτηκαν. Εφόσον πραγματοποιηθεί η επιλογή των όρων που ο χρήστης κρίνει ως σημαντικούς για την επίτευξη καλύτερης ομαδοποίησης, με το πλήκτρο 'Process' εκτελείται η επεξεργασία για την δημιουργία των σχετικών διανυσματικών αναπαραστάσεων των εγγράφων που προέκυψαν από την αναζήτηση.

▪ *Όψη μείωσης της διάστασης του διανυσματικού μοντέλου αναπαράστασης εγγράφων*

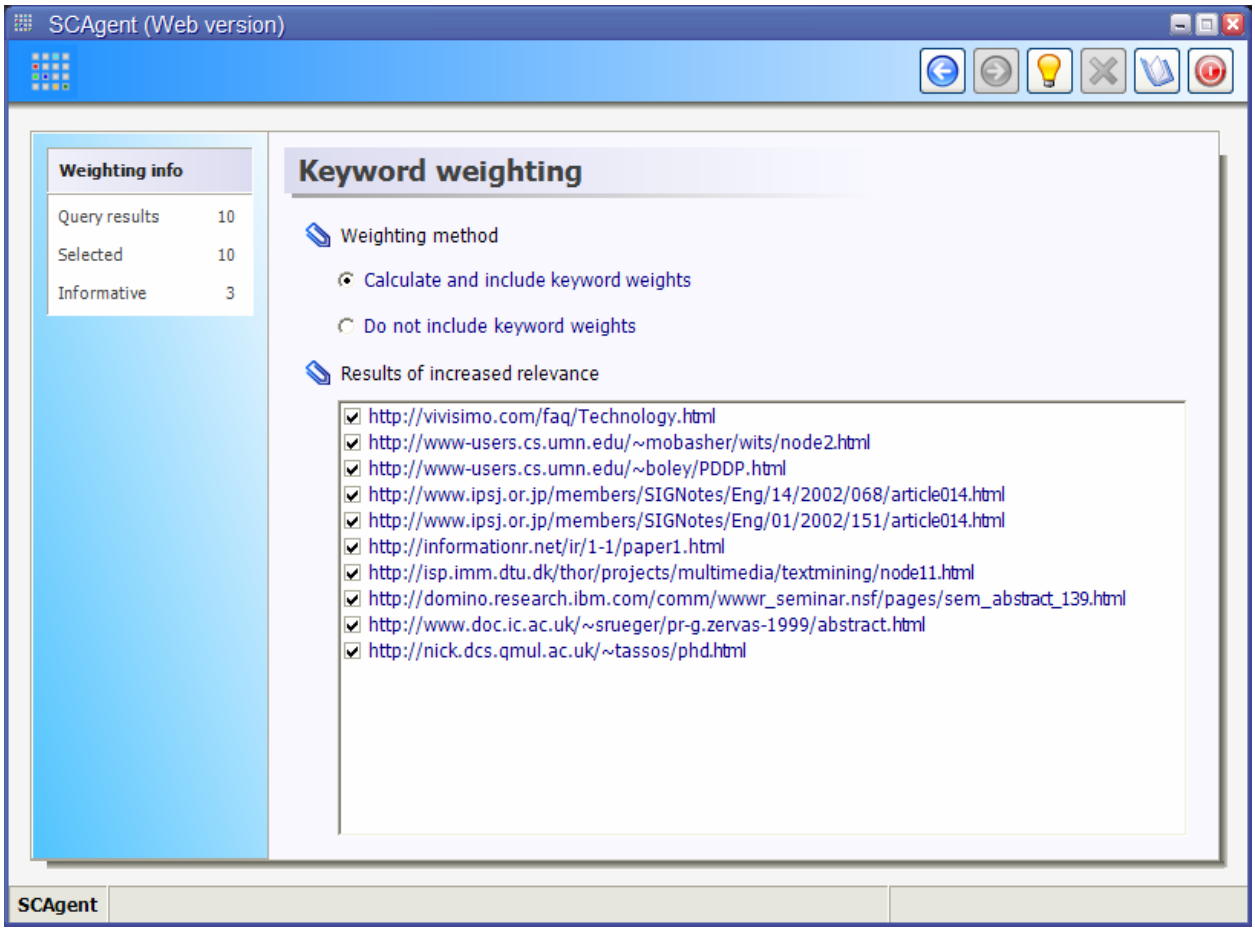


Σχήμα 2.6.4: Όψη μείωσης της διάστασης του διανυσματικού μοντέλου αναπαράστασης εγγράφων

Η όψη αυτή χρησιμοποιείται για τον καθορισμό του κατωφλίου αποκοπής όρων και κατά συνέπεια μείωσης της διάστασης του διανυσματικού μοντέλου αναπαράστασης εγγράφων. Ο χώρος εργασίας περιλαμβάνει κατά πρώτο λόγο ένα πεδίο για την εισαγωγή τιμής για το κατώφλι αποκοπής και κατά δεύτερο λόγο ένα γράφημα που απεικονίζει την κατανομή των σχετικών συχνοτήτων των όρων στο σύνολο των εγγράφων που αποτελούν τα δεδομένα εισόδου. Το γράφημα χρησιμοποιείται για την παροχή μιας ενδεικτικής εποπτικής εικόνας στον

χρήστη που τον βοηθά στην επιλογή της τιμής κατωφλίου που επιθυμεί να εφαρμόσει. Μέσα στα ίδια πλαίσια, στον χώρο πληροφοριών στο αριστερό τμήμα του χώρου εργασίας δίνονται οι τιμές για την μέση συχνότητα και την τυπική απόκλιση από την μέση τιμή για τις συχνότητες όρων όπως προέκυψαν από την επεξεργασία των δεδομένων εισόδου στο προηγούμενο βήμα. Εισαγωγή μηδενικής τιμής κατωφλίου ισοδυναμεί με αποδοχή όλων των όρων χωρίς να πραγματοποιηθεί μείωση στην περίπτωση που αυτό είναι επιθυμητό. Μετά τον καθορισμό του κατωφλίου και εκτέλεση της σχετικής επεξεργασίας, ενεργοποιείται το πλήκτρο 'Next' που καθιστά δυνατή την μετάβαση στο επόμενο βήμα. Σημειώνεται επίσης ότι είναι πάντοτε διαθέσιμη η επιλογή μετάβασης στο προηγούμενο βήμα για ενδεχόμενη νέα επιλογή σημαντικών όρων ή ακόμη και διεξαγωγή νέας έρευνας. Σε αυτή την περίπτωση, τα αποτελέσματα επεξεργασίας για μείωση όρων που είχε ενδεχομένως πραγματοποιηθεί, ακυρώνονται και απαιτείται νέα επεξεργασία για την λήψη ενημερωμένων αποτελεσμάτων.

▪ *Όψη προαιρετικού υπολογισμού διανύσματος στάθμισης των όρων του μοντέλου αναπαράστασης*

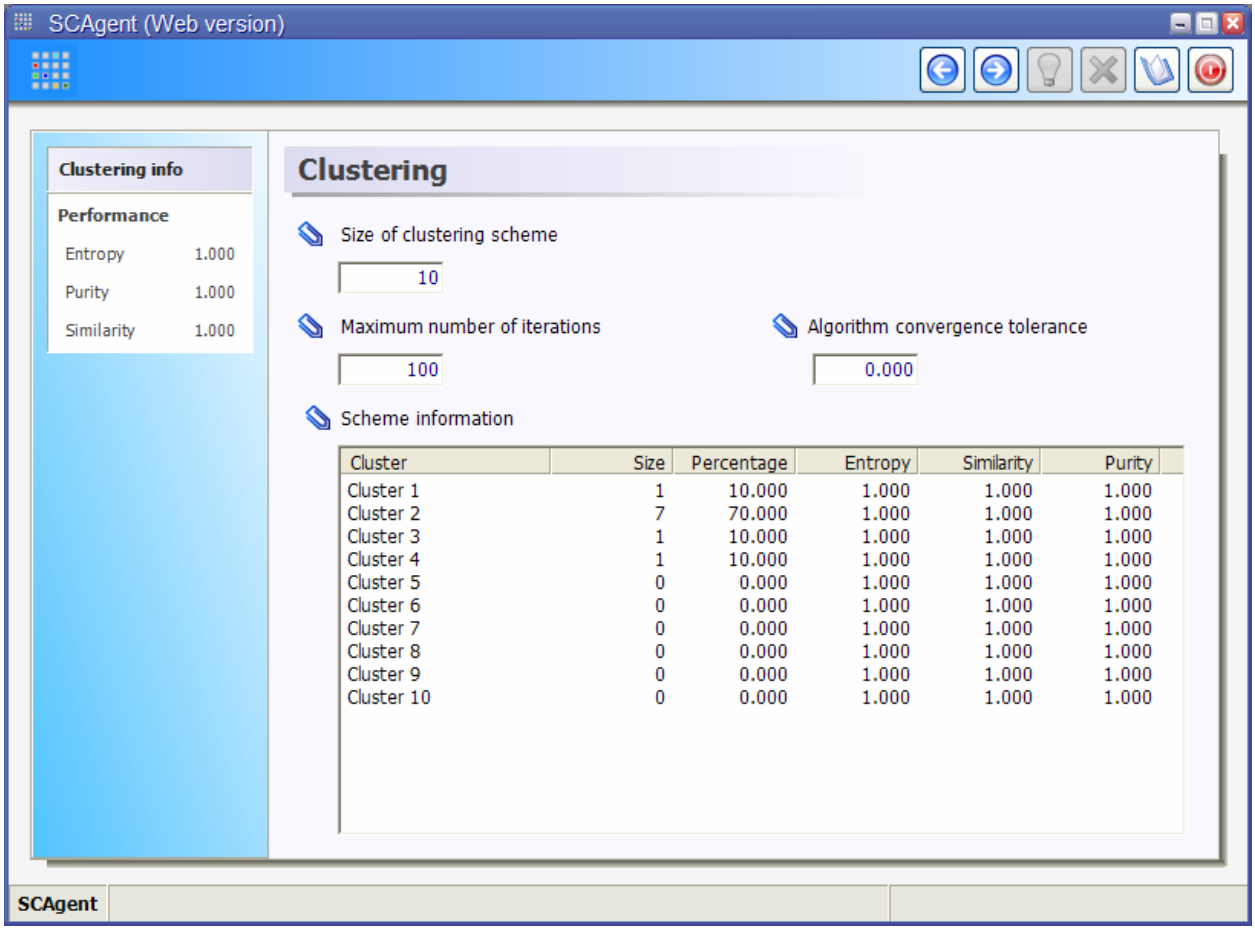


Σχήμα 2.6.5: Όψη προαιρετικού υπολογισμού διανύσματος στάθμισης

Μέσω της συγκεκριμένης όψης είναι δυνατός ο προσδιορισμός βαρών στάθμισης για κάθε όρο του μοντέλου αναπαράστασης εγγράφων μετά από επεξεργασία των εγγράφων.

Συγκεκριμένα, στην επεξεργασία λαμβάνονται υπόψη μόνο τα έγγραφα που ορίζονται από τον χρήστη ως χαρακτηριστικά του θεματικού ενδιαφέροντος του και με υπολογισμό των μέσων συχνοτήτων των όρων στα έγγραφα αυτά εξάγεται ένα διάνυσμα στάθμισης για την ενίσχυση των όρων εκείνων που εμφανίζονται να έχουν διακριτική ικανότητα για το συγκεκριμένο θέμα στο οποίο αναφέρονται τα έγγραφα. Με την επιλογή της μη συμπερίληψης βαρών για τους όρους είναι δυνατή η παράκαμψη του υπολογισμού του διανύσματος στάθμισης και κατά συνέπεια η εκτέλεση της εκδοχής του αλγόριθμου ομαδοποίησης χωρίς στάθμιση.

▪ *Όψη εκτέλεσης αλγόριθμου ομαδοποίησης*



Σχήμα 2.6.6: Όψη εκτέλεσης αλγόριθμου ομαδοποίησης

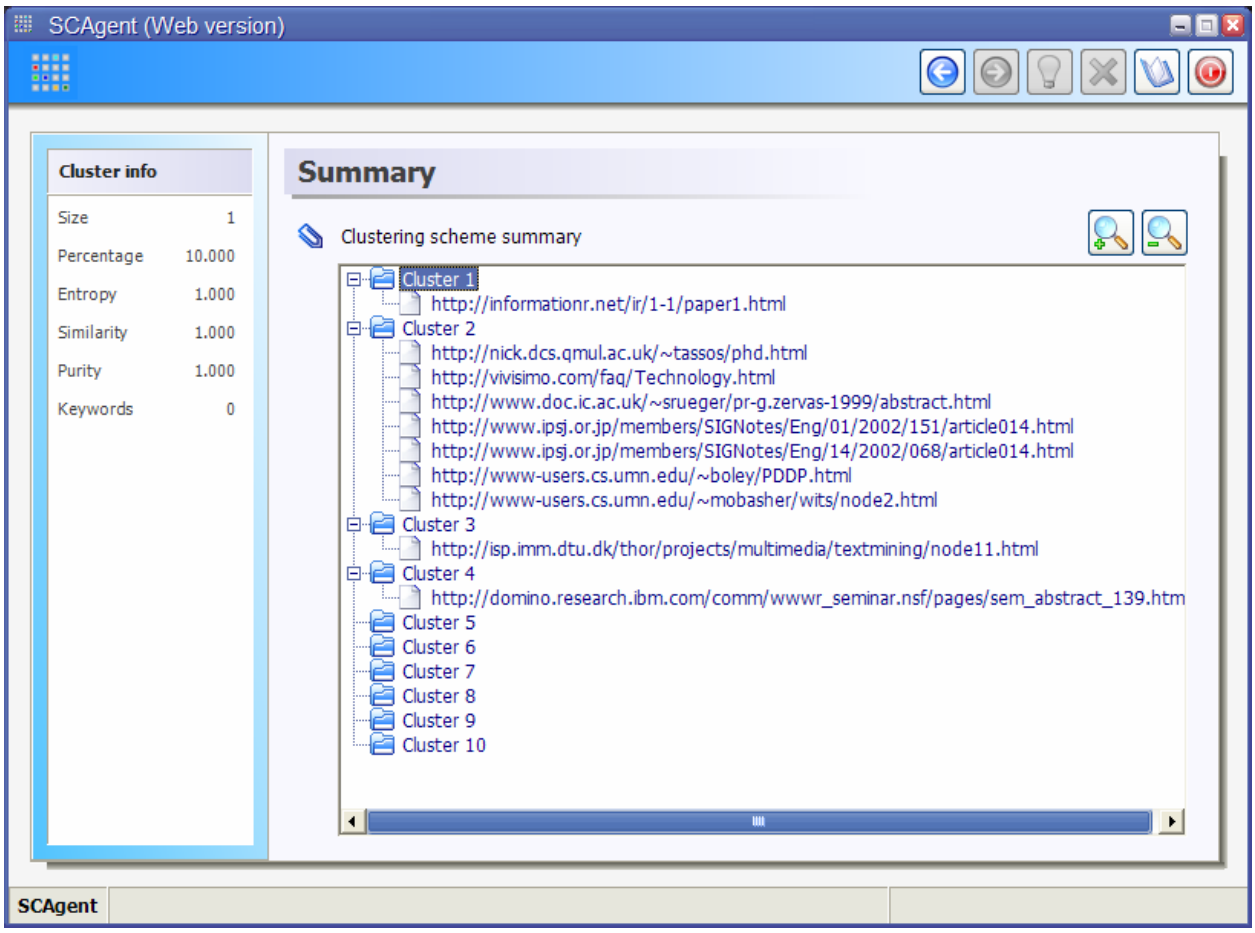
Η συγκεκριμένη όψη χρησιμοποιείται για τον καθορισμό των αναγκαίων παραμέτρων και την εκτέλεση του αλγόριθμου ομαδοποίησης K-Means που υλοποιεί η εφαρμογή. Συγκεκριμένα, στην όψη αυτή ο χρήστης έχει την δυνατότητα να ορίσει το μέγεθος του σχήματος ομαδοποίησης, το μέγιστο πλήθος επαναλήψεων εκτέλεσης του αλγόριθμου καθώς και το όριο σύγκλισης του τελευταίου. Οι διάφοροι συνδυασμοί των πιο πάνω παραμέτρων είναι δυνατό να οδηγήσουν σε πολύ διαφορετικά μεταξύ τους σχήματα ομαδοποίησης. Ενδεικτικά, ένα σχήμα

ομαδοποίησης μικρού μεγέθους οδηγεί σε εξαντλητική εκτέλεση όλων των καθορισμένων επαναλήψεων εκτέλεσης του αλγόριθμου κατάτμησης και καταλήγει εν γένει σε κάποιο μη ιδανικό σχήμα κατάτμησης. Ανάλογα, ο ορισμός μεγάλης τιμής ορίου σύγκλισης του αλγόριθμου δεν εξαντλεί τις καθορισμένες επαναλήψεις εκτέλεσης και επίσης καταλήγει εν γένει σε κάποιο μη αποδοτικό σχήμα ομαδοποίησης. Γενικά, η ρύθμιση των παραμέτρων εκτέλεσης του αλγόριθμου ομαδοποίησης στην συγκεκριμένη όψη εξαρτάται στενά τόσο από την ποιότητα των δεδομένων εισόδου, όσο και από τον συνδυασμό των τιμών των παραμέτρων που ορίζονται.

Κατά την διάρκεια εκτέλεσης του αλγόριθμου ομαδοποίησης, στον χώρο πληροφοριών εμφανίζονται στοιχεία που αφορούν στην πρόοδο εκτέλεσης του αλγόριθμου ενώ στον χώρο εργασίας παρουσιάζεται ένας κατάλογος με τα ζητούμενα clusters που προσδιορίζει η εφαρμογή μαζί με ορισμένες μετρικές πληροφορίες για το κάθε ένα. Με το πέρας εκτέλεσης του αλγόριθμου, ο κατάλογος αυτός απεικονίζει την τελική μορφή του σχήματος ομαδοποίησης και στον χώρο πληροφοριών εμφανίζονται οι τιμές διαφόρων μέτρων επίδοσης για το σχήμα ομαδοποίησης που προέκυψε. Η αλλαγή της τιμής οποιασδήποτε παραμέτρου οδηγεί σε ακύρωση του αποτελέσματος ομαδοποίησης που ενδεχομένως έχει ήδη προσδιοριστεί με βάση τις παλιές τιμές και δίνεται η δυνατότητα επανεκτέλεσης του αλγόριθμου ομαδοποίησης ούτως ώστε να προσδιοριστεί ένα καινούριο σχήμα ομαδοποίησης το οποίο να συμφωνεί με τις νέες τιμές των παραμέτρων.

▪ *Όψη εμφάνισης σχήματος ομαδοποίησης*

Στην όψη αυτή παρουσιάζεται μια άποψη του σχήματος ομαδοποίησης όπως αυτό έχει προσδιοριστεί στο προηγούμενο βήμα, με σκοπό την εμφάνιση κάποιων αναλυτικότερων λεπτομερειών που αφορούν στα επιμέρους clusters. Στην όψη γίνεται χρήση της δενδρικής απεικόνισης (*tree-view*) στην οποία κάθε ομάδα παρουσιάζεται ως κύριος κόμβος και τα έγγραφα που έχουν ανατεθεί σε αυτή ως τα αντίστοιχα φύλλα. Η επιλογή κάποιου κόμβου που αντιστοιχεί σε ομάδα προκαλεί την εμφάνιση στον χώρο πληροφοριών δεδομένων που αφορούν στην συγκεκριμένη ομάδα. Με χρήση του ποντικιού είναι δυνατή η εμφάνιση ενός πτυσσόμενου μενού μέσω του οποίου γίνεται η μετάβαση σε άλλες όμοιες ομάδες. Ανάλογα, η χρήση του ποντικιού σε φύλλο που αντιστοιχεί σε κάποιο αποτέλεσμα δίνει την δυνατότητα επισκόπησης του σχετικού εγγράφου σε ένα καινούριο παράθυρο.



Σχήμα 2.6.7: Όψη εμφάνισης σχήματος ομαδοποίησης

2.7 Μελλοντικές επεκτάσεις

Η υλοποίηση της εφαρμογής SCAgent έγινε κατά τέτοιο τρόπο έτσι ώστε να εφαρμόζονται οι σύγχρονες αρχές του αντικειμενοστραφούς προγραμματισμού στο μεγαλύτερο δυνατό βαθμό. Η εφαρμογή τέτοιων τεχνικών κατέστη δυνατή από τις παρεχόμενες δυνατότητες της πλατφόρμας ανάπτυξης, η οποία κατά τον χρόνο συγγραφής του πηγαίου κώδικα ενσωμάτωνε την πλέον σύγχρονη προγραμματιστική τεχνολογία. Για τον λόγο αυτό, η μελλοντική επέκταση της εφαρμογής κρίνεται εφικτή χωρίς να απαιτούνται ριζικές τροποποιήσεις μεγάλης κλίμακας.

Στη συνέχεια, δίνονται μερικές ενδεικτικές κατευθύνσεις επεκτάσεων που μπορούν να πραγματοποιηθούν για την περαιτέρω εκλέπτυνση των λειτουργιών του SCAgent:

▪ Τροποποίηση της μεθόδου εκτέλεσης αναζήτησης

Η αναζήτηση αποτελεσμάτων που ικανοποιούν τα κριτήρια του χρήστη θα ήταν δυνατό να εκτελείται σε δύο φάσεις. Κατά την πρώτη φάση, η αναζήτηση μπορεί να πραγματοποιείται ανάμεσα στα περιεχόμενα μιας τοπικής βάσης δεδομένων η οποία εκτελεί τον ρόλο μιας μνήμης cache και παρέχει αυξημένη ταχύτητα εκτέλεσης. Προαιρετικά και εφόσον τα αποτελέσματα της

αναζήτησης δεν είναι ικανοποιητικά τα αποτελέσματα μπορούν να εμπλουτίζονται με την κλασσική αναζήτηση στο Web μέσω κάποιας παραδοσιακής μηχανής αναζήτησης.

▪ Ανάλυση περιεχομένου εγγράφων

Η μέθοδος ανάλυσης του περιεχόμενου των εγγράφων-αποτελεσμάτων θα μπορούσε να γίνει πιο σύνθετη έτσι ώστε η αναπαράσταση κάθε εγγράφου από απόψεως θεματικού περιεχομένου να γίνεται κατά τον αποδοτικότερο τρόπο. Ιδιαίτερα κρίσιμης σημασίας για την αποδοτική εφαρμογή οποιουδήποτε αλγόριθμου ομαδοποίησης είναι η όσο το δυνατό καλύτερη αναπαράσταση των εγγράφων και προς αυτή την κατεύθυνση θα μπορούσαν να γίνουν οι κατάλληλες τροποποιήσεις έτσι ώστε η διαδικασία δημιουργίας των αναπαραστάσεων των εγγράφων να εφαρμόζει σύγχρονες, εξεζητημένες τεχνικές ανάλυσης και επεξεργασίας κειμένου.

▪ Ενσωμάτωση εναλλακτικών μεθοδολογιών

Η ενσωμάτωση εναλλακτικών μεθόδων επεξεργασίας για την κατασκευή των αναπαραστάσεων των εγγράφων, την μείωση των όρων του μοντέλου αναπαράστασης, την στάθμιση των όρων αλλά και την οπτική αναπαράσταση της λύσης ομαδοποίησης σαφώς είναι δυνατόν να προσφέρει μεγαλύτερη ευελιξία στην χρήση της εφαρμογής και την δυνατότητα πολυμορφικής αντιμετώπισης του προβλήματος της ομαδοποίησης τόσο σε θεωρητικό-πειραματικό όσο και σε πρακτικό επίπεδο.

3

Πειραματική μελέτη

3.1 Εισαγωγή

Στο παρόν κεφάλαιο περιγράφεται η διαδικασία που ακολουθήθηκε για την αξιολόγηση της επίδοσης του αλγόριθμου K-Means κατά την ομαδοποίηση ενός πειραματικού συνόλου εγγράφων. Στις παραγράφους που ακολουθούν περιγράφονται αναλυτικά τα δεδομένα που χρησιμοποιήθηκαν, η πειραματική μεθοδολογία και τέλος τα αποτελέσματα που προέκυψαν από την όλη διαδικασία.

3.2 Συλλογή εγγράφων

Στα πειράματα που διεξήχθησαν χρησιμοποιήθηκε ένα σύνολο 6,702 εγγράφων τα οποία ανακτήθηκαν από το Διαδίκτυο και πιο συγκεκριμένα από την δικτυακή σελίδα του γνωστού πρακτορείου ειδήσεων Reuters. Για το κάθε έγγραφο χρησιμοποιήθηκε μια διανυσματική αναπαράσταση των συχνοτήτων των όρων που εντοπίστηκαν σε αυτό με το σχετικό μοντέλο να έχει διάσταση 12,834 όρους. Από το σύνολο των εγγράφων δημιουργήθηκαν επιπλέον τέσσερα υποσύνολα που αποτελούνταν από 447, 671, 1,341 και 2,234 έγγραφα αντίστοιχα με ομοιόμορφη επιλογή από το αρχικό σύνολο και με στόχο την μελέτη της επίδοσης του αλγόριθμου ομαδοποίησης K-Means σε συλλογές εγγράφων διαφόρων μεγεθών. Σε όλα τα έγγραφα προηγήθηκε επεξεργασία με στόχο την αφαίρεση όρων (λέξεων) που είναι πολύ συνηθισμένοι και κατά συνέπεια δεν συνεισφέρουν στην αποδοτική λεξική ανάλυση ενός εγγράφου. Επιπλέον, για αρκετά έγγραφα ήταν διαθέσιμη η πληροφορία που αφορούσε στις θεματικές κατηγορίες στις οποίες ανήκαν και αυτή η πληροφορία χρησιμοποιήθηκε μετά την εκτέλεση του αλγόριθμου για την αξιολόγηση του προκύπτοντος σχήματος ομαδοποίησης. Το σύνολο των διαφορετικών θεματικών κατηγοριών στις οποίες ήταν κατανεμημένα τα έγγραφα ήταν 10 και σε αρκετές περιπτώσεις οι θεματικές κατηγορίες ενός εγγράφου ήταν περισσότερες της μίας.

3.3 Πειραματική μεθοδολογία

Για κάθε ένα από τα σύνολα εγγράφων που αναφέρθηκαν στην προηγούμενη παράγραφο, κατασκευάστηκαν σχήματα ομαδοποίησης μεγέθους 5, 10, 25, 50 και 75 ομάδων. Η επιλογή των συγκεκριμένων μεγεθών κρίθηκε αναγκαία για την εξέταση των περιπτώσεων στις οποίες η ομαδοποίηση γίνεται σε μικρότερο, ίσο ή μεγαλύτερο αριθμό ομάδων σε σχέση με το πλήθος

των θεματικών κατηγοριών που ανιχνεύτηκαν στο σύνολο των εγγράφων. Ιδιαίτερα για τις δύο τελευταίες περιπτώσεις, το πλήθος των ομάδων του σχήματος ομαδοποίησης επιλέγηκε αρκετά μεγαλύτερο του πραγματικού πλήθους των θεματικών κατηγοριών καθώς η ύπαρξη αρκετών εγγράφων που παρουσίαζαν πολλαπλή συμμετοχή σε θεματικές κατηγορίες εμμέσως εισάγει υβριδικές κατηγορίες εγγράφων και ανεβάζει το πλήθος τους σε περισσότερες από 10, καθιστώντας έτσι ενδιαφέρουσα (αν όχι αναγκαία) την εξέταση της ποιότητας του σχήματος ομαδοποίησης όταν αυτό κατασκευάζεται με μεγαλύτερο πλήθος θεματικών κατηγοριών σε σχέση με αυτές που ανιχνεύτηκαν στα δεδομένα εισόδου.

Επιπλέον, τα πιο πάνω μεγέθη σχημάτων ομαδοποίησης δοκιμάστηκαν με διάφορες ρυθμίσεις όσον αφορά στην μείωση της διάστασης του διανυσματικού μοντέλου αναπαράστασης των εγγράφων. Συγκεκριμένα, για κάθε όρο υπολογίστηκε η μέση συχνότητα εμφάνισης σε κάθε έγγραφο ως προς την αντίστροφη συχνότητα εμφάνισης του όρου στο σύνολο των εγγράφων (*term frequency - inverse document frequency, TD-IDF*) με στόχο την αποδυνάμωση των όρων που εμφανίζονταν σε μεγάλο πλήθος εγγράφων και συνεπώς είχαν μειωμένη διακριτική ικανότητα και αντίστροφα. Στις πιο πάνω συχνότητες έγινε επεξεργασία με εφαρμογή τιμών κατωφλίου 0.003, 0.004 και 0.005 με αποτέλεσμα την μείωση της διάστασης του διανυσματικού μοντέλου και συμπερίληψη μικρότερου αριθμού όρων στην περαιτέρω διαδικασία σε επίπεδα της τάξης του 98%, 98.5% και 99% αντίστοιχα.

Τέλος, όλες οι σειρές πειραμάτων εκτελέστηκαν δύο φορές χρησιμοποιώντας τόσο την απλή εκδοχή του αλγόριθμου ομαδοποίησης K-Means όσο και την εκδοχή με χρήση βαρών για την στάθμιση των όρων του διανυσματικού μοντέλου αναπαράστασης των εγγράφων. Ο υπολογισμός των βαρών στάθμισης των όρων έγινε με χρήση της πληροφορίας που ήταν διαθέσιμη σχετικά με τις θεματικές κατηγορίες των εγγράφων. Συγκεκριμένα, για τις *m* θεματικές κατηγορίες υπολογίστηκαν τα χαρακτηριστικά-κεντροειδή διανύσματα $\left\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_m\right\}$ με βάση τα γνωστά έγγραφα για την κάθε κατηγορία. Στην συνέχεια, για κάθε όρο *i* κατασκευάστηκε το διάνυσμα $\vec{T}_i = \left\{C_{1,i}, C_{2,i}, ..., C_{m,i}\right\}$ το οποίο αναπαριστά τις συχνότητες εμφάνισης του συγκεκριμένου όρου στο σύνολο των κεντροειδών διανυσμάτων όλων των θεματικών κατηγοριών. Από την κανονικοποιημένη μορφή αυτών των διανυσμάτων $\vec{T}_i' = \vec{T}_i / \left\|\vec{T}_i\right\|_1$ το τελικό βάρος κάθε όρου υπολογίστηκε ως $P_i = \sum_{j=1}^{m} T_{j,i}'^2$, δηλαδή το τετράγωνο του μήκους του αντίστοιχου διανύσματος $\vec{T}_i'$ κάθε όρου, με την τιμή P_i να κυμαίνεται πάντοτε στο διάστημα $\left[1/m, 1\right]$. Η χαμηλότερη τιμή *1/m* προκύπτει όταν $T_{i,1}' = T_{i,2}' = ... = T_{i,m}'$, δηλαδή όταν οι συχνότητες του όρου *i* είναι ίσες σε όλα τα κεντροειδή διανύσματα $\left\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_m\right\}$.

Κατά αναλογία, η ψηλότερη τιμή προκύπτει όταν ο όρος έχει πεπερασμένη συχνότητα εμφάνισης σε ακριβώς ένα κεντροειδές διάνυσμα και μηδενική συχνότητα σε όλα τα υπόλοιπα. Συνεπώς, οι τιμές P_i χαρακτηρίζουν την διακριτική ικανότητα του κάθε όρου και το διάνυσμα $\vec{P} = \{P_1, P_2, ..., P_n\}$ που προέκυψε από την σχετική επεξεργασία χρησιμοποιήθηκε στην σταθμισμένη εκδοχή του αλγόριθμου ομαδοποίησης [KS].

Συνολικά, ο αριθμός των πειραμάτων που απαιτήθηκαν για την κάλυψη των συνδυασμών των πιο πάνω παραμέτρων ανήλθε στα 30 πειράματα (5 x 3 x 2). Κάθε πείραμα εκτελέστηκε 3 φορές για την αποφυγή περιπτώσεων στις οποίες ο αλγόριθμος ομαδοποίησης κατέληγε σε λύση η οποία δεν ήταν η βέλτιστη δυνατή αλλά ένα τοπικό ελάχιστο του κριτηρίου που χρησιμοποιήθηκε για την αξιολόγηση της επίδοσης του. Οι τιμές των μέτρων αξιολόγησης που σημειώνονται στις επόμενες παραγράφους αντιστοιχούν στις μέσες τιμές που παρατηρήθηκαν κατά τις μετρήσεις αυτές. Επιπλέον των πιο πάνω πειραμάτων, ελέγχθηκε η επίδοση της σταθμισμένης εκδοχής του αλγόριθμου ομαδοποίησης στην περίπτωση που ο υπολογισμός του διανύσματος στάθμισης $\vec{P}$ γίνεται λαμβάνοντας υπόψη μεταβλητό πλήθος προτύπων. Για αυτή την σειρά πειραμάτων χρησιμοποιήθηκε ένα σύνολο σταθερών τιμών όσον αφορά στις υπόλοιπες παραμέτρους που περιγράφηκαν στα προηγούμενα.

3.4 Μέτρα αξιολόγησης

Η ποιότητα κάθε λύσης αξιολογήθηκε με την χρήση τριών διαφορετικών κριτηρίων αξιολόγησης τα οποία χρησιμοποιούσαν την πληροφορία που αφορούσε στις θεματικές κατηγορίες των εγγράφων που τοποθετήθηκαν σε κάθε ομάδα [KZ02].

Το πρώτο κριτήριο ήταν το μέτρο εντροπίας (*entropy measure*) το οποίο αξιολογεί την κατανομή των διαφορετικών θεματικών κατηγοριών των εγγράφων μέσα στα πλαίσια κάθε ομάδας. Για μια δεδομένη ομάδα S_r του σχήματος ομαδοποίησης που περιέχει n_r έγγραφα, το μέτρο εντροπίας της ομάδας ορίζεται ως

$$E(S_r) = -\frac{1}{\log q} \cdot \sum_{i=1}^{q} \frac{n_r^i}{n_r} \cdot \log \frac{n_r^i}{n_r}$$

όπου k είναι το πλήθος των διαφορετικών θεματικών κατηγοριών στο σύνολο των εγγράφων και n^i είναι το πλήθος των εγγράφων της θεματικής κατηγορίας i που ανατέθηκαν στην ομάδα S_r. Το μέτρο εντροπίας του σχήματος ομαδοποίησης στο σύνολο του υπολογίζεται ως το σταθμισμένο άθροισμα των μέτρων εντροπίας κάθε επιμέρους ομάδας με βάρος ανάλογο του μεγέθους της ομάδας ως προς το σύνολο των εγγράφων, δηλαδή

$$\textit{Εντροπία σχήματος ομαδοποίησης} = \sum_{r=1}^{k} \frac{n_r}{n} \cdot E\left(S_r\right)$$

Είναι προφανές ότι ένα ιδανικό σχήμα ομαδοποίησης αποτελείται από ομάδες οι οποίες περιέχουν έγγραφα ακριβώς μιας θεματικής κατηγορίας και σε αυτή την περίπτωση τόσο το μέτρο εντροπίας της κάθε ομάδας όσο και ολόκληρου του σχήματος ομαδοποίησης θα είναι μηδενικό. Γενικά, ένα καλό σχήμα ομαδοποίησης χαρακτηρίζεται από χαμηλές τιμές του μέτρου εντροπίας.

Το δεύτερο κριτήριο ήταν το μέτρο εσωτερικής ομοιότητας (*internal similarity measure*) το οποίο προσδιορίζει τον μέσο βαθμό ομοιότητας των εγγράφων που περιέχονται σε κάθε ομάδα. Για μια δεδομένη ομάδα S_r που περιέχει n_r έγγραφα το μέτρο εσωτερικής ομοιότητας της ομάδας ορίζεται ως

$$IS(S_r) = \frac{1}{\binom{n_r}{2}} \cdot \sum_{i \neq j} d_r^i \cdot d_r^j$$

όπου d^i και d_r^j είναι έγγραφα της συγκεκριμένης ομάδας. Κατά αναλογία ως προς το μέτρο εντροπίας, το μέτρο εσωτερικής ομοιότητας ολόκληρου του σχήματος ομαδοποίησης υπολογίζεται ως το σταθμισμένο άθροισμα των μέτρων ομοιότητας κάθε επιμέρους ομάδας με βάρος ανάλογο του μεγέθους της ομάδας ως προς το σύνολο των εγγράφων.

$$\textit{Εσωτερική ομοιότητα ομάδων σχήματος ομαδοποίησης} = \sum_{r=1}^{k} \frac{n_r}{n} \cdot IS\left(S_r\right)$$

Γενικά, ένα καλό σχήμα ομαδοποίησης χαρακτηρίζεται από μεγάλες τιμές του μέτρου εσωτερικής ομοιότητας.

Το τρίτο κριτήριο ήταν το μέτρο καθαρότητας (*purity measure*) το οποίο προσδιορίζει τον λόγο του μεγέθους της μεγαλύτερης θεματικής κατηγορίας μιας ομάδας ως προς το μέγεθος της τελευταίας. Για μια δεδομένη ομάδα S_r που περιέχει n_r έγγραφα το μέτρο καθαρότητας της ομάδας ορίζεται ως

$$P\left(S_r\right) = \frac{1}{n_r} \cdot \max_i \left(n_r^i\right)$$

όπου n^i είναι το πλήθος των εγγράφων της θεματικής κατηγορίας i που ανατέθηκαν στην ομάδα S_r. Το μέτρο καθαρότητας ολόκληρου του σχήματος ομαδοποίησης υπολογίζεται ως το σταθμισμένο άθροισμα των μέτρων καθαρότητας κάθε ομάδας με βάρος ανάλογο του μεγέθους της ως προς το σύνολο των εγγράφων.

$$\textit{Καθαρότητα σχήματος ομαδοποίησης} = \sum_{r=1}^{k} \frac{n_r}{n} \cdot P\left(S_r\right)$$

Ένα καλό σχήμα ομαδοποίησης χαρακτηρίζεται από ψηλές τιμές του μέτρου καθαρότητας, οι οποίες φανερώνουν τον βαθμό στον οποίο μπορεί να θεωρηθεί ότι οι ομάδες ταυτίζονται με τις θεματικές κατηγορίες.

3.5 Αποτελέσματα

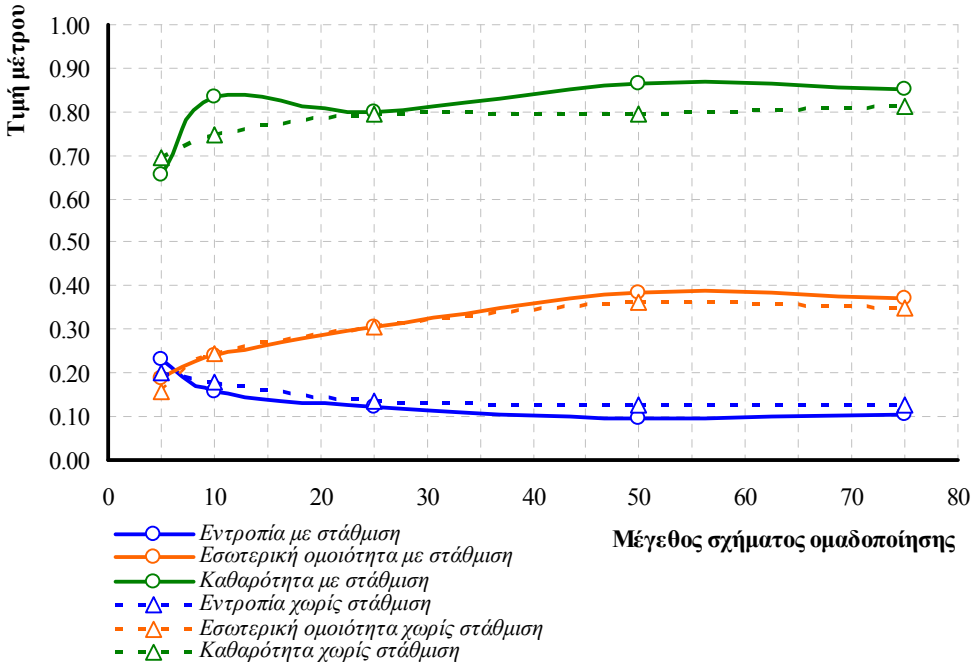
Κατά την εκτέλεση των πειραμάτων που περιγράφηκαν στα προηγούμενα, καταγράφηκαν για κάθε περίπτωση οι μέσες τιμές των μέτρων αξιολόγησης και παρουσιάζονται στον πίνακα που ακολουθεί.

Κατώφλι μείωσης όρων	Χρήση βαρών στάθμισης	Μέγεθος σχήματος ομαδοποίησης	Μέτρο εντροπίας	Μέτρο εσωτερικής ομοιότητας	Μέτρο καθαρότητας
0.003	Ναι	5	0.232	0.186	0.654
0.003	Ναι	10	0.159	0.240	0.834
0.003	Ναι	25	0.121	0.304	0.799
0.003	Ναι	50	0.094	0.386	0.863
0.003	Ναι	75	0.105	0.371	0.850
0.003	Όχι	5	0.200	0.156	0.695
0.003	Όχι	10	0.178	0.244	0.745
0.003	Όχι	25	0.134	0.306	0.793
0.003	Όχι	50	0.125	0.362	0.793
0.003	Όχι	75	0.126	0.350	0.813
0.004	Ναι	5	0.240	0.170	0.662
0.004	Ναι	10	0.188	0.262	0.778
0.004	Ναι	25	0.133	0.389	0.835
0.004	Ναι	50	0.103	0.461	0.836
0.004	Ναι	75	0.098	0.466	0.846
0.004	Όχι	5	0.234	0.169	0.599
0.004	Όχι	10	0.211	0.275	0.649
0.004	Όχι	25	0.156	0.351	0.743
0.004	Όχι	50	0.121	0.401	0.831
0.004	Όχι	75	0.119	0.450	0.803
0.005	Ναι	5	0.251	0.211	0.607
0.005	Ναι	10	0.186	0.299	0.763
0.005	Ναι	25	0.141	0.455	0.812
0.005	Ναι	50	0.105	0.528	0.832
0.005	Ναι	75	0.102	0.532	0.828
0.005	Όχι	5	0.207	0.203	0.662
0.005	Όχι	10	0.189	0.281	0.707
0.005	Όχι	25	0.139	0.456	0.789
0.005	Όχι	50	0.126	0.495	0.806
0.005	Όχι	75	0.131	0.500	0.786

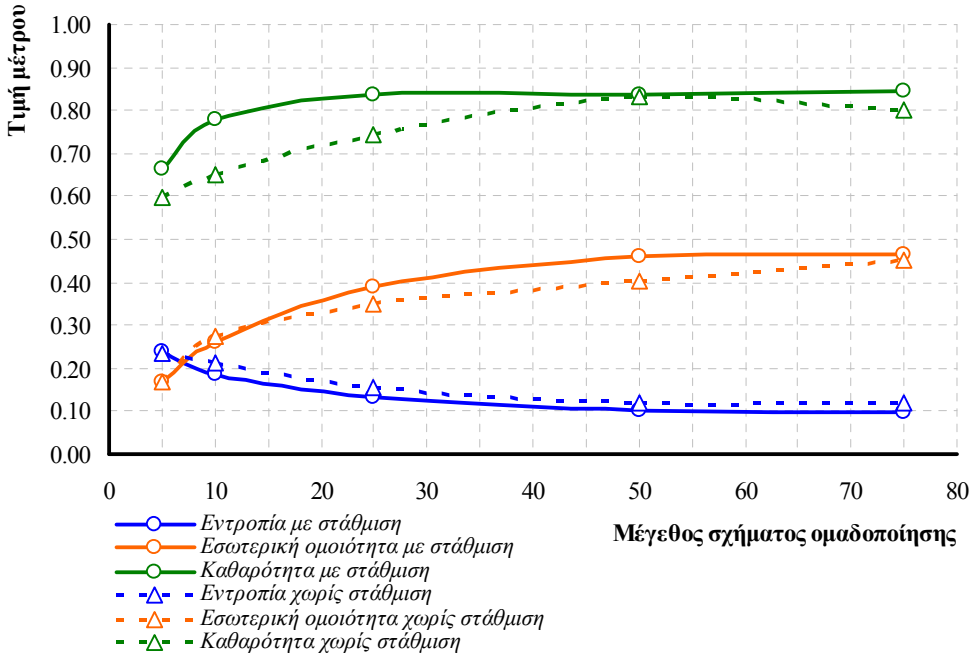
Πίνακας 3.5.1: Πειραματικά αποτελέσματα εκτελέσεων αλγόριθμου ομαδοποίησης. Με μπλε χρώμα σημειώνεται η καλύτερη επίδοση κάθε περίπτωσης και με κόκκινο οι επιδόσεις που διέφεραν μέχρι 5% της καλύτερης.

Για την καλύτερη αναπαράσταση των πιο πάνω επιδόσεων και την ευκολότερη εξαγωγή συμπερασμάτων όσον αφορά στην επίδραση των παραμέτρων που χρησιμοποιήθηκαν στην

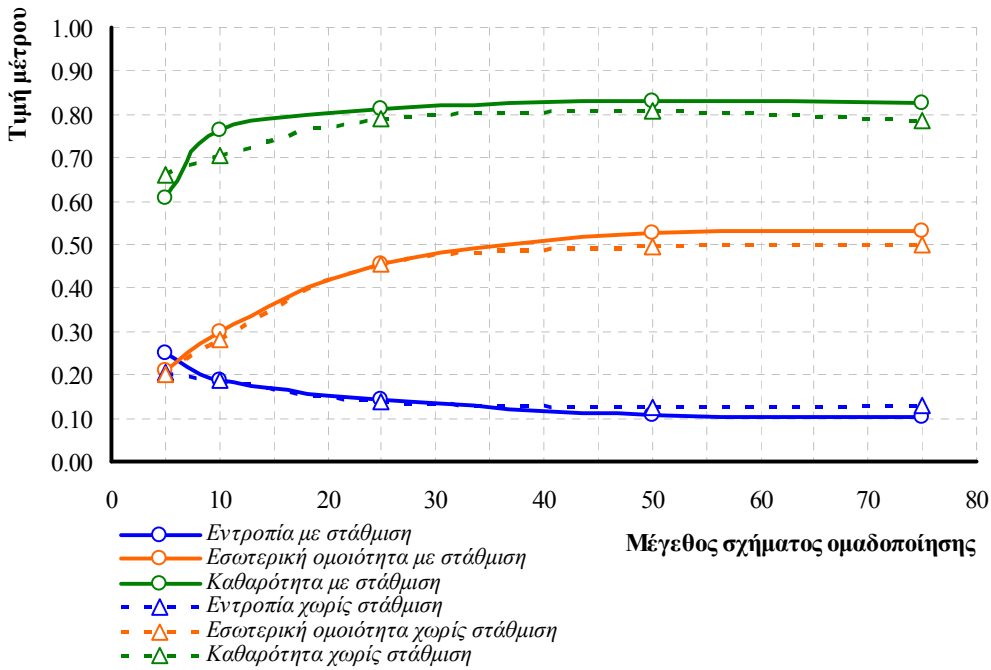
διαμόρφωση των ομάδων εγγράφων, παρατίθενται στην συνέχεια διαγράμματα των μέτρων αξιολόγησης σε σχέση με το μέγεθος του τελικού σχήματος ομαδοποίησης. Τα διαγράμματα που παρουσιάζονται αναφέρονται στις πειραματικές δοκιμές για τις διάφορες τιμές κατωφλίου μείωσης της διάστασης του διανυσματικού μοντέλου αναπαράστασης και για την κάθε περίπτωση σημειώνονται οι τιμές των τριών μέτρων επίδοσης τόσο στην περίπτωση εκτέλεσης



Διάγραμμα 3.5.2: Τιμές μέτρων αξιολόγησης σε σχέση με το μέγεθος του σχήματος ομαδοποίησης στην περίπτωση που εφαρμόζεται κατώφλι μείωσης διάστασης διανυσματικού μοντέλου ίσο με 0.003.



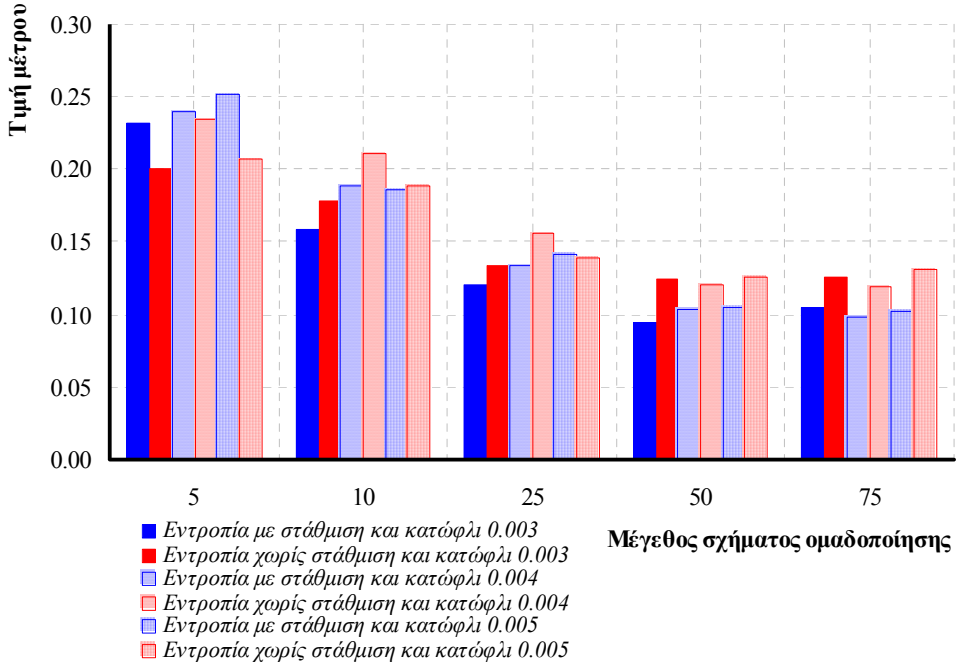
Διάγραμμα 3.5.3: Τιμές μέτρων αξιολόγησης σε σχέση με το μέγεθος του σχήματος ομαδοποίησης στην περίπτωση που εφαρμόζεται κατώφλι μείωσης διάστασης διανυσματικού μοντέλου ίσο με 0.004.



Διάγραμμα 3.5.4: Τιμές μέτρων αξιολόγησης σε σχέση με το μέγεθος του σχήματος ομαδοποίησης στην περίπτωση που εφαρμόζεται κατώφλι μείωσης διάστασης διανυσματικού μοντέλου ίσο με 0.005.

του αλγόριθμου ομαδοποίησης με χρήση βαρών των όρων όσο και στην περίπτωση που δεν χρησιμοποιείται αυτή η στάθμιση.

Στην συνέχεια, παρατίθεται διαγραμματική αναπαράσταση των τιμών των μέτρων εντροπίας για τις παραλλαγές του αλγόριθμου με στάθμιση των όρων και χωρίς και για τις 3 σειρές πειραμάτων που εκτελέστηκαν με τις διάφορες τιμές κατωφλίου αποκοπής όρων.

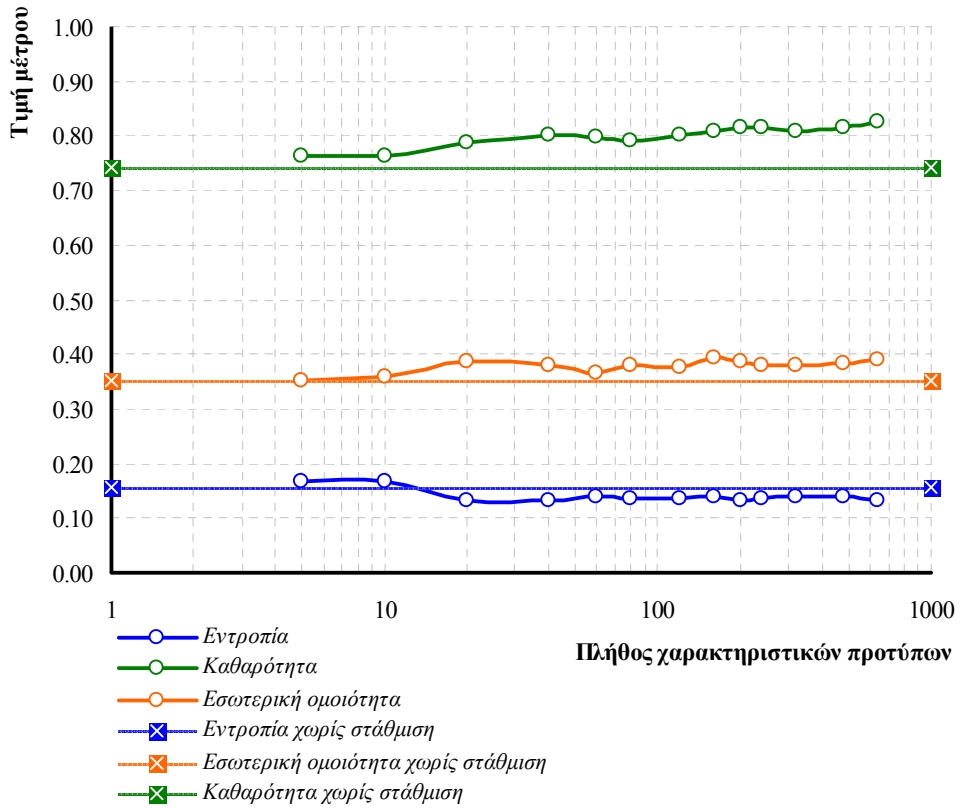


Διάγραμμα 3.5.5: Τιμές μέτρων εντροπίας σε σχέση με τις περιπτώσεις χρήσης στάθμισης ή όχι και για τις διαφορετικές τιμές κατωφλίου μείωσης διάστασης του διανυσματικού μοντέλου.

Τέλος, παρατίθενται οι τιμές που καταγράφηκαν και το σχετικό διάγραμμα των τριών μέτρων επίδοσης κατά την εκτέλεση πειραμάτων με μεταβλητό πλήθος προτύπων για την κατασκευή του διανύσματος στάθμισης των όρων. Οι σταθερές τιμές για τις λοιπές παραμέτρους που χρησιμοποιήθηκαν ήταν 0.004 για το κατώφλι μείωσης της διανυσματικής αναπαράστασης και μέγεθος σχήματος ομαδοποίησης ίσο με 25.

Χρησιμοποιούμενα πρότυπα	Μέτρο εντροπίας	Μέτρο εσωτερικής ομοιότητας	Μέτρο καθαρότητας
5	0.167	0.352	0.762
10	0.167	0.358	0.763
20	0.134	0.386	0.788
40	0.132	0.381	0.802
60	0.141	0.367	0.798
80	0.135	0.380	0.792
120	0.135	0.376	0.801
160	0.140	0.392	0.808
200	0.134	0.387	0.815
240	0.136	0.380	0.817
320	0.140	0.380	0.807
480	0.139	0.384	0.817
640	0.133	0.389	0.826

Πίνακας 3.5.6: Πειραματικά αποτελέσματα εκτελέσεων σταθμισμένης εκδοχής αλγόριθμου ομαδοποίησης με μεταβλητό πλήθος χρησιμοποιούμενων προτύπων για τον προσδιορισμό του διανύσματος στάθμισης.



Διάγραμμα 3.5.7: Τιμές μέτρων επίδοσης σε σχέση με το πλήθος των χρησιμοποιούμενων προτύπων για την κατασκευή του διανύσματος στάθμισης βαρών των όρων.

3.6 Σχολιασμός και ανάλυση αποτελεσμάτων

Η ανάλυση των αποτελεσμάτων που παρουσιάζονται στον πίνακα αλλά και στα διαγράμματα της προηγούμενης παραγράφου, μπορεί να οδηγήσει σε κάποιες σημαντικές διαπιστώσεις.

Αρχικά, σημειώνεται ότι η επιλογή ενσωμάτωσης βαρών στάθμισης των όρων κατά την διαδικασία εκτέλεσης του αλγόριθμου οδηγεί σχεδόν πάντοτε σε καλύτερες επιδόσεις και για τα τρία μέτρα αξιολόγησης που χρησιμοποιήθηκαν για τα συγκεκριμένα πειράματα. Η συμπεριφορά αυτή είναι αναμενόμενη καθώς βασίζεται στην λογική διαπίστωση ότι όλοι οι όροι δεν διαθέτουν την ίδια ή τουλάχιστον παρόμοια διακριτική ικανότητα για τον αποτελεσματικό διαχωρισμό εγγράφων σε θεματικές κατηγορίες. Η γνώση της βαρύτητας που διαθέτει κάθε όρος στην διάκριση μιας θεματικής ενότητας ή η εξαγωγή ενός αρκετά ενδεικτικού μέτρου αυτής της βαρύτητας μέσα από επεξεργασία χαρακτηριστικών για την κατηγορία εγγράφων είναι ίσως ο πρώτος σε σημασία παράγοντας που μπορεί να οδηγήσει σε αρκετά καλά αποτελέσματα ομαδοποίησης. Για την αξιοποίηση αυτής της παραμέτρου είναι βέβαια αναγκαίο να υπάρχει σε κάποιο βαθμό γνώση πληροφορίας που αφορά στις θεματικές κατανομές των εγγράφων και φυσικά η γνώση αυτή να αποτελεί μια αντιπροσωπευτική ένδειξη για το σύνολο των εγγράφων. Συγκεκριμένα, η ύπαρξη πληροφορίας για κάποιο έγγραφο που αποτελεί τυπικό αντιπρόσωπο μιας πλατύτερης θεματικής κατηγορίας είναι καθοριστικής σημασίας για την «έξυπνη» ανίχνευση των χαρακτηριστικών εκείνων που προσφέρουν την δυνατότητα διαχωρισμού της συγκεκριμένης κατηγορίας σε σχέση με τις υπόλοιπες. Η απαίτηση ύπαρξης μιας τέτοιας πληροφορίας είναι βέβαια στην πλειονότητα των περιπτώσεων δύσκολο να ικανοποιηθεί και κάτω από αυτές τις περιστάσεις είναι ενδεχομένως αρκετά χρήσιμη η ανθρώπινη παρέμβαση-συμμετοχή στην όλη διαδικασία της ομαδοποίησης με την παροχή ενδεικτικών κατευθύνσεων που μπορούν να υποκαταστήσουν την έλλειψη της πληροφορίας.

Μια ακόμη σημαντική παρατήρηση που εξάγεται από τα διαγράμματα είναι ότι το μέγεθος του σχήματος ομαδοποίησης καθορίζει δραστικά τα επίπεδα στο οποίο κινούνται τα μέτρα εντροπίας και εσωτερικής ομοιότητας. Η επιλογή του προσδιορισμού ομάδων οι οποίες είναι λιγότερες σε πλήθος από τις κατηγορίες που συνολικά εντοπίζονται στο σύνολο των εγγράφων έχει ως αποτέλεσμα σχετικά κακές ομαδοποιήσεις. Προφανώς, μια τέτοια επιλογή οδηγεί αναπόφευκτα σε συμπερίληψη περισσότερων θεματικών κατηγοριών μέσα στα πλαίσια μιας ενιαίας ομάδας και δεν εξυπηρετεί την επίτευξη του επιθυμητού διαχωρισμού κάθε κατηγορίας. Επιπλέον, ο προσδιορισμός ομάδων πλήθους σαφώς μεγαλύτερου του πλήθους των θεματικών κατηγοριών που πραγματικά υπάρχουν εκ πρώτης όψεως οδηγεί σε σχήματα ομαδοποίησης αρκετά καλύτερων επιδόσεων. Όμως, με προσεκτική εξέταση της δομής των δημιουργούμενων

ομάδων αποκαλύπτεται ότι οι καλύτερες επιδόσεις του σχήματος ομαδοποίησης οφείλονται στα μικρά μεγέθη των ομάδων λόγω της υπερβολικής κατάτμησης κάθε θεματικής κατηγορίας. Στις περιπτώσεις αυτές το σχήμα ομαδοποίησης αποτελείται από πολυπληθείς και μικρές σε μέγεθος ομάδες και ενώ υλοποιείται ο στόχος του διαχωρισμού μεταξύ των θεματικών κατηγοριών εντούτοις ο διαχωρισμός αυτός δεν είναι ο πλέον πρακτικός και χρήσιμος που θα ήταν δυνατό να προκύψει.

Μια άλλη παράμετρος που εξετάστηκε με την σειρά των πειραμάτων που εκτελέστηκαν ήταν ο βαθμός μείωσης της διάστασης του διανυσματικού μοντέλου αναπαράστασης των όρων. Για την συγκεκριμένη παράμετρο δοκιμάστηκαν τρεις μόνο διαφορετικές τιμές οι οποίες για την συγκεκριμένη συλλογή εγγράφων που χρησιμοποιήθηκε στα πειράματα, επέφεραν μειώσεις στην διανυσματική διάσταση της τάξης του 98%, 98.5% και 99% με τους όρους οι οποίοι εμφανίζονταν σε μεγάλο ποσοστό των εγγράφων και κατά συνέπεια είχαν χαμηλό λόγο συχνοτήτων TF-IDF να αποκλείονται από την μετέπειτα επεξεργασία. Οι μειώσεις αυτές, παρόλο που ήταν αρκετά ψηλού επιπέδου, δεν κρίνονται επιβαρυντικές για την αξιοπιστία των αποτελεσμάτων που λήφθηκαν. Αντίθετα, ο αποκλεισμός και μη χρησιμοποίηση ενός μεγάλου μέρους των διαθέσιμων όρων (οι οποίοι αποτελούν τα χαρακτηριστικά γνωρίσματα ταυτοποίησης κάθε εγγράφου) ήταν εν γένει επιθυμητός ούτως ώστε να στην τελική φάση επεξεργασίας να συμμετέχουν μόνο εκείνοι οι όροι που αποδεδειγμένα εμφάνιζαν αυξημένη διακριτική ικανότητα μεταξύ των εγγράφων. Τα πειραματικά αποτελέσματα αποκάλυψαν ότι όσον αφορά στο μέτρο εντροπίας υπάρχει μία οριακή τιμή κατωφλίου μέχρι την οποία η μείωση επιφέρει θετικά αποτελέσματα στην ποιότητα του σχήματος ομαδοποίησης. Περαιτέρω μείωση πέραν αυτής της οριακής τιμής ενδέχεται όμως να αντιστρέψει αυτή την θετική επίδραση και να υποβαθμίσει την ποιότητα του σχήματος. Αυτή η τάση είναι ιδιαίτερα ευδιάκριτη στην περίπτωση που δεν εφαρμόζεται στάθμιση των όρων όπως μπορεί εύκολα να διαπιστωθεί από το τελευταίο διάγραμμα της προηγούμενης παραγράφου. Γενικά, η εφαρμογή μιας τιμής κατωφλίου είναι ένας κρίσιμος παράγοντας ο οποίος είναι δυνατό να επιφέρει δραματικές αλλαγές στην ποιότητα της ομαδοποίησης καθώς εξαρτάται πολύ στενά με το περιεχόμενο των εγγράφων.

Η τελευταία παράμετρος που εξετάστηκε με μια ανεξάρτητη σειρά πειραμάτων ήταν το πλήθος των προτύπων που χρησιμοποιούνται για την κατασκευή του διανύσματος στάθμισης βαρών των όρων. Από τις τιμές των μέτρων επίδοσης που καταγράφηκαν διαπιστώθηκε ότι η συγκεκριμένη παράμετρος επηρεάζει σε σημαντικό βαθμό την επίδοση του προκύπτοντος σχήματος ομαδοποίησης μόνο στις περιπτώσεις που τα χρησιμοποιούμενα πρότυπα είναι λίγα σε πλήθος. Αντίθετα, όταν ο αριθμός των προτύπων αυξηθεί η επίδοση του σχήματος παραμένει σχεδόν σταθερή και αυτή η διαπίστωση φαίνεται πολύ καθαρά στο σχετικό διάγραμμα των

τιμών. Αυτή η συμπεριφορά είναι αναμενόμενη καθώς η χρήση περισσότερων προτύπων κατά την διαδικασία κατασκευής του διανύσματος στάθμισης, εξασφαλίζει την όσο το δυνατό καλύτερη εκτίμηση του κεντροειδούς διανύσματος κάθε θεματικής κατηγορίας. Η κατασκευή κεντροειδών διανυσμάτων τα οποία να αποτυπώνουν με τον πιο παραστατικό τρόπο την βαρύτητα των όρων μέσα στα πλαίσια της κάθε θεματικής κατηγορίας αποτελεί κρίσιμο σημείο αναφοράς για την αποδοτική ανίχνευση μοτίβων που υπάρχουν στα πρότυπα εισόδου. Σύμφωνα και με τα προηγούμενα συμπεράσματα, η επιτυχής ανίχνευση τέτοιων μοτίβων (μεταφρασμένη σε γνώση της διακριτικής ικανότητας του κάθε όρου) αποτελεί παράγοντα βαρύνουσας σημασίας για το προσδιορισμό ενός αποδοτικού σχήματος ομαδοποίησης.

Η τελευταία διαπίστωση τονίζεται ακόμη περισσότερο αν συγκριθεί η επίδοση του σχήματος ομαδοποίησης όταν το διάνυσμα στάθμισης κατασκευάζεται με χρήση λίγων προτύπων σε σχέση με το σχήμα που προκύπτει χωρίς την εφαρμογή στάθμισης. Στα πειραματικά αποτελέσματα καταγράφηκαν τιμές της εντροπίας ίσες με 0.167 και 0.156 αντίστοιχα, γεγονός που αποκαλύπτει ότι είναι προτιμότερη η εκτέλεση του αλγόριθμου χωρίς στάθμιση παρά την χρησιμοποίηση μιας στάθμισης η οποία δεν είναι αντιπροσωπευτική των δεδομένων εισόδου. Τελικά, η μη αντιπροσωπευτική στάθμιση των χαρακτηριστικών των προτύπων λόγω ανεπαρκούς πλήθους δειγμάτων όχι μόνο δεν προσφέρει στην αύξηση της επίδοσης αλλά φαίνεται να την μειώνει.

Βιβλιογραφία

[Bar02] Barber D., *Learning from Data – Linear Dimension Reduction*, 2001-2002

[BBM02] Basu S., Banerjee A. and Mooney R., *Semi-supervised Clustering by Seeding*, ICML, 2002

[Bor97] Borgatti S., *Multidimensional Scaling*, 1997

[BBD00] Bradley P. S., Bennett K. P. and Demiriz A., *Constrained K-Means Clustering*, MSR-TR-2000-65, 2000

[CCM00] Cohn D., Caruana R. and McCallum A., *Semi-supervised Clustering with User Feedback*, AAAI, 2000

[Dub96] Dubin D., *Structure in Document Browsing Spaces*, PhD thesis, School of Information Sciences, University of Pittsburgh, 1996.

[HMM99] Harper D. J., Mechkour M. and Muresan G., *Document Clustering for Mediated Information Access*, 21st BCS-IRSG Colloquium on Information Retrieval, Glasgow, 1999

[HWR03] Huang S., Ward M. O. and Rundensteiner E. A., *Exploration of Dimensionality Reduction for Text Visualization*, 2003

[JR71] Jardine N. and Rijsbergen van C. J., *The use of hierarchical clustering in information retrieval*, Information Storage and Retrieval, 7:217-240, 1971

[JPR] Jeon M., Park H. and Rosen J. B., *Dimension reduction based on centroids and least squares for efficient processing of text data*

[KH00] Karypis G. and Han E. H., *Centroid-Based Document Classification: Analysis & Experimental Results*, Technical Report TR-00-017, Department of Computer Science, University of Minnesota, Minneapolis, 2000

[KS] Karypis G. and Shankar S., *Weight adjustment schemes for a centroid based classifier*, Technical Report TR 00-035

[KZ02] Karypis G. and Zhao Y., *Criterion Functions for Document Clustering – Experiments and Analysis*, Technical Report TR-01-40, 2002

[Kas98] Kaski S., *Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering*, 1998

[KL00] Kim H. J. and Lee S. G., *A Semi-Supervised Document Clustering Technique for Information Organization*, CIKM 2000, McLean, VA USA, 2000

[MS99] Mendes M. E. S. and Sacks L., *Evaluating Fuzzy Clustering for Relevance-based Information Access*, 1999

[Mur02] Muresan G., *Using Document Clustering and Language Modelling in Mediated Information Retrieval*, 2002

[ND] Nürnberger A. and Detyniecki M., *Weighted Self-Organizing Maps: Incorporating User Feedback*

[RG] Rüger S. M. and Gauch S. E., *Feature Reduction for Document Clustering and Classification*

[SKK00] Steinbach M., Karypis G. and Kumar V., *A Comparison of Document Clustering Techniques*, Technical Report #00-034, In KDD Workshop on Text Mining, 2000

[WCR+01] Wagstaff K., Cardie C., Rogers S. and Schroedl S., *Constrained K-means Clustering with Background Knowledge*, Proceedings of the 18th International Conference on Machine Learning (pp. 577-584), 2001

[Wil83] Willet P., *Similarity coefficients and weighting functions for automatic document classification: an empirical comparison*, International Classification, 10(3):138–142, 1983

[WCW02] Wong K. F., Chan N. K. and Wong K. L., *Improving Document Clustering by Utilizing Meta-Data*, The Chinese University of Hong Kong, 2002

[XJN+] Xing E. P., Jordan M. I., Ng A. Y. and Russell S., *Distance metric learning, with application to clustering with side-information*

[YP97] Yang Y. and Pedersen J. O., *A Comparative Study on Feature Selection in Text Categorization*, Proceedings of the 14th International Conference on Machine Learning, 1997

[ZKY03] Zhang Z., Kwok J. T. and Yeung D., *Parametric Distance Metric Learning with Label Information*, 2003

Παράρτημα

Στο παρόν παράρτημα παρατίθεται μέρος του κώδικα της εφαρμογής SCAgent και ειδικότερα οι σημαντικότερες συναρτήσεις και μέθοδοι που υλοποιούν τα κρίσιμα σημεία της θεωρίας που αναπτύχθηκε στα προηγούμενα κεφάλαια. Κρίθηκε σκόπιμο όπως παραληφθεί το τμήμα του κώδικα το οποίο αφορά σε πτυχές δευτερευούσης σημασίας όπως η υλοποίηση της γραφικής διεπαφής με τον χρήστη (*Graphical user interface, GUI*) και οργάνωσης της γενικότερης αρχιτεκτονικής της εφαρμογής.

Π.1 Βασικότερες μέθοδοι της κλάσης formMain

```
Private Sub ExecuteQuery()
        Dim lvIntermediateQuery As New ArrayList     ' Contains the results of an
                                                       intermediate query
        Dim lvResultOffset As Integer                ' The offset from which Google
                                                       should return results
        Dim lvInformative As Integer                 ' The number of results which
                                                       contain keyword information
        Dim lvCurrentLink As String                  ' The URL of the current result
        Dim lvCurrentHTML As String                  ' The raw HTML content of the
                                                       current result
        Dim lvCurrentKeywords As ArrayList           ' The keywords' list of the
                                                       current result
        Dim lvKeywordsAll As ArrayList               ' The keywords' super-list of all
                                                       results

        ' Initialization
        Me.listboxQueryResults.Items.Clear()
        Me.listboxKeywords.Items.Clear()
        Me.listboxRelevantDocuments.Items.Clear()
        poKernel.QueryPerformed = False
        poKernel.ApplicationState = classKernel.enumApplicationState.Processing

        ' Updates GUI
        Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)

        ' Detects availability of network connection
        SetStatus(" Detecting availability of network connection...", 0)

        If Not IsConnectionAvailable() Then
            SetStatus(, 100)
            MsgBox("No available network connection !", _
                   MsgBoxStyle.Critical + MsgBoxStyle.OKOnly, _
                   " Exception")
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Performs query using Google API
        SetStatus(" Executing query...", 0)
        Do
            System.Windows.Forms.Application.DoEvents()

            ' Detects cancellation request
            If poKernel.CancelRequested Then
                Me.listboxQueryResults.Items.Clear()
                Me.listboxKeywords.Items.Clear()
                Me.listboxRelevantDocuments.Items.Clear()
```

```
            poKernel.CancelRequested = False
            poKernel.QueryPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Retrieves intermediate query results
        lvIntermediateQuery = GoogleAPIQuery(Me.textboxQueryString.Text.Trim & _
            " filetype:html", lvResultOffset)
        lvResultOffset += 10

        ' Detects exception during intermediate query execution
        If lvIntermediateQuery Is Nothing Then
            MsgBox("An exception has occured while executing query !", _
                   MsgBoxStyle.Critical + MsgBoxStyle.OKOnly, _
                   " Exception")
            Me.listboxQueryResults.Items.Clear()
            Me.listboxKeywords.Items.Clear()
            Me.listboxRelevantDocuments.Items.Clear()
            poKernel.QueryPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)
            Exit Sub
        Else
            For i As Integer = 1 To lvIntermediateQuery.Count
                System.Windows.Forms.Application.DoEvents()

                ' Detects cancellation request
                If poKernel.CancelRequested Then
                    Me.listboxQueryResults.Items.Clear()
                    Me.listboxKeywords.Items.Clear()
                    Me.listboxRelevantDocuments.Items.Clear()
                    poKernel.QueryPerformed = False
                    poKernel.CancelRequested = False
                    poKernel.ApplicationState = _
                      classKernel.enumApplicationState.Idle
                    Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)
                    Exit Sub
                End If

                ' Parses current result
                lvCurrentLink = lvIntermediateQuery.Item(i - 1).ToString

                If Not poQuery.ResultExists(lvCurrentLink) Then
                    lvCurrentHTML = RetrieveHTML(lvCurrentLink, 3000)
                    lvCurrentKeywords = New _
                      ArrayList(StripKeywords(StripMetaTag(lvCurrentHTML)))

                    poQuery.ResultAdd( _
                      New classResult(lvCurrentLink, lvCurrentKeywords))
                    Me.listboxQueryResults.Items.Add(lvCurrentLink)
                    Me.listboxRelevantDocuments.Items.Add(lvCurrentLink, True)
                    Me.labelInfoQueryResults.Text = _
                      Format(poQuery.ResultCount, "#,##0")
                    Me.labelInfoDescriptive.Text = _
                      Format(poQuery.Informative, "#,##0")

                    ' Shows progress
                    SetStatus(" Retrieving query results...", _
                        CInt(poQuery.ResultCount / _
                        Math.Min(pvMaxResults, _
                                 CInt(Me.comboLimitResults.SelectedItem)) * 100))
                End If
            Next
        End If
    Loop Until lvIntermediateQuery.Count = 0 OrElse _
               poQuery.ResultCount = pvMaxResults OrElse _
               poQuery.ResultCount >= CInt(Me.comboLimitResults.SelectedItem)
    ' Fills keyword listbox with all keywords detected
    For i As Integer = 1 To poQuery.KeywordsAllCount
```

```
            Me.listboxKeywords.Enabled = True
            Me.listboxKeywords.Items.Add(poQuery.KeywordsAll.Item(i - 1), True)
        Next

        ' Updates GUI
        poKernel.QueryPerformed = True
        poKernel.ApplicationState = classKernel.enumApplicationState.Idle
        Me.tabpageWebQuery_VisibleChanged(Nothing, Nothing)
    End Sub

Private Sub ExecuteTransformation()
        Dim loTDF As New classVector                    ' Vector which contains keywords'
                                                          document frequencies

        ' Initialization
        loTDF.Length = poQuery.KeywordsSelectedCount
        loTDF.Clear()
        poQuery.MeansVector.Length = poQuery.KeywordsSelectedCount
        poQuery.MeansVector.Clear()
        poKernel.VectorsCreated = False
        poKernel.ApplicationState = classKernel.enumApplicationState.Processing

        ' Updates GUI
        Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)

        ' Constructs vector representation for each query result
        SetStatus(" Creating results' vector representations...", 0)
        For i As Integer = 1 To poQuery.ResultCount
            ' Initializes results' vector to accomodate current selection of keywords
            poQuery.ResultByIndex(i).VectorFull.Length = poQuery.KeywordsSelectedCount
            poQuery.ResultByIndex(i).VectorFull.Clear()

            For j As Integer = 1 To poQuery.KeywordsSelectedCount
                System.Windows.Forms.Application.DoEvents()

                ' Detects cancellation request
                If poKernel.CancelRequested Then
                    poKernel.CancelRequested = False
                    poKernel.VectorsCreated = False
                    poKernel.ApplicationState = classKernel.enumApplicationState.Idle
                    Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)
                    Exit Sub
                End If

                ' Constructs results' full vector & calculates keywords' df
                If poQuery.ResultByIndex(i).KeywordExists( _
                     poQuery.KeywordsSelected.Item(j - 1).ToString) Then
                    poQuery.ResultByIndex(i).VectorFull.Item(j) = 1
                    loTDF.Item(j) += 1
                End If

                ' Updates GUI
                SetStatus(" Creating document vector representations...", _
                          CInt(((i - 1) * poQuery.KeywordsSelectedCount + j) / _
                          (poQuery.ResultCount * poQuery.KeywordsSelectedCount)*100))
            Next
        Next

        ' Calculates log(N/df) for each keyword
        SetStatus(" Calculating inverse document frequencies (IDF) for all
                     keywords...", 0)
        For i As Integer = 1 To loTDF.Length
            System.Windows.Forms.Application.DoEvents()

            ' Detects cancellation request
            If poKernel.CancelRequested Then
                poKernel.CancelRequested = False
                poKernel.VectorsCreated = False
                poKernel.ApplicationState = classKernel.enumApplicationState.Idle
                Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)
```

```
            Exit Sub
        End If


        ' Prevents division by zero exception
        If loTDF.Item(i) > 0 Then _
            loTDF.Item(i) = Math.Log10(poQuery.ResultCount / loTDF.Item(i))

        ' Shows progress
        SetStatus(, CInt(i / loTDF.Length * 100))
    Next

    ' Calculates tf * log(N/dfi) for each keyword
    SetStatus(" Calculating KF-IDF for all keywords and normalizing keyword
                  vectors...", 0)

    For i As Integer = 1 To poQuery.ResultCount
        System.Windows.Forms.Application.DoEvents()

        ' Detects cancellation request
        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.VectorsCreated = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Performs result vectors' adjustment and normalization
        poQuery.ResultByIndex(i).VectorFull.Multiply(loTDF)
        poQuery.ResultByIndex(i).VectorFull.Normalize()

        ' Constructs keywords' mean vector
        poQuery.MeansVector.Add(poQuery.ResultByIndex(i).VectorFull)
        If i = poQuery.ResultCount Then _
            poQuery.MeansVector.Multiply(1 / poQuery.ResultCount)

        ' Shows progress
        SetStatus(, CInt(i / poQuery.ResultCount * 100))
    Next

    ' Draws KF-IDF chart
    SetStatus(" Charting keyword frequencies...", 0)

    With Me.chartKeywordTFIDF
        .ChartGroups.ChartGroupsCollection(0).ChartData.SeriesList(0). _
            PointData.Clear()
        .ChartArea.AxisX.AutoMin = False
        .ChartArea.AxisX.AutoMax = False
        .ChartArea.AxisX.Min = 1
        .ChartArea.AxisX.Max = poQuery.KeywordsSelectedCount
        .ChartArea.AxisX.UnitMajor = ((poQuery.KeywordsSelectedCount - 1) \ 25) + 1
        .ChartArea.AxisX.UnitMinor = Me.chartKeywordTFIDF.ChartArea.AxisX.UnitMajor
        .ChartArea.AxisY.AutoMin = False
        .ChartArea.AxisY.AutoMax = False
        .ChartArea.AxisY.Min = 0
        .ChartArea.AxisY.Max = poQuery.MeansVector.MaxItem
        .ChartArea.AxisY.UnitMajor = Format(.ChartArea.AxisY.Max / 10, "0.0000")
        .ChartArea.AxisY.UnitMinor = _
            Format(.ChartArea.AxisY.UnitMajor / 5, "0.0000")
        .ChartArea.AxisY.GridMajor.Spacing =.ChartArea.AxisY.UnitMajor
    End With

    For i As Integer = 1 To poQuery.MeansVector.Length
        System.Windows.Forms.Application.DoEvents()

        ' Detects cancellation request
        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.VectorsCreated = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
```

```
            Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Adds new data point to chart
        Me.chartKeywordTFIDF.ChartGroups.ChartGroupsCollection(0). _
            ChartData.SeriesList(0).PointData.Add( _
                 New PointF(i, poQuery.MeansVector.Item(i)))

        ' Shows progress
        SetStatus(, CInt(i / poQuery.MeansVector.Length * 100))
    Next

    ' Updates GUI
    Me.trackbarThreshold.Maximum = 1000 * poQuery.MeansVector.MaxItem
    Me.trackbarThreshold.LargeChange = CInt(Me.trackbarThreshold.Maximum / 10)
    Me.trackbarThreshold.SmallChange = 1
    Me.trackbarThreshold.TickFrequency = CInt(Me.trackbarThreshold.Maximum / 20)
    Me.trackbarThreshold.Value = Me.trackbarThreshold.Maximum
    Me.trackbarThreshold.Value = 0

    poKernel.VectorsCreated = True
    poKernel.ApplicationState = classKernel.enumApplicationState.Idle
    Me.tabpageDetectedKeywords_VisibleChanged(Nothing, Nothing)
End Sub

Private Sub ExecuteKeywordReduction()
    Dim lvSignificantTerms As New ArrayList     ' List of the keywords that meet
                                                  reduction criteria
    Dim lvThreshold As Single                   ' The frequency threshold

    ' Initialization
    poKernel.TermReductionPerformed = False
    poKernel.ApplicationState = classKernel.enumApplicationState.Processing

    ' Updates GUI
    Me.tabpageKeywordReduction_VisibleChanged(Nothing, Nothing)

    ' Determines significant keywords based on TFIDF mean values
    SetStatus(" Determining keywords for removal...", 0)

    ' Parses textbox content
    Try
        lvThreshold = CSng(Me.textboxKeywordThreshold.Text)
    Catch ex As Exception
        MsgBox("An exception has occured while parsing frequency threshold !", _
               MsgBoxStyle.Critical + MsgBoxStyle.OKOnly, _
               " Exception")
        poKernel.TermReductionPerformed = False
        poKernel.ApplicationState = classKernel.enumApplicationState.Idle
        Me.tabpageKeywordReduction_VisibleChanged(Nothing, Nothing)
        Exit Sub
    End Try

    For i As Integer = 1 To poQuery.MeansVector.Length
        System.Windows.Forms.Application.DoEvents()

        ' Detects cancellation request
        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.TermReductionPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageKeywordReduction_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Checks current keyword's mean value
        If poQuery.MeansVector.Item(i) >= lvThreshold Then
            poQuery.KeywordsReduced.Add(poQuery.KeywordsSelected.Item(i - 1))
            lvSignificantTerms.Add(CInt(i))
            Me.labelInfoKeywordsAfter.Text = _
```

```
                Format(lvSignificantTerms.Count, "###,##0")
            Me.labelInfoKeywordReduction.Text = _
                Format((poQuery.KeywordsSelectedCount -
            poQuery.KeywordsReducedCount)/poQuery.KeywordsSelectedCount,"#0.00%")
        End If

        ' Shows progress
        SetStatus(, CInt(i / poQuery.MeansVector.Length * 100))
    Next

    ' Applies term reduction
    SetStatus(" Applying keyword reduction...", 0)

    For i As Integer = 1 To poQuery.ResultCount
        System.Windows.Forms.Application.DoEvents()

        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.TermReductionPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageKeywordReduction_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        poQuery.ResultByIndex(i).ReduceTerms(lvSignificantTerms)

        ' Shows progress
        SetStatus(, CInt(i / poQuery.ResultCount * 100))
    Next

    ' Updates GUI
    poKernel.TermReductionPerformed = True
    poKernel.ApplicationState = classKernel.enumApplicationState.Idle
    Me.tabpageKeywordReduction_VisibleChanged(Nothing, Nothing)
End Sub

Private Sub ExecuteKeywordWeighting()
    ' Initialization
    poKernel.TermWeightingPerformed = False
    poQuery.PurityVector = New classVector(poQuery.KeywordsReducedCount)

    ' Updates GUI
    poKernel.ApplicationState = classKernel.enumApplicationState.Processing
    Me.tabpageKeywordWeighting_VisibleChanged(Nothing, Nothing)

    ' Calculates purity vector
    SetStatus(" Calculating keywords' purity vector", 0)

    For i As Integer = 1 To Me.listboxRelevantDocuments.CheckedItems.Count
        System.Windows.Forms.Application.DoEvents()

        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.TermWeightingPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageKeywordWeighting_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        ' Adds current result's reduced vector
        poQuery.PurityVector.Add(poQuery.ResultByURL( _
            Me.listboxRelevantDocuments.CheckedItems(i - 1)).VectorReduced)

        ' Calculates mean vector if at last iteration
        If i = Me.listboxRelevantDocuments.CheckedItems.Count Then _
                        poQuery.PurityVector.Multiply(1 / _
                        Me.listboxRelevantDocuments.CheckedItems.Count)

        ' Shows progress
        SetStatus(, CInt(i / Me.listboxRelevantDocuments.CheckedItems.Count*100))
    Next
```

```
        ' Updates GUI
        poKernel.TermWeightingPerformed = True
        poKernel.ApplicationState = classKernel.enumApplicationState.Idle
        Me.tabpageKeywordWeighting_VisibleChanged(Nothing, Nothing)
    End Sub

Private Sub ExecuteClustering()
        Dim lvOldCentroid As classVector            ' Old cluster centroid
        Dim lvNewCentroid As classVector            ' New cluster centroid
        Dim lvTolerance As Single                   ' Convergence tolerance limit
        Dim lvSchemeSize As Short                   ' The size of the scheme to create
        Dim lvIteration As Short                    ' Current iteration number
        Dim lvConvergence As Boolean                ' Convergence flag
        Dim lvMinimumDistance As Single
        Dim lvCurrentDistance As Single             ' Current vectors' distance
        Dim lvDocumentsSoFar As Integer             ' Results processed so far
        Dim lvStartingTime As TimeSpan              ' Executing starting time
        Dim lvTargetClusters As New SortedList      ' List of clusters sorted on
                                                      minimum distance

        ' Initialization
        poKernel.ClusteringPerformed = False
        Me.listviewSchemeInformation.Items.Clear()
        Me.listviewSchemeInformation.BackColor = Color.FromArgb(245, 255, 245)
        Me.listviewSchemeInformation.Columns.Item(0).Width = 138
        Me.listviewSchemeInformation.Columns.Item(1).Width = 80
        Me.listviewSchemeInformation.Columns.Item(2).Width = 80
        Me.listviewSchemeInformation.Columns.Item(3).Width = 80
        Me.listviewSchemeInformation.Columns.Item(4).Width = 80
        Me.listviewSchemeInformation.Columns.Item(5).Width = 78
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(0).PointData.Clear()
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(0).PointData.Add(New PointF(0, 0))
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(1).PointData.Clear()
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(1).PointData.Add(New PointF(0, 0))
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(2).PointData.Clear()
        Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
           ChartData.SeriesList(2).PointData.Add(New PointF(0, 0))

        ' Updates GUI
        poKernel.ApplicationState = classKernel.enumApplicationState.Processing
        Me.tabpageClustering_VisibleChanged(Nothing, Nothing)

        ' Creates clustering scheme
        SetStatus(" Creating clustering scheme...", 0)

        Try
            lvSchemeSize = CInt(Me.textboxSchemeSize.Text)
            lvTolerance = CSng(Me.textboxConvergenceTolerance.Text)
            poScheme.Create(lvSchemeSize)
        Catch ex As Exception
            MsgBox("An exception has occured while creating clustering scheme !", _
                   MsgBoxStyle.Critical + MsgBoxStyle.OKOnly, _
                   " Exception")
            poKernel.ClusteringPerformed = False
            poKernel.ApplicationState = classKernel.enumApplicationState.Idle
            Me.tabpageClustering_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End Try


        ' K-Means algorithm main body
        lvStartingTime = Now.TimeOfDay

        Do
            SetStatus(" Assigning results to clusters...", 0)
```

```
lvIteration += 1
lvConvergence = True
lvDocumentsSoFar = 0

Me.labelInfoIteration.Text = lvIteration.ToString

For i As Integer = 1 To poQuery.ResultCount
    lvTargetClusters.Clear()

    ' Calculates distance of current result from all clusters
    For j As Integer = 1 To poScheme.Size
        System.Windows.Forms.Application.DoEvents()

        ' Detects cancellation request
        If poKernel.CancelRequested Then
            poKernel.CancelRequested = False
            poKernel.ClusteringPerformed = False
            poKernel.ApplicationState = _
              classKernel.enumApplicationState.Idle
            Me.tabpageClustering_VisibleChanged(Nothing, Nothing)
            Exit Sub
        End If

        If poQuery.PurityVector Is Nothing Then
            ' Non-weighted version of K-Means
            lvCurrentDistance = classVector.Norm( _
              classVector.Add(poQuery.ResultByIndex(i).VectorReduced, _
              classVector.Multiply( _
             poScheme.ClusterByIndex(j).VectorCentroid, -1)), 2)
        Else
            ' Weighted version of K-Means
            lvCurrentDistance = classVector.Norm( _
                                 classVector.Multiply( _
              classVector.Add(poQuery.ResultByIndex(i).VectorReduced, _
             classVector.Multiply( _
             poScheme.ClusterByIndex(j).VectorCentroid, -1)), _
             poQuery.PurityVector), 2)
        End If

        ' Constructs list of target clusters
        If Not lvTargetClusters.Contains(CSng(lvCurrentDistance)) Then _
            lvTargetClusters.Add(CSng(lvCurrentDistance), _
              poScheme.ClusterByIndex(j))

        ' Shows progress
        SetStatus(, CInt(((i - 1) * poScheme.Size + j) / _
            (poScheme.Size * poQuery.ResultCount) * 100))
        Me.labelInfoTimeElapsed.Text = TimeElapsed(lvStartingTime)
    Next

    ' Assigns current document to the most close cluster
    poScheme.MoveDocument(poQuery.ResultByIndex(i), _
         lvTargetClusters.GetByIndex(0))
Next
' Calculates centroid vector for each cluster
SetStatus(" Updating clusters' centroid vectors...", 0)

For i As Integer = 1 To poScheme.Size
    System.Windows.Forms.Application.DoEvents()

    ' Detects cancellation request
    If poKernel.CancelRequested Then
        poKernel.CancelRequested = False
        poKernel.ClusteringPerformed = False
        poKernel.ApplicationState = classKernel.enumApplicationState.Idle
        Me.tabpageClustering_VisibleChanged(Nothing, Nothing)
        Exit Sub
    End If

    lvDocumentsSoFar += poScheme.ClusterByIndex(i).Size
    lvOldCentroid = _
```

```
            New classVector(poScheme.ClusterByIndex(i).VectorCentroid)
        poScheme.ClusterByIndex(i).CalculateCentroid()
        lvNewCentroid = _
            New classVector(poScheme.ClusterByIndex(i).VectorCentroid)
        lvNewCentroid.Add(classVector.Multiply(lvOldCentroid, -1))
        If lvNewCentroid.Norm(2) > lvTolerance Then lvConvergence = False

        ' Updates cluster information
        If Me.listviewSchemeInformation.Items.Count < poScheme.Size Then
            Dim lvListItem As New ListViewItem

            With lvListItem
              .Text = poScheme.ClusterByIndex(i).Name
              .SubItems.Add(poScheme.ClusterByIndex(i).Size)
              .SubItems.Add(Format(poScheme.ClusterByIndex(i).Size / _
                  poQuery.ResultCount * 100, "#0.000"))
              .SubItems.Add(Format(poScheme.ClusterByIndex(i).Entropy, _
                  "#0.000"))
              .SubItems.Add(Format(poScheme.ClusterByIndex(i).Similarity, _
                  "#0.000"))
              .SubItems.Add(Format(poScheme.ClusterByIndex(i).Purity, _
                  "#0.000"))
            End With

            Me.listviewSchemeInformation.Items.Add(lvListItem)
        Else
            With Me.listviewSchemeInformation
             Items(i - 1).SubItems(1).Text = _
                  Format(poScheme.ClusterByIndex(i).Size, "#0")
             .Items(i - 1).SubItems(2).Text = _
                  Format(poScheme.ClusterByIndex(i).Size / _
                  poQuery.ResultCount * 100, "#0.000")
             .Items(i - 1).SubItems(3).Text = _
                  Format(poScheme.ClusterByIndex(i).Entropy, "#0.000")
            .Items(i - 1).SubItems(4).Text = _
                  Format(poScheme.ClusterByIndex(i).Similarity, _
                  "#0.000")
            .Items(i - 1).SubItems(5).Text = _
                  Format(poScheme.ClusterByIndex(i).Purity, "#0.000")
            End With
        End If

        ' Shows progress
        SetStatus(, CInt(lvDocumentsSoFar / poQuery.ResultCount * 100))
        Me.labelInfoTimeElapsed.Text = TimeElapsed(lvStartingTime)
    Next

    ' Updates side information bar
    SetStatus(" Calculating scheme measures...", 0)

    Me.labelInfoEntropy.Text = Format(poScheme.Entropy, "#0.000")
    Me.labelInfoSimilarity.Text = Format(poScheme.Similarity, "#0.000")
    Me.labelInfoPurity.Text = Format(poScheme.Purity, "#0.000")

    ' Updates entropy chart
    Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
        ChartData.SeriesList(0).PointData.Add( _
            New PointF(lvIteration, CSng(Me.labelInfoEntropy.Text.Trim)))
    Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
        ChartData.SeriesList(1).PointData.Add( _
            New PointF(lvIteration, CSng(Me.labelInfoSimilarity.Text.Trim)))
    Me.chartProgress.ChartGroups.ChartGroupsCollection(0). _
        ChartData.SeriesList(2).PointData.Add( _
            New PointF(lvIteration, CSng(Me.labelInfoPurity.Text.Trim)))


    Me.labelInfoTimeElapsed.Text = TimeElapsed(lvStartingTime)
    SetStatus(, 100)
Loop Until lvIteration = CInt(Me.textboxMaximumIterations.Text) OrElse _
            lvConvergence = True
```

```
        ' Updates GUI
        poKernel.ClusteringPerformed = True
        ExecuteSummary()
        poKernel.ApplicationState = classKernel.enumApplicationState.Idle
        Me.listviewSchemeInformation.BackColor = Color.White
        Me.tabpageClustering_VisibleChanged(Nothing, Nothing)
    End Sub
```

Π.2 Κλάσεις μοντελοποίησης οντοτήτων

```
' This class implements a float number vector with variable size
Public NotInheritable Class classVector

#Region " Instance members "
    ' The array containing vector's values
    Private cvSubscripts() As Single
#End Region

#Region " Properties "
    ' Property that returns the value of supreme subscript
    Public ReadOnly Property MaxItem() As Single
        Get
            Dim lvResult As Single = 0

            For i As Integer = 1 To cvSubscripts.Length
                If cvSubscripts(i - 1) > lvResult Then lvResult = cvSubscripts(i - 1)
            Next

            Return lvResult
        End Get
    End Property

    ' Property that sets/returns the size of the current instance
    Public Property Length() As Integer
        Get
            Return cvSubscripts.Length
        End Get
        Set(ByVal Value As Integer)
            If Value >= 0 AndAlso Value <> cvSubscripts.Length Then
                Try
                    ReDim Preserve cvSubscripts(Value - 1)
                Catch ex As Exception
                    MsgBox("Exception raised due to invalid argument specification" &_
                           vbNewLine & _
                           "during call of property classVector.Length().")
                End Try
            End If
        End Set
    End Property

    ' Property that sets/returns a specific subscript value of the current instance
    Public Property Item(ByVal s As Integer) As Single
        Get
            If s > 0 AndAlso s <= cvSubscripts.Length Then
                Return cvSubscripts(s - 1)
            Else
                MsgBox("Exception raised due to invalid vector subscript index " & _
                       "specification" & vbNewLine & _
                       "during call of property classVector.Item(ByVal s As " & _
                       "Integer).")
                Return Nothing
            End If
        End Get
        Set(ByVal Value As Single)
            If s > 0 AndAlso s <= cvSubscripts.Length Then
                cvSubscripts(s - 1) = Value
            Else
                MsgBox("Exception raised due to invalid vector subscript index " & _
                       "specification" & vbNewLine & _
```

```
                    "during call of property classVector.Item(ByVal s As " & _
                    "Integer).")
        End If
    End Set
End Property

' Property that returns a value that indicates whether a vector is empty
Public ReadOnly Property IsZero() As Boolean
    Get
        For i As Integer = 1 To cvSubscripts.Length
            If cvSubscripts(i - 1) > 0 Then Return False
        Next
        Return True
    End Get
End Property

' Property that returns the geometric mean of current instance's values
Public ReadOnly Property GeometricMean() As Single
    Get
        If cvSubscripts.Length > 0 Then
            Dim lvResult As Single = 1

            For i As Integer = 1 To cvSubscripts.Length
                lvResult *= cvSubscripts(i - 1)
            Next

            Return Math.Pow(lvResult, 1 / cvSubscripts.Length)
        Else
            Return 0
        End If
    End Get
End Property

' Property that returns the arithmetic mean of current instance's values
Public ReadOnly Property ArithmeticMean() As Single
    Get
        Dim lvResult As Single

        If cvSubscripts.Length > 0 Then
            For i As Integer = 1 To cvSubscripts.Length
                lvResult += cvSubscripts(i - 1)
            Next

            Return lvResult / cvSubscripts.Length
        Else
            Return 0
        End If
    End Get
End Property

' Property that returns a specific norm of the current instance
Public ReadOnly Property Norm(ByVal s As Integer) As Single
    Get
        If s >= 1 Then
            Dim lvResult As Single

            For i As Integer = 1 To cvSubscripts.Length
                lvResult += Math.Pow(cvSubscripts(i - 1), s)
            Next

            Return Math.Pow(lvResult, 1 / s)
        Else
            MsgBox("Exception raised due to invalid norm specification" & _
                   vbNewLine & _
                   "during call of property classVector.Norm(ByVal s As " & _
                   "Integer).")
            Return Nothing
        End If
    End Get
End Property
```

```
    ' Property that returns a specific norm of the current instance
    Public Shared ReadOnly Property Norm(ByVal s1 As classVector, ByVal s2 As Integer)
As Single
        Get
            If Not s1 Is Nothing AndAlso s2 >= 1 Then
                Dim lvResult As Single

                For i As Integer = 1 To s1.cvSubscripts.Length
                    lvResult += Math.Pow(s1.cvSubscripts(i - 1), s2)
                Next

                Return Math.Pow(lvResult, 1 / s2)
            Else
                MsgBox("Exception raised due to invalid argument specification" & _
                       vbNewLine & _
                       "during call of shared property classVector.Norm(Byval " & _
                       "s1 As classVector, ByVal s2 As Integer).")
                Return Nothing
            End If
        End Get
    End Property

    ' Property that returns the standard deviation of current instance's values
    Public ReadOnly Property StandardDeviation() As Single
        Get
            Return Math.Sqrt(Me.Variance)
        End Get
    End Property

    ' Property that returns a string representation of the current instance
    Public Shadows ReadOnly Property ToString() As String
        Get
            If cvSubscripts.Length > 0 Then
                Dim lvResult As String

                For i As Integer = 1 To cvSubscripts.Length
                    lvResult &= Format(cvSubscripts(i - 1), "0.000 ")
                Next

                Return "[ " & lvResult & "]"
            Else
                Return String.Empty
            End If
        End Get
    End Property

    ' Property that returns the variance of current instance's values
    Public ReadOnly Property Variance() As Single
        Get
            If cvSubscripts.Length > 0 Then
                Return (cvSubscripts.Length * Math.Pow(Me.Norm(2), 2) - _
                        Math.Pow(Me.Norm(1), 2)) / _
                        Math.Pow(cvSubscripts.Length, 2)
            Else
                Return 0
            End If
        End Get
    End Property
#End Region

#Region " Methods "
    ' Method that adds the specified vector to the current instance
    Public Sub Add(ByRef s As classVector)
        If s Is Nothing OrElse cvSubscripts.Length <> s.cvSubscripts.Length Then
            MsgBox("Exception raised due to incompatible vector specification" & _
                   vbNewLine & _
                   "during call of method classVector.Add(ByRef s As classVector).")
        Else
            For i As Integer = 1 To cvSubscripts.Length
                cvSubscripts(i - 1) += s.cvSubscripts(i - 1)
            Next
```

```
        End If
    End Sub

    ' Method that returns the result of the addition of the specified vectors
    Public Shared Function Add(ByRef s1 As classVector, ByRef s2 As classVector) As
classVector
        If s1 Is Nothing OrElse s2 Is Nothing OrElse s1.cvSubscripts.Length <>
s2.cvSubscripts.Length Then
            MsgBox("Exception raised due to incompatible vectors specification" & _
                   vbNewLine & _
                   "during call of shared method classVector.Add(ByRef s1 As " & _
                   classVector, ByRef s2 As classVector).")
            Return Nothing
        Else
            Dim lvResult As New classVector(s1)

            For i As Integer = 1 To lvResult.cvSubscripts.Length
                lvResult.cvSubscripts(i - 1) += s2.cvSubscripts(i - 1)
            Next

            Return lvResult
        End If
    End Function

    ' Method that clears (= sets to zero) all subscripts of the current instance
    Public Sub Clear()
        For i As Integer = 1 To cvSubscripts.Length
            cvSubscripts(i - 1) = 0
        Next
    End Sub

    ' Method that clears (= sets to zero) all subscripts of the specified vector
    Public Shared Sub Clear(ByRef s As classVector)
        For i As Integer = 1 To s.cvSubscripts.Length
            s.cvSubscripts(i - 1) = 0
        Next
    End Sub

    ' Method that reduces the size of the current instance by preserving only the
specified subscripts
    Public Sub Reduce(ByRef s As ArrayList)
        If Not s Is Nothing Then
            s.Sort()

            For i As Integer = 1 To s.Count
                cvSubscripts(i - 1) = cvSubscripts(CInt(s.Item(i - 1)) - 1)
                'Me.Item(i) = Me.Item(CInt(s.Item(i - 1)))
            Next

            Me.Length = s.Count
        Else
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of method classVector.Reduce(ByRef s As Arraylist).")
        End If
    End Sub

    ' Method that returns the result of reducing the specified vector's size by
preserving only the specified subscripts
    Public Shared Function Reduce(ByRef s1 As classVector, ByRef s2 As ArrayList) As
classVector
        Dim lvResult As New classVector(s2.Count)

        If Not s1 Is Nothing OrElse s2 Is Nothing Then
            s2.Sort()

            For i As Integer = 1 To s2.Count
                lvResult.cvSubscripts(i - 1) = s1.cvSubscripts(CInt(s2.Item(i - 1)) -
1)
                'lvResult.Item(i) = s1.Item(CInt(s2.Item(i - 1)))
            Next
```

```
            Return lvResult
        Else
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of shared method classVector.Reduce(ByRef s1 as " & _
                   classVector, ByRef s2 As Arraylist).")
            Return Nothing
        End If
    End Function

    ' Method that multiplies current instance by the specified constant
    Public Sub Multiply(ByVal s As Single)
        For i As Integer = 1 To cvSubscripts.Length
            cvSubscripts(i - 1) *= s
        Next
    End Sub

    ' Method that returns the result of multiplication between the specified vector
and the specified constant
    Public Shared Function Multiply(ByRef s1 As classVector, ByVal s2 As Single) As
classVector
        Dim lvResult As New classVector(s1)

        If Not s1 Is Nothing Then
            For i As Integer = 1 To lvResult.cvSubscripts.Length
                lvResult.cvSubscripts(i - 1) *= s2
            Next

            Return lvResult
        Else
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of shared method classVector.Multiply(ByRef s1 " & _
                   "As classVector, ByVal s2 As Single).")
            Return Nothing
        End If
    End Function

    ' Method that multiplies current instance's subscripts by the respective specified
vector's subscripts
    Public Sub Multiply(ByRef s As classVector)
        If s Is Nothing OrElse cvSubscripts.Length <> s.cvSubscripts.Length Then
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of method classVector.Multiply(ByRef s As " & _
                   "classVector).")
        Else
            For i As Integer = 1 To cvSubscripts.Length
                cvSubscripts(i - 1) *= s.cvSubscripts(i - 1)
            Next
        End If
    End Sub

    ' Method that returns the result of multiplication between the specified vectors'
respective subscripts
    Public Shared Function Multiply(ByRef s1 As classVector, ByRef s2 As classVector)
As classVector
        If s1 Is Nothing OrElse s2 Is Nothing OrElse s1.cvSubscripts.Length <>
s2.cvSubscripts.Length Then
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of shared method classVector.Add(ByRef s1 As " & _
                   "classVector, ByRef s2 As classVector).")
            Return Nothing
        Else
            Dim lvResult As New classVector(s1)

            For i As Integer = 1 To lvResult.cvSubscripts.Length
                lvResult.cvSubscripts(i - 1) *= s2.cvSubscripts(i - 1)
            Next
```

```
            Return lvResult
        End If
    End Function

    ' Method that returns the dot product of the current instance and the specified
vector
    Public Function DotProduct(ByRef s As classVector) As Single
        If s Is Nothing OrElse cvSubscripts.Length <> s.cvSubscripts.Length Then
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of method classVector.DotProduct(ByRef s As " & _
                   classVector).")
            Return Nothing
        Else
            Dim lvResult As Single

            For i As Integer = 1 To cvSubscripts.Length
                lvResult += cvSubscripts(i - 1) * s.cvSubscripts(i - 1)
            Next

            Return lvResult
        End If
    End Function

    ' Method that returns the dot product of the specified vectors
    Public Shared Function DotProduct(ByRef s1 As classVector, ByRef s2 As
classVector) As Single
        If s1 Is Nothing OrElse s2 Is Nothing OrElse s1.Length <> s2.Length Then
            MsgBox("Exception raised due to invalid argument specification" & _
                   vbNewLine & _
                   "during call of shared method classVector.DotProduct(ByRef s1 " & _
                   "As classVector, ByRef s2 As classVector).")
            Return Nothing
        Else
            Dim lvResult As Single

            For i As Integer = 1 To s1.cvSubscripts.Length
                lvResult += s1.cvSubscripts(i - 1) * s2.cvSubscripts(i - 1)
            Next

            Return lvResult
        End If
    End Function

    ' Method that normalizes current instance to unit length
    Public Sub Normalize()
        Dim lclResult As Single = Me.Norm(2)

        If lclResult > 0 Then Me.Multiply(1 / lclResult)
    End Sub

    ' Method that returns the result of normalization of the specified vector to unit
length
    Public Shared Function Normalize(ByRef s As classVector) As classVector
        Dim lclResult As New classVector(s)

        If lclResult.Norm(2) > 0 Then lclResult.Multiply(1 / lclResult.Norm(2))

        Return lclResult
    End Function
#End Region

#Region " Constructors "
    ' Default class constructor
    Public Sub New()
        ReDim cvSubscripts(-1)
    End Sub

    ' Alternative constructor that initializes vector's size
    Public Sub New(ByVal s As Integer)
```

```
        If s >= 0 Then
            ReDim cvSubscripts(s - 1)
        Else
            ReDim cvSubscripts(-1)
        End If
    End Sub

    ' Alternative constructor that initializes current instance to a copy of the
specified vector
    Public Sub New(ByRef s As classVector)
        If Not s Is Nothing Then
            cvSubscripts = s.cvSubscripts.Clone
        Else
            ReDim cvSubscripts(-1)
        End If
    End Sub
#End Region

End Class

' This class implements the entity "Query result"
Public NotInheritable Class classResult

#Region " Instance members "
    ' The relevancy flag of the current instance
    Private cvIsRelevant As Boolean

    ' The URL of the current instance
    Private cvURL As String

    ' The keywords extracted from meta tag of the current instance
    Private cvKeywords As ArrayList

    ' The full term TF-IDF vector of the current instance
    Private coVectorFull As classVector

    ' The reduced term TF-IDF vector of the current instance
    Private coVectorReduced As classVector
#End Region

#Region " Properties "
    ' Property that returns the informative flag of the current instance
    Public ReadOnly Property IsInformative() As Boolean
        Get
            If cvKeywords.Count = 0 Then
                Return False
            Else
                If poKernel.VectorsCreated Then Return Not coVectorReduced.IsZero
                Return True
            End If
        End Get
    End Property

    ' Property that sets/returns the relevancy flag of the current instance
    Public Property IsRelevant() As Boolean
        Get
            Return cvIsRelevant
        End Get
        Set(ByVal Value As Boolean)
            cvIsRelevant = Value
        End Set
    End Property

    ' Property that returns a copy of the keywords list of the current instance
    Public ReadOnly Property KeywordList() As ArrayList
        Get
            Return cvKeywords.Clone
        End Get
    End Property

    ' Property that returns membership information about a specific keyword
```

```
    Public ReadOnly Property KeywordExists(ByVal s As String) As Boolean
        Get
            Return cvKeywords.Contains(s.ToString.ToLower.Trim)
        End Get
    End Property

    ' Property that returns the number of keywords for the current instance
    Public ReadOnly Property KeywordCount() As Integer
        Get
            Return cvKeywords.Count
        End Get
    End Property

    ' Property that returns a keyword accessed by index
    Public ReadOnly Property KeywordByIndex(ByVal s As Integer) As String
        Get
            If s >= 1 AndAlso s <= cvKeywords.Count Then _
                Return cvKeywords.Item(s - 1).ToString
            Return Nothing
        End Get
    End Property

    ' Property that returns the full term vector of the current instance
    Public ReadOnly Property VectorFull() As classVector
        Get
            Return coVectorFull
        End Get
    End Property

    ' Property that returns the reduced term vector of the current instance
    Public ReadOnly Property VectorReduced() As classVector
        Get
            Return coVectorReduced
        End Get
    End Property

    ' Property that returns the URL of the current instance
    Public ReadOnly Property URL() As String
        Get
            Return cvURL
        End Get
    End Property
#End Region

#Region " Methods "
    ' Method that adds the specified keyword to the keyword list of the current
instance
    Public Sub KeywordAdd(ByVal s As String)
        If s.Trim.Length > 0 AndAlso _
            Not cvKeywords.Contains(s.ToLower.ToLower.Trim) Then
            cvKeywords.Add(s.ToString.ToLower.Trim)
            cvKeywords.TrimToSize()
            cvKeywords.Sort()
        End If
    End Sub

    ' Method that clears the keyword list of the current instance
    Public Sub KeywordClear()
        cvKeywords.Clear()
        cvKeywords.TrimToSize()
    End Sub

    ' Method that removes the specified keyword from the keyword list of the current
instance
    Public Sub KeywordRemove(ByVal s As String)
        If cvKeywords.Contains(s.ToString.ToLower.Trim) Then
            cvKeywords.Remove(s.ToString.ToLower.Trim)
            cvKeywords.TrimToSize()
        End If
    End Sub
```

```
    ' Method that removes the specified keyword from the keyword list of the current
instance
    Public Sub KeywordRemoveAt(ByVal s As Integer)
        If s >= 1 AndAlso s <= cvKeywords.Count Then
            cvKeywords.RemoveAt(s - 1)
            cvKeywords.TrimToSize()
        End If
    End Sub

    ' Method that performs term reduction for the current instance
    Public Sub ReduceTerms(ByRef s As ArrayList)
        coVectorReduced = classVector.Reduce(coVectorFull, s)
        coVectorReduced.Normalize()
    End Sub
#End Region

#Region " Constructors "
    ' Default class constructor
    Public Sub New(ByVal s1 As String, _
                   Optional ByRef s2 As ArrayList = Nothing, _
                   Optional ByVal s3 As Boolean = True, _
                   Optional ByVal s4 As Integer = 0, _
                   Optional ByVal s5 As Integer = 0)

        If s1.Trim.Length = 0 Then
            MsgBox("A run-time exception was raised because of invalid argument" & _
                   vbNewLine & "supplied in classResult.New()", _
                   MsgBoxStyle.Critical + MsgBoxStyle.OKOnly, _
                   " Error")
            MyBase.Finalize()
        End If

        cvURL = s1.Trim
        If s2 Is Nothing Then : cvKeywords = New ArrayList
        Else : cvKeywords = New ArrayList(s2) : End If
        cvIsRelevant = s3
        coVectorFull = New classVector(s4)
        coVectorReduced = New classVector(s5)
    End Sub
#End Region

End Class

' This class implements the collection of "Query results"
Public NotInheritable Class classQuery

#Region " Instance members "
    ' The keywords' super-list for all results in the current instance
    Private cvKeywordsAll As SortedList

    ' The selected keywords' super-list for all results in the current instance
    Private cvKeywordsSelected As ArrayList

    ' The keywords' reduced super-list for all results in the current instance
    Private cvKeywordsReduced As ArrayList

    ' The keyword purity vector
    Private coPurityVector As classVector

    ' The list of results in the current instance
    Private cvResults As ArrayList

    ' The keywords' mean TF-IDF vector
    Private coMeansVector As classVector
#End Region


#Region " Properties "
    Public ReadOnly Property InformativeSelected() As Integer
        Get
            Dim lvResult As Integer
```

```
            For i As Integer = 1 To cvResults.Count
                If DirectCast(cvResults.Item(i - 1), classResult).IsRelevant AndAlso _
                   DirectCast(cvResults.Item(i - 1), classResult).IsInformative Then _
                          lvResult += 1
            Next

            Return lvResult
        End Get
    End Property
    ' Property that returns the number of informative results in the current instance
    Public ReadOnly Property Informative() As Integer
        Get
            Dim lvResult As Integer

            For i As Integer = 1 To cvResults.Count
                If DirectCast(cvResults.Item(i - 1), classResult).IsInformative Then _
                     lvResult += 1
            Next

            Return lvResult
        End Get
    End Property

    ' Property that returns the number of results marked as relevant in the current
instance
    Public ReadOnly Property Relevant() As Integer
        Get
            Dim lvResult As Integer

            For i As Integer = 1 To cvResults.Count
                If DirectCast(cvResults.Item(i - 1), classResult).IsRelevant Then _
                     lvResult += 1
            Next

            Return lvResult
        End Get
    End Property

    ' Property that returns the keywords' super-list
    Public ReadOnly Property KeywordsAll() As ArrayList
        Get
            Dim lvResult As New ArrayList

            For i As Integer = 1 To cvKeywordsAll.Count
                lvResult.Add(cvKeywordsAll.GetKey(i - 1).ToString)
            Next

            Return lvResult
        End Get
    End Property

    ' Property that returns the size of the keywords' super-list
    Public ReadOnly Property KeywordsAllCount() As Integer
        Get
            Return cvKeywordsAll.Count
        End Get
    End Property

    ' Property that returns the reduced keywords' super-list
    Public ReadOnly Property KeywordsReduced() As ArrayList
        Get
            Return cvKeywordsReduced
        End Get
    End Property


    ' Property that returns the size of the keywords' reduced super-list
    Public ReadOnly Property KeywordsReducedCount() As Integer
        Get
            Return cvKeywordsReduced.Count
```

```
        End Get
    End Property

    ' Property that returns the selected keywords' super-list
    Public ReadOnly Property KeywordsSelected() As ArrayList
        Get
            Return cvKeywordsSelected
        End Get
    End Property

    ' Property that returns the size of the selected keywords' super-list
    Public ReadOnly Property KeywordsSelectedCount() As Integer
        Get
            Return cvKeywordsSelected.Count
        End Get
    End Property

    ' Property that returns the keywords' TF-IDF means vector
    Public ReadOnly Property MeansVector() As classVector
        Get
            Return coMeansVector
        End Get
    End Property

    ' Property that sets/returns the term purity vector
    Public Property PurityVector() As classVector
        Get
            Return coPurityVector
        End Get
        Set(ByVal Value As classVector)
            coPurityVector = Value
        End Set
    End Property

    ' Property that returns the specified result of the current instance accessed by
index
    Public ReadOnly Property ResultByIndex(ByVal s As Integer) As classResult
        Get
            If s >= 1 AndAlso s <= cvResults.Count Then _
                Return DirectCast(cvResults.Item(s - 1), classResult)
            Return Nothing
        End Get
    End Property

    ' Property that returns the specified result of the current instance accessed by
index
    Public ReadOnly Property ResultByURL(ByVal s As String) As classResult
        Get
            For i As Integer = 1 To cvResults.Count
                If DirectCast(cvResults.Item(i - 1), classResult).URL = s.Trim Then _
                    Return DirectCast(cvResults.Item(i - 1), classResult)
            Next
            Return Nothing
        End Get
    End Property

    ' Property that returns the number of results in the current instance
    Public ReadOnly Property ResultCount() As Integer
        Get
            Return cvResults.Count
        End Get
    End Property
#End Region



#Region " Methods "
    ' Method that clears the list of results of the current instance
    Public Sub Clear()
        cvKeywordsAll.Clear()
```

```
        cvKeywordsAll.TrimToSize()
        cvKeywordsReduced.Clear()
        cvKeywordsReduced.TrimToSize()
        cvResults.Clear()
        cvResults.TrimToSize()
        coMeansVector.Length = 0
        coPurityVector = Nothing
    End Sub

    ' Method that adds the specified result to the current instance
    Public Sub ResultAdd(ByRef s As classResult)
        If Not cvResults.Contains(s) Then
            cvResults.Add(s)

            For i As Integer = 1 To s.KeywordCount
                If Not cvKeywordsAll.ContainsKey(s.KeywordByIndex(i).ToString) Then
                    cvKeywordsAll.Add(s.KeywordByIndex(i), 1)
                Else
                    cvKeywordsAll.Item(s.KeywordByIndex(i)) += 1
                End If
            Next

            cvKeywordsAll.TrimToSize()
            cvResults.TrimToSize()
        End If
    End Sub

    ' Method that removes the specified result from the current instance
    Public Sub ResultRemove(ByVal s As String)
        For i As Integer = 1 To cvResults.Count
            If DirectCast(cvResults.Item(i - 1), classResult).URL = s.Trim Then
                cvResults.RemoveAt(i - 1)
                cvResults.TrimToSize()
                Exit Sub
            End If
        Next
    End Sub

    ' Method that removes the specified result from the current instance
    Public Sub ResultRemoveAt(ByVal s As Integer)
        If s >= 1 AndAlso s <= cvResults.Count Then
            For j As Integer = 1 To DirectCast(cvResults.Item(s - 1), _
                                               classResult).KeywordCount
                If cvKeywordsAll.ContainsKey(DirectCast(cvResults.Item(s - 1), _
                   classResult).KeywordByIndex(j)) Then
                    If cvKeywordsAll.Item(DirectCast(cvResults.Item(s - 1), _
                       classResult).KeywordByIndex(j)) = 1 Then
                        cvKeywordsAll.Remove(DirectCast(cvResults.Item(s - 1), _
                          classResult).KeywordByIndex(j))
                    Else
                        cvKeywordsAll.Item(DirectCast(cvResults.Item(s - 1), _
                          classResult).KeywordByIndex(j)) -= 1
                    End If
                End If
            Next

            cvResults.RemoveAt(s - 1)
            cvResults.TrimToSize()
            cvKeywordsAll.TrimToSize()
        End If
    End Sub




    ' Method that returns membership information of the specified result to the
current instance
    Public Function ResultExists(ByVal s As String) As Boolean
        For i As Integer = 1 To cvResults.Count
            If DirectCast(cvResults.Item(i - 1), classResult).URL = s.Trim Then _
```

```
                Return True
        Next
        Return False
    End Function

    ' Method that returns membership information of the specified result to the
current instance
    Public Function ResultExists(ByRef s As classResult) As Boolean
        Return cvResults.Contains(s)
    End Function

    ' Method that removes specified keyword from selected keywords list
    Public Sub KeywordDeSelect(ByVal s1 As String)
        If cvKeywordsSelected.Contains(s1.Trim) Then
            cvKeywordsSelected.Remove(s1.Trim)
            cvKeywordsSelected.TrimToSize()
        End If
    End Sub

    ' Method that adds specified keyword to selected keywords list
    Public Sub KeywordSelect(ByVal s1 As String)
        If Not cvKeywordsSelected.Contains(s1.Trim) Then
            cvKeywordsSelected.Add(s1.Trim)
            cvKeywordsSelected.Sort()
            cvKeywordsSelected.TrimToSize()
        End If
    End Sub
#End Region

#Region " Constructors "
    ' Default class constructor
    Public Sub New()
        cvKeywordsAll = New SortedList
        cvKeywordsReduced = New ArrayList
        cvKeywordsSelected = New ArrayList
        cvResults = New ArrayList
        coMeansVector = New classVector
        coPurityVector = Nothing
    End Sub
#End Region

End Class

' This class implements the entity "Cluster"
Public NotInheritable Class classCluster

#Region " Instance members "
    ' The name-description of the cluster
    Private cvName As String

    ' The list of documents that belong to the current instance
    Private cvResultList As SortedList

    ' The centroid vector of the current instance
    Private coVectorCentroid As classVector
#End Region

#Region " Properties "
    ' Property that sets/returns the name of the current instance
    Public Property Name() As String
        Get
            Return cvName
        End Get
        Set(ByVal Value As String)
            cvName = Value
        End Set
    End Property

    ' Property that returns the specified document of the current instance accessed by
index
    Public ReadOnly Property ResultByIndex(ByVal s As Integer) As classResult
```

```
    Get
        If s >= 1 AndAlso s <= cvResultList.Count Then _
            Return DirectCast(cvResultList.GetByIndex(s - 1), classResult)
        Return Nothing
    End Get
End Property

' Property that returns the Entropy measure for the current instance
Public ReadOnly Property Entropy() As Single
    Get
       Dim lvCategories() As SortedList = {New SortedList, _
                                           New SortedList, _
                                           New SortedList}
       Dim lvPure() As Integer = {0, 0, 0}, lvResult() As Single = {0, 0, 0}
       Dim lvCategories1 As New SortedList
       Dim lvCategories2 As New SortedList
       Dim lvCategories3 As New SortedList
       Dim lvPure1, lvPure2, lvPure3 As Integer
       Dim lvResult1, lvResult2, lvResult3 As Single
       Dim hybrid As String

       For i As Integer = 1 To cvDocumentList.Count
            System.Windows.Forms.Application.DoEvents()

            With DirectCast(cvDocumentList.GetByIndex(i - 1), classDocument)
                 If .CategoryCount >= 1 AndAlso .CategoryCount <= 3 Then
                      lvPure(.CategoryCount - 1) += 1
                      hybrid = String.Empty
                      For j As Integer = 1 To .CategoryCount
                           hybrid &= .CategoryByIndex(j)
                           If j < .CategoryCount Then hybrid &= "-"
                      Next
                      If lvCategories(.CategoryCount - 1).ContainsKey(hybrid) Then
                           lvCategories(.CategoryCount - 1).Item(hybrid) += 1
                      Else
                           lvCategories(.CategoryCount - 1).Add(hybrid, 1)
                      End If
                 End If
            End With
       Next

       For i As Integer = 1 To 3
            If lvPure(i - 1) > 0 Then
                 For j As Integer = 1 To lvCategories(i - 1).Count
                      System.Windows.Forms.Application.DoEvents()

                      lvResult(i - 1) += CInt(lvCategories(i-1).GetByIndex(j-1)) _
                           /(lvPure(0) + lvPure(1) + lvPure(2)) * _
                          Math.Log10(CInt(lvCategories(i-1).GetByIndex(j-1)) / _
                           (lvPure(0) + lvPure(1) + lvPure(2)))
                 Next
            End If
       Next

       Return (-1) * (lvResult(0) + lvResult(1) + lvResult(2)) / _
                 Math.Log10(poDataset.CategoriesCount + 45 + 120)
    End Get
End Property

' Property that returns the Purity measure for the current instance
Public ReadOnly Property Purity() As Single
    Get
       Dim lvResult As Single                  ' Stores property result
       Dim lvPureDocuments As Integer          ' Counter
       Dim lvLabelCounter As New SortedList

       For i As Integer = 1 To cvResultList.Count
            If cvResultList.GetByIndex(i - 1).CategoryCount = 1 Then
                 lvPureDocuments += 1
                 If lvLabelCounter.ContainsKey( _
                      cvDocumentList.GetByIndex(i-1).CategoryByIndex(1)) Then
```

```
                        lvLabelCounter.Item( _
                          cvDocumentList.GetByIndex(i-1).CategoryByIndex(1)) += 1
                    Else
                         lvLabelCounter.Add( _
                          cvDocumentList.GetByIndex(i-1).CategoryByIndex(1), 1)
                    End If
                End If
            Next

            If lvPureDocuments > 0 Then
                For i As Integer = 1 To lvLabelCounter.Count
                  System.Windows.Forms.Application.DoEvents()

                  lvResult = Math.Max(lvResult, CInt(lvLabelCounter.GetByIndex(i-1)))
                Next
                Return (lvResult / lvPureDocuments)
            Else
                Return 0
            End If
        End Get
    End Property

    ' Property that returns the internal similarity measure for the current instance
    Public ReadOnly Property Similarity() As Single
        Get
            Dim lvResult As Single                  ' Stores property result
            Dim loVectorA, loVectorB As classVector ' Temporary buffers

            If cvResultList.Count > 1 Then
                For i As Integer = 1 To (cvResultList.Count - 1)
                    For j As Integer = (i + 1) To cvResultList.Count
                        ' Creates copies of original vectors
                        loVectorA = New classVector( _
                                    DirectCast(cvResultList.GetByIndex(i - 1), _
                                    classResult).VectorReduced)
                        loVectorB = New classVector( _
                                    DirectCast(cvResultList.GetByIndex(j - 1), _
                                    classResult).VectorReduced)

                        ' Calculates distance between current vectors
                        If loVectorA.IsZero And loVectorB.IsZero Then _
                          lvResult += 0
                        If loVectorA.IsZero And Not loVectorB.IsZero Then _
                          lvResult += loVectorB.Norm(2)
                        If Not loVectorA.IsZero And loVectorB.IsZero Then _
                          lvResult += loVectorA.Norm(2)
                        If Not loVectorA.IsZero And Not loVectorB.IsZero Then _
                          loVectorB.Multiply(-1)
                           lvResult += classVector.Norm(classVector.Add(loVectorA, _
                                          loVectorB), 2)
                        End If
                    Next
                Next

                ' Returns the percentage of the calculated distances over the maximum
potential distances
                Return 1 - lvResult / (Math.Sqrt(2) * cvResultList.Count * _
                           (cvResultList.Count - 1) / 2)
            Else
                Return 1
            End If
        End Get
    End Property


    ' Property that returns the number of documents that belong to the current
instance
    Public ReadOnly Property Size() As Integer
        Get
            Return cvResultList.Count
        End Get
```

```
    End Property

    ' Property that returns the centroid vector of the current instance
    Public ReadOnly Property VectorCentroid() As classVector
        Get
            Return coVectorCentroid
        End Get
    End Property
#End Region

#Region " Methods "
    ' Method that calculated the centroid vector of the current instance
    Public Sub CalculateCentroid()
        coVectorCentroid.Clear()
        coVectorCentroid.Length = poQuery.KeywordsReducedCount

        For i As Integer = 1 To cvResultList.Count
            coVectorCentroid.Add(DirectCast(cvResultList.GetByIndex(i - 1), _
                classResult).VectorReduced)
            If i = cvResultList.Count Then _
                coVectorCentroid.Multiply(1 / cvResultList.Count)
        Next
    End Sub

    ' Method that adds the specified result to the current instance
    Public Sub ResultAdd(ByRef s As classResult)
        If Not cvResultList.ContainsValue(s) Then
            cvResultList.Add(s.URL, s)
            cvResultList.TrimToSize()
        End If
    End Sub

    ' Method that removes the specified document to the current instance
    Public Sub ResultRemove(ByRef s As classResult)
        If cvResultList.ContainsValue(s) Then
            cvResultList.Remove(s.URL)
            cvResultList.TrimToSize()
        End If
    End Sub

    ' Method that searches for the specified document in current instance
    Public Function ResultExists(ByVal s As String) As Boolean
        Return cvResultList.ContainsKey(s.Trim)
    End Function

    ' Method that searches for the specified document in current instance
    Public Function ResultExists(ByRef s As classResult) As Boolean
        Return cvResultList.ContainsValue(s)
    End Function

    ' Method that clears all documents of current instance
    Public Sub Clear()
        cvResultList.Clear()
        cvResultList.TrimToSize()
    End Sub
#End Region
#Region " Constructors "
    ' Default class constructor
    Public Sub New(ByVal s As String)
        cvName = s
        cvResultList = New SortedList
        coVectorCentroid = New classVector
    End Sub
#End Region
End Class

' This class implements the collection of "Clusters"
Public NotInheritable Class classScheme

#Region " Instance members "
    ' The list of clusters which constitute current clustering scheme
```

```
    Private pvClusters As ArrayList
#End Region

#Region " Properties "
    ' Property that returns a specified cluster of the current instance accessed by
index
    Public ReadOnly Property ClusterByIndex(ByVal s As Integer) As classCluster
        Get
            If s >= 1 AndAlso s <= pvClusters.Count Then _
                Return pvClusters.Item(s - 1)
            Return Nothing
        End Get
    End Property

    ' Property that returns the number of clusters of the current instance
    Public ReadOnly Property Size() As Integer
        Get
            Return pvClusters.Count
        End Get
    End Property

    ' Property that returns the Entropy measure for the current clustering scheme
instance
    Public ReadOnly Property Entropy() As Single
        Get
            Dim lvResult As Single                  ' Stores property result

            For i As Integer = 1 To pvClusters.Count
                lvResult += ClusterByIndex(i).Size / poQuery.ResultCount * _
                                ClusterByIndex(i).Entropy
            Next

            Return lvResult
        End Get
    End Property

    ' Property that returns the Purity measure for the current clustering scheme
instance
    Public ReadOnly Property Purity() As Single
        Get
            Dim lvResult As Single                  ' Stores property result

            For i As Integer = 1 To pvClusters.Count
                lvResult += ClusterByIndex(i).Size / poQuery.ResultCount * _
                                ClusterByIndex(i).Purity
            Next

            Return lvResult
        End Get
    End Property

    ' Property that returns the Internal Similarity measure for the current clustering
scheme instance
    Public ReadOnly Property Similarity() As Single
        Get
            Dim lvResult As Single                  ' Stores property result

            For i As Integer = 1 To pvClusters.Count
                lvResult += ClusterByIndex(i).Size / poQuery.ResultCount * _
                                ClusterByIndex(i).Similarity
            Next

            Return lvResult
        End Get
    End Property
#End Region
#Region " Methods "
    ' Method that creates the specified number of clusters for the current instance by
random selection of input documents
    Public Sub Create(ByVal s As Integer)
        Dim lvNewCluster As classCluster            ' Pointer to the new cluster
```

```
        Randomize()
        pvClusters.Clear()

        For i As Integer = 1 To s
            lvNewCluster = New classCluster("Cluster " & i.ToString)
            lvNewCluster.VectorCentroid.Length = poQuery.KeywordsReducedCount

            For j As Integer = 1 To lvNewCluster.VectorCentroid.Length
                lvNewCluster.VectorCentroid.Item(j) = (i - 1) / (s - 1)
            Next

            pvClusters.Add(lvNewCluster)
        Next

        pvClusters.TrimToSize()
    End Sub

    ' Method that clears the list of clusters of the current instance
    Public Sub Clear()
        pvClusters.Clear()
        pvClusters.TrimToSize()
    End Sub

    ' Method that adds the specified cluster to the list of clusters of the current
instance
    Public Sub Add(ByRef s As classCluster)
        pvClusters.Add(s)
        pvClusters.TrimToSize()
    End Sub

    ' Method that re-assigns the specified result to the specified cluster of the
current instance
    Public Sub MoveDocument(ByRef s1 As classResult, ByRef s2 As classCluster)
        For i As Integer = 1 To pvClusters.Count
            If DirectCast(pvClusters.Item(i - 1), classCluster).ResultExists(s1) _
                AndAlso Not Equals(DirectCast(pvClusters.Item(i-1),classCluster),s2) _
                Then
                     DirectCast(pvClusters.Item(i - 1), classCluster).ResultRemove(s1)
                Exit For
            End If
        Next

        For i As Integer = 1 To pvClusters.Count
            If Equals(DirectCast(pvClusters.Item(i - 1), classCluster), s2) AndAlso _
                Not DirectCast(pvClusters.Item(i-1), classCluster).ResultExists(s1) _
                Then
                     DirectCast(pvClusters.Item(i - 1), classCluster).ResultAdd(s1)
                Exit For
            End If
        Next
    End Sub
#End Region

#Region " Constructors "
    ' Default class constructor
    Public Sub New()
        pvClusters = New ArrayList
    End Sub
#End Region

End Class



' This single-instance class encapsulates all major variables used in the application
'   and facilitates manipulation of their values in a centralized, more efficient,
'   not error-prone manner
Public NotInheritable Class classKernel
```

```
#Region " Data members "
    ' Enumerates the various application states
    Public Enum enumApplicationState
        Idle = 1
        Processing = 2
    End Enum

    ' A flag that provides information whether a query has been executed
    Private cvQueryPerformed As Boolean

    ' A flag that provides information whether vector representations have been
created
    Private cvVectorsCreated As Boolean

    ' A flag that signals current action cancellation request
    Private cvCancelRequested As Boolean

    ' The current state of the application
    Private cvApplicationState As enumApplicationState

    ' A flag that provides information whether clustering has been performed
    Private cvClusteringPerformed As Boolean

    ' A flag that provides information whether clustering summary has been created
    Private cvSummaryCreated As Boolean

    ' A flag that provides information whether term reduction has been performed
    Private cvKeywordReductionPerformed As Boolean

    ' A flag that provides information whether term weighting has been performed
    Private cvKeywordWeightingPerformed As Boolean
#End Region

#Region " Properties "
    Public Property QueryPerformed() As Boolean
        Get
            Return cvQueryPerformed
        End Get
        Set(ByVal Value As Boolean)
            cvQueryPerformed = Value

            If Not cvQueryPerformed Then
                poQuery.Clear()
                Me.VectorsCreated = False
            End If
        End Set
    End Property

    ' Property that sets/returns information on whether current action cancelling has
been requested
    Public Property CancelRequested() As Boolean
        Get
            Return cvCancelRequested
        End Get
        Set(ByVal Value As Boolean)
            cvCancelRequested = Value
        End Set
    End Property

    ' Property that sets/returns information on whether clustering has been performed
    Public Property ClusteringPerformed() As Boolean
        Get
            Return cvClusteringPerformed
        End Get
        Set(ByVal Value As Boolean)
            cvClusteringPerformed = Value

            If Not cvClusteringPerformed Then
                poScheme.Clear()
                cvSummaryCreated = False
            End If
```

```
        End Set
    End Property

    ' Property that sets/returns information on whether datafile has been read
    Public Property VectorsCreated() As Boolean
        Get
            Return cvVectorsCreated
        End Get
        Set(ByVal Value As Boolean)
            cvVectorsCreated = Value

            If Not cvVectorsCreated Then Me.TermReductionPerformed = False
        End Set
    End Property

    ' Property that sets/returns information on whether summary has been created
    Public Property SummaryPerformed() As Boolean
        Get
            Return cvSummaryCreated
        End Get
        Set(ByVal Value As Boolean)
            cvSummaryCreated = Value
        End Set
    End Property

    ' Property that sets/returns information on whether term reduction has been
performed
    Public Property TermReductionPerformed() As Boolean
        Get
            Return cvKeywordReductionPerformed
        End Get
        Set(ByVal Value As Boolean)
            cvKeywordReductionPerformed = Value

            If Not cvKeywordReductionPerformed Then
                poQuery.KeywordsReduced.Clear()
                Me.TermWeightingPerformed = False
            End If
        End Set
    End Property

    ' Property that sets/returns information on whether term weighting has been
performed
    Public Property TermWeightingPerformed() As Boolean
        Get
            Return cvKeywordWeightingPerformed
        End Get
        Set(ByVal Value As Boolean)
            cvKeywordWeightingPerformed = Value

            If Not cvKeywordWeightingPerformed Then
                poQuery.PurityVector = Nothing
                Me.ClusteringPerformed = False
            End If
        End Set
    End Property

    ' Property that sets/returns information on current application state
    Public Property ApplicationState() As enumApplicationState
        Get
            Return cvApplicationState
        End Get
        Set(ByVal Value As enumApplicationState)
            cvApplicationState = Value
        End Set
    End Property
#End Region

#Region " Constructors "
    ' Default class constructor
    Public Sub New()
```

```
        cvApplicationState = enumApplicationState.Idle
        cvCancelRequested = False
        cvVectorsCreated = False
        cvSummaryCreated = False
        cvClusteringPerformed = False
        cvKeywordReductionPerformed = False
        cvKeywordWeightingPerformed = False
    End Sub
#End Region

End Class
```