ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ

ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Μέθοδοι Εξόρυξης Κειμένου για Ομαδοποίηση Ιδεών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΣΤΥΛΙΑΝΗΣ ΠΑΧΙΔΗ

Επιβλέπων : Γρηγόριος Μέντζας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2008

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΑΠΟΦΑΣΕΩΝ

Μέθοδοι Εξόρυξης Κειμένου για Ομαδοποίηση Ιδεών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΣΤΥΛΙΑΝΗΣ ΠΑΧΙΔΗ

Επιβλέπων : Γρηγόριος Μέντζας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Ιουλίου 2008.

.....	.....	.....
Δημήτριος Ασκούνης	Γρηγόριος Μέντζας	Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2008

.....

ΣΤΥΛΙΑΝΗ ΠΑΧΙΔΗ

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη μεθόδων του τομέα της εξόρυξης κειμένου καθώς και της επεξεργασίας φυσικής γλώσσας, προκειμένου να υλοποιηθεί ένα εργαλείο ομαδοποίησης αρχείων κειμένου. Η ανάγκη για την υλοποίηση ενός τέτοιου εργαλείου προέκυψε από την ανάγκη για ομαδοποίηση των ιδεών που εισάγουν οι χρήστες στο IDeM, ένα σύστημα διαχείρισης ιδεών με τη χρήση προγνωστικών αγορών.

Το σύστημα της ομαδοποίησης κειμένου υλοποιήθηκε πάνω στον κώδικα του Weka, ενός εργαλείου εξόρυξης γνώσης από δεδομένα ανοιχτού κώδικα, υλοποιημένο σε Java. Στη διαδικασία της ομαδοποίησης, όπως αυτή ορίζεται στις τεχνικές εξόρυξης κειμένου (αναπαράσταση των αρχείων κειμένου, ορισμός μέτρου ομοιότητας, εφαρμογή αλγορίθμου ομαδοποίησης, προσδιορισμός και εκτίμηση του αποτελέσματος), προστέθηκαν λειτουργίες βασισμένες σε μεθόδους της επεξεργασίας φυσικής γλώσσας για τη βελτίωση της αποτελεσματικότητας της ομαδοποίησης: γλωσσική επεξεργασία, αφαίρεση stopwords, εύρεση της ρίζας των λέξεων, επισημείωση των μερών του λόγου, αποσαφήνιση της έννοιας των λέξεων, εύρεση και συγχώνευση συνώνυμων όρων.

Το σύστημα που αναπτύχθηκε ενσωματώθηκε στην αρχιτεκτονική του IDeM, ώστε να μπορούν οι χρήστες να εκτελούν την ομαδοποίηση ιδεών ως μία λειτουργία του συστήματος. Επιπλέον, το σύστημα μπορεί να λειτουργήσει και αυτόνομα, γεγονός το οποίο μας βοήθησε στη διεξαγωγή ελέγχων για την αξιολόγηση του συστήματος και τη μέτρηση της ακρίβειας των αποτελεσμάτων ομαδοποίησης, καθώς και τη σύγκρισή τους με άλλες υλοποιήσεις.

Λέξεις Κλειδιά: Εξόρυξη κειμένου, Επεξεργασία Φυσικής Γλώσσας,, Ομαδοποίηση αρχείων κειμένου, Διαχείριση Ιδεών

Abstract

The scope of this thesis was the implementation of text mining methods as also natural language processing techniques, in order to develop a tool for text clustering. The need for such tool came from the urge for clustering users' ideas in IDeM, an idea management system with the use of information aggregation markets.

The text clustering system was developed based on Weka, an open source tool for data mining. The techniques of text clustering, as they are defined in the field of text mining (representation of text documents, defining a similarity measure, implementation of a clustering algorithm, assessment of the output) were expanded by natural language processing techniques, aiming to improve the clustering efficiency and effectiveness: tokenization, stopwords removal, stemming, part-of-speech tagging, word sense disambiguation, finding synonyms.

The text clustering system was integrated in IDeM architecture, in order to add a new function that could be used by IDeM users for clustering ideas. Furthermore, the possibility to use the system as an autonomous module was very useful in the performance of tests with the use of pre-categorized corpora, in an attempt to test the system's accuracy and compare it to other implementations.

Keywords: Text mining. Natural Language Processing, Text clustering, Idea Management

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Γρηγόριο Μέντζα, για την υποστήριξή του στην εκπόνηση της διπλωματικής εργασίας. Η πολύτιμη καθοδήγησή του, ο χρόνος που αφιέρωσε και η άριστη συνεργασία που είχαμε συνέβαλαν σημαντικά στην εκπλήρωση αυτής της εργασίας. Επίσης, ευχαριστώ τον υποψήφιο διδάκτορα Ευθύμιο Μπόθο για την ουσιαστική βοήθεια που προσέφερε στη διεξαγωγή της εργασίας, καθώς και τον λέκτορα κ. Δημήτριο Αποστόλου για τη συνεργασία που είχαμε και την υποστήριξή του στα πλαίσια της διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για την πολύτιμη βοήθεια και συμπαράσταση όλα αυτά τα χρόνια.

Πίνακας περιεχομένων

1 Εισαγωγή

1.1 Η χρήση μεθόδων ομαδοποίησης κειμένου και επεξεργασίας φυσικής γλώσσας για την ομαδοποίηση ιδεών

Σε μια εποχή όπου καθημερινά δεχόμαστε πλήθος ερεθισμάτων από διαφορετικές πηγές πληροφόρησης, όπου η ανταλλαγή πληροφοριών αποτελεί λειτουργία υψίστης σημασίας σε όλους τους τομείς, ο τομέας της εξόρυξης γνώσης από κείμενα γνωρίζει τεράστια ανάπτυξη. Στον τομέα αυτό υπάγεται και η ομαδοποίηση αρχείων κειμένου, η προσπάθεια δηλαδή διαχωρισμού τους σε ομάδες με όμοιο ή παρόμοιο νοηματικό περιεχόμενο. Στο πλαίσιο αυτό θα λέγαμε ότι τοποθετούμε το έργο της διπλωματικής εργασίας.

1.2 Αντικείμενο διπλωματικής εργασίας

Στην εργασία αυτή ασχοληθήκαμε με την ανάπτυξη ενός εργαλείου ομαδοποίησης αρχείων κειμένου. Η ανάγκη για το εργαλείο προέκυψε ως ανάγκη για την ομαδοποίηση των ιδεών που τοποθετούν οι χρήστες σε ένα σύστημα διαχείρισης καινοτομίας με τη χρήση προγνωστικών αγορών, το IDeM.

Στην εργασία αυτή λοιπόν, μελετήσαμε τις μεθόδους εξόρυξης γνώσης από κείμενα που ακολουθούνται για την ομαδοποίηση αρχείων κειμένου. Στη συνέχεια, προκειμένου να βελτιώσουμε την αποτελεσματικότητα της ομαδοποίησης, εφόσον μελετάμε τη νοηματική ομοιότητα μεταξύ των κειμένων, θελήσαμε να εμβαθύνουμε περισσότερο στη νοηματική ανάλυση των κειμένων, οπότε και μελετήσαμε πώς μπορούμε να χρησιμοποιήσουμε

μεθόδους της επεξεργασίας φυσικής γλώσσας για την βελτιστοποίηση της διαδικασίας της ομαδοποίησης.

Με βάση τις μεθόδους που μελετήσαμε, καταλήξαμε στην επιλογή συγκεκριμένων μεθοδολογιών, τεχνικών, αλγορίθμων και εργαλείων για την ανάπτυξη ενός συστήματος ομαδοποίησης αρχείων κειμένου. Το σύστημα αυτό υλοποιήθηκε έτσι ώστε και να μπορεί να επιτελεί τη λειτουργία της ομαδοποίησης αυτόνομα, αλλά επίσης ολοκληρώθηκε και η ενσωμάτωσή του με το IDeM ώστε να επιτελείται η λειτουργία της ομαδοποίησης των ιδεών.

Για την αξιολόγηση της αποδοτικότητας και της αποτελεσματικότητας του συστήματος έγιναν διάφοροι έλεγχοι ομαδοποίησης αρχείων κειμένου χρησιμοποιώντας ως δεδομένα σύνολα κειμένων ήδη κατηγοριοποιημένων, ώστε να μετρηθεί η ακρίβεια των αποτελεσμάτων της ομαδοποίησης.

1.3 Διάρθρωση της εργασίας

Το υπόλοιπο της εργασίας οργανώνεται ως ακολούθως: στο κεφάλαιο 2 μελετάμε το αντικείμενο του συστήματος IDeM, προκειμένου να καταλάβουμε πώς προέκυψε η ανάγκη για ομαδοποίηση ιδεών. Το κεφάλαιο 3 αποτελεί ουσιαστικά το state of the art της διπλωματικής εργασίας: μελετούνται οι μέθοδοι και τεχνικές εξόρυξης κειμένου και επεξεργασίας φυσικής γλώσσας. Έπειτα, στο κεφάλαιο 4 παρουσιάζονται οι τεχνικές, οι αλγόριθμοι και τα εργαλεία που επιλέχθηκαν για την υλοποίηση του συστήματος ομαδοποίησης αρχείων κειμένου, καθώς και πώς αυτό ανταποκρίνεται στις λειτουργικές απαιτήσεις του IDeM και πώς ενσωματώνεται στην αρχιτεκτονική του. Στο κεφάλαιο 5 περιγράφονται οι έλεγχοι που δημιουργήθηκαν για την αξιολόγηση του συστήματος, ενώ η εργασία ολοκληρώνεται με το κεφάλαιο 6, στο οποίο γίνεται μια σύνοψη και αναφέρονται συμπεράσματα και προοπτικές επέκτασης.

2 Ανάγκη για Ημιαυτόματη Ομαδοποίηση Ιδεών

Ο σκοπός της δημιουργίας ενός συστήματος ημιαυτόματης ομαδοποίησης ιδεών προέκυψε από τις ανάγκες του IDeM, ενός συστήματος διαχείρισης ιδεών με τη χρήση Προγνωστικών Αγορών. Στο κεφάλαιο αυτό θα μελετήσουμε συνοπτικά τι είναι οι Προγνωστικές Αγορές και πώς άρχισαν να χρησιμοποιούνται ως εργαλεία για τη διαχείριση ιδεών μέσα σε εταιρείες και οργανισμούς, ενώ επίσης θα δούμε πώς λειτουργεί το IDeM και πώς προέκυψε η ανάγκη για ημιαυτόματη ομαδοποίηση ιδεών.

2.1 Η χρήση των Προγνωστικών Αγορών για τη Διαχείριση Ιδεών

Η συνεχής εξέλιξη των νέων ηλεκτρονικών τεχνολογιών πληροφορίας και επικοινωνίας (Information and Communications Technologies), κατάφερε να οδηγήσει σε δραστικές αλλαγές, επηρεάζοντας τις διαπροσωπικές σχέσεις και τις επικοινωνίες σε πολλούς τομείς της κοινωνικής, πολιτισμικής, οικονομικής και πολιτικής ζωής. Σήμερα, οι χρήστες έχουν τη δυνατότητα μέσω της πρόσβασης στο διαδίκτυο να ανακτούν αλλά και να ανταλλάσσουν πληροφορίες. Η ανάπτυξη online κοινοτήτων, σε συνδυασμό με τις τεχνολογίες Web 2.0 που με ταχύ ρυθμό έχουν αρχίσει να κατακλύζουν τον Παγκόσμιο Ιστό, επιτρέπουν στους χρήστες να χρησιμοποιούν υπηρεσίες και εφαρμογές για την ανταλλαγή γνώσης και ιδεών. Ανάμεσα στις διάφορες εφαρμογές οι οποίες, με χρήση της Web 2.0 τεχνολογίας προωθούν

την ιδέα της "συλλογικής νοημοσύνης", εντοπίζουμε τις Προγνωστικές Αγορές (Prediction Markets) ή αλλιώς Αγορές Συνάθροισης Πληροφοριών (Information Aggregation Markets), οι οποίες αποτελούν "εικονικές χρηματαγορές" (virtual stock markets) με σκοπό τη συλλογή και συνάθροιση πληροφοριών. Οι συμμετέχοντες σε μια τέτοια αγορά συναλλάσσονται μετοχές οι οποίες αναπαριστούν διαφορετικές εκβάσεις ενός μελλοντικού γεγονότος. Στο κλείσιμο της αγοράς, η τιμή αυτών των εικονικών μετοχών ενσωματώνει τις διαθέσιμες πληροφορίες σχετικά με το γεγονός.

Σύμφωνα με τον Eugene Fama και τη θεμελιώδη υπόθεση αποτελεσματικής αγοράς (efficient-market hypothesis) που εξέφρασε το 1965, όταν μια αποτελεσματική αγορά φτάνει σε ισορροπία, τότε περικλείει όλη τη διαθέσιμη πληροφορία. Παρατηρώντας δηλαδή τις τιμές της αγοράς μπορούμε να αποκτήσουμε μια αρκετά ακριβή εκτίμηση για το μέλλον. Έχοντας ως βάση την ικανότητα των αγορών να αθροίζουν πληροφορίες, και μάλιστα μέσω του όγκου των συναλλαγών τους να αποδίδουν τη βέλτιστη εκτίμηση για το δεδομένο όγκο πληροφορίας (όταν λειτουργούν αποτελεσματικά), αναπτύχθηκαν οι Προγνωστικές Αγορές.

Ένας ορισμός που δίνεται από τον καθηγητή Τζιραλή είναι:

«Ορίζουμε ως προγνωστική αγορά (prediction market) την αγορά εκείνη που δημιουργείται και λειτουργεί με πρώτιστο σκοπό την εξόρυξη και άθροιση διάσπαρτων ανά τους παίκτες πληροφοριών και τη μετέπειτα χρήση του πληροφοριακού της περιεχομένου, όπως αυτό απεικονίζεται στις τιμές αγοράς κατάλληλα διαμορφωμένων αξιογράφων, για την εκπόνηση προγνώσεων συσχετιζόμενων με συγκεκριμένα μελλοντικά γεγονότα.»

Η ιδέα των αγορών ως εργαλεία για προγνώσεις υπάρχει εδώ και αρκετά χρόνια, ωστόσο έχει αρχίσει να αναπτύσσεται τα τελευταία 5 χρόνια.

Τα Information Aggregation Markets έχουν χρησιμοποιηθεί από πολλές εταιρείες για τη στήριξη αποφάσεων, για παράδειγμα ποιες ημερομηνίες θα λανσάρουν ένα προϊόν για να επιτύχουν μεγαλύτερες μελλοντικές πωλήσεις, έχοντας επιτυχή αποτελέσματα στην πλειοψηφία των περιπτώσεων. Έτσι, άρχισε να στρέφεται το ερευνητικό ενδιαφέρον στη χρήση των Information Aggregation Markets για τη διαχείριση των ιδεών σε μια εταιρεία.

Νέες ιδέες μπορεί να προέρχονται είτε από το εσωτερικό της εταιρείας (για παράδειγμα υπαλλήλους, managers, άλλα μέλη του προσωπικού) ή και έξω από την εταιρεία (lead users, πελάτες, το κοινό γενικά). Η διαδικασία της διαχείρισης των ιδεών (idea management) περιλαμβάνει τη δημιουργία, τη συλλογή, την ανάπτυξη, την αξιολόγηση και την επιλογή νέων ιδεών. Το στάδιο της αξιολόγησης στη διαδικασία της διαχείρισης ιδεών περιλαμβάνει την απόφαση για το ποιες είναι οι καλύτερες ιδέες και την επιλογή εκείνων που θα επιφέρουν

στην εταιρεία αυξημένα κέρδη και αύξηση του αριθμού των πελατών της. Η αξιολόγηση των ιδεών αποτελεί μια δύσκολη δουλειά επειδή περιλαμβάνει την μετατροπή ενός μεγάλου αριθμού από ιδέες με διαφορετική ποιότητα, ωριμότητα και βαθμό περιπλοκότητας σε εναλλακτικές από τις οποίες πρέπει να γίνει κάποια επιλογή. Μια μεθοδολογία που ξεκίνησε να αναπτύσσεται λοιπόν ήταν η χρήση των Prediction Markets, εικονικών χρηματαγορών που μπορούσαν να διαμορφωθούν έτσι ώστε να εξυπηρετούν την πρόβλεψη της εξέλιξης των ιδεών σε μια εταιρεία. Μια αναφορά του Gartner τοποθετεί τα Information Aggregation Markets ως μια από τις ανερχόμενες μεθόδους για δημιουργία κοινοτήτων μέσα σε εταιρείες, η οποία εκμεταλλεύεται την συλλογική νοημοσύνη των χρηστών που παράγεται από μικρές συνεισφορές κάθε χρήστη μιας μεγάλης κοινότητας.

Η μέχρι τώρα εφαρμογή της χρήσης των Information Aggregation Markets στο πλαίσιο της διαχείρισης καινοτομίας, έχει δείξει ότι οι αγορές αυτές ακολουθούν τους περιορισμούς της διαδικασίας της αξιολόγησης ιδεών ενώ επίσης με τον κατάλληλο σχεδιασμό μπορούν να στηρίξουν και τη φάση της παραγωγής ιδεών του idea management. Η έρευνα ως τώρα έχει δείξει ότι η ποιότητα των νέων ιδεών επηρεάζεται θετικά από τη συμμετοχή πολλών υπαλλήλων. Μάλιστα, η χρήση ομάδων από διαφορετικό εργασιακό και πνευματικό υπόβαθρο, καθώς προκαλεί την ανταλλαγή περισσότερων και διαφορετικών πληροφοριών, δημιουργεί τη δυνατότητα δημιουργίας και συνεπώς αξιολόγησης διαφορετικών ιδεών από διαφορετικές οπτικές γωνίες.

Δύο μεγάλα πειράματα που έχουν διεξαχθεί σε μεγάλες εταιρείες (ένα στη General Electric κι ένα στη γερμανική εταιρεία B2B), ήταν επιτυχή και έδειξαν ενθαρρυντικά αποτελέσματα για τη χρήση Prediction Markets στη διαχείριση ιδεών, ελκύοντας έναν αρκετά μεγάλο αριθμό συμμετεχόντων από διαφορετικά τμήματα, με αποτέλεσμα την ενσωμάτωση της γνώμης των πολλών στο τελικό αποτέλεσμα.

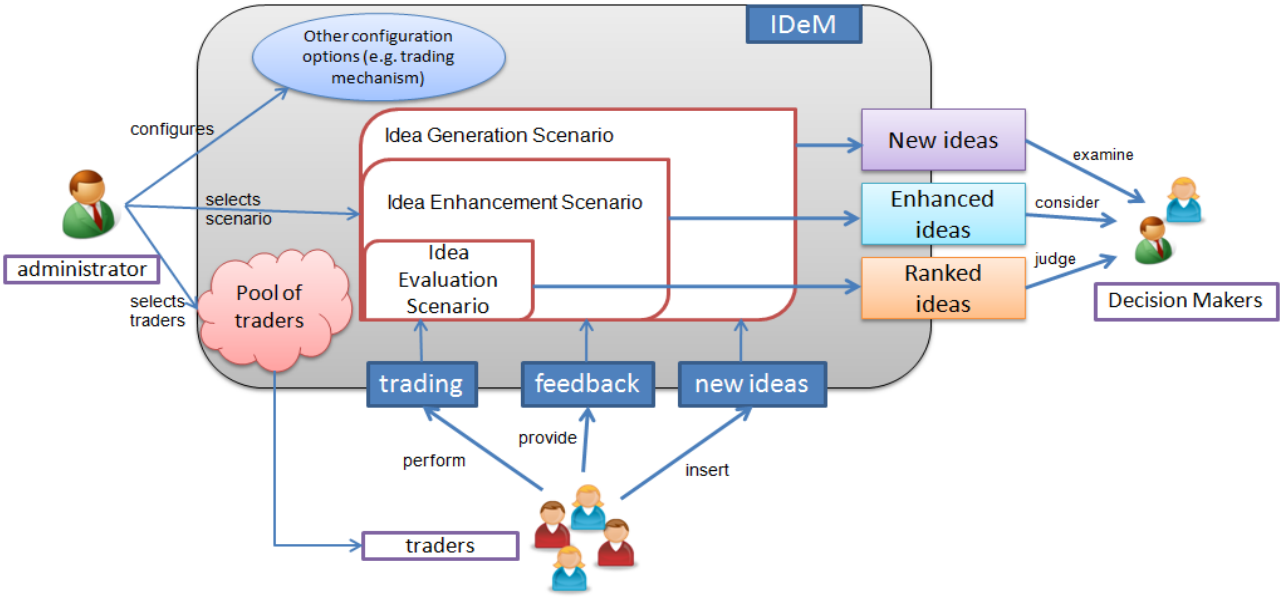
Σήμερα έχουν αρχίσει να αναπτύσσονται διάφορες πλατφόρμες για τη χρήση των Information Aggregation Markets στη διαχείριση ιδεών, από τις οποίες άλλες πωλούνται, άλλες διατίθενται δωρεάν και άλλες είναι ανοιχτού κώδικα. Ορισμένες γνωστές πλατφόρμες είναι το Hollywood Stock eXchange (www.hsx.com), ConsensusPont (www.consensuspont.com), Inklingmarkets (www.inklingmarkets.com), Zocalo, κα.

Στην επόμενη ενότητα θα μελετήσουμε ένα σύστημα διαχείρισης ιδεών με τη χρήση prediction market που αναπτύχθηκε από την Information Management Unit του ΕΜΠ, το IDeM.

2.2 Ένα σύστημα διαχείρισης ιδεών: IDeM

Το IDeM είναι ένα σύστημα λογισμικού που ενσωματώνει χαρακτηριστικά της χρήσης των Information Aggregation Markets για τη διαχείριση ιδεών. Ο αρχικός λόγος δημιουργίας αυτής της πλατφόρμας ήταν η υποστήριξη επιπλέον λειτουργιών, χρήσιμων για τη διαχείριση των ιδεών και αρκετά σημαντικότερων από την απλή συνάθροιση πληροφοριών για μελλοντικά γεγονότα (που δεν είναι ο άμεσος στόχος της διαχείρισης ιδεών), τις οποίες δεν προσέφεραν ως τότε άλλες πλατφόρμες σχετικές με την αξιολόγηση ιδεών με τη χρήση Prediction Markets. Τέτοιες λειτουργίες είναι η υποστήριξη feedback από τους χρήστες και η εξαγωγή επιπλέον πληροφοριών πέρα από τις τιμές της αγοράς όπως οι δοσοληψίες που έγιναν και η συμμετοχή των χρηστών.

Στην παρακάτω εικόνα μπορούμε να δούμε την οργάνωση των λειτουργιών του IDeM:



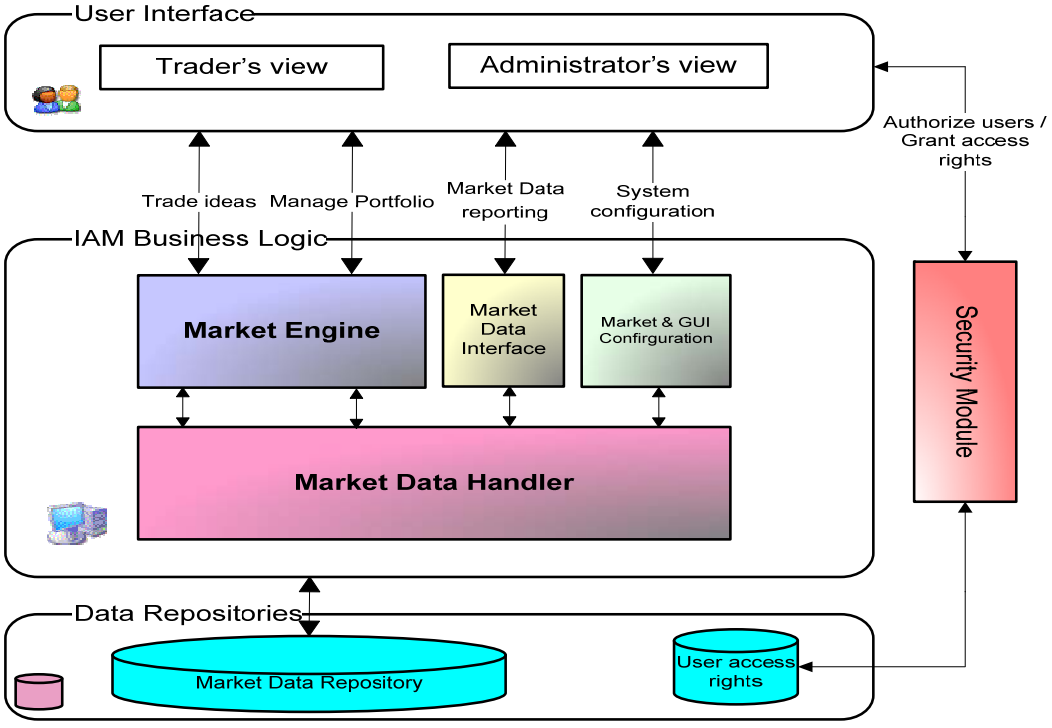
Εικόνα 2-1 : Οι λειτουργίες του IDeM

Το IDeM μπορεί να χρησιμοποιηθεί σε αρκετά σενάρια χρήσης μέσα στη διαδικασία της διαχείρισης ιδεών, όπως οι ακόλουθες, οι οποίες μάλιστα αποτελούν τυπικές διαδικασίες διαχείρισης καινοτομίας:

- Γέννηση μιας ιδέας (Idea generation): Ο σκοπός αυτού του σεναρίου είναι η παραγωγή νέων ιδεών. Το γεγονός ότι η διαδικασία στα Information Aggregation Markets μοιάζει με παιχνίδι, παροτρύνει τους χρήστες να προτείνουν τις ιδέες τους, οι οποίες μπαίνουν στην αγορά. Η αγοραπωλησία γίνεται πάνω σε όλες τις ιδέες, καινούριες που μόλις έχουν προταθεί και παλιές. Οι συμμετέχοντες αποζημιώνονται όχι μόνο σύμφωνα με το χαρτοφυλάκιο των μετοχών (stock portfolio) τους, αλλά και για τη συνεισφορά τους σε ιδέες.

- Εμπλουτισμός της ιδέας (Idea enhancement): Ο σκοπός αυτού του σεναρίου είναι ο εμπλουτισμός και η ανάπτυξη των ιδεών της αγοράς. Οι συμμετέχοντες στην αγορά μπορούν να επενδύουν σε μια ιδέα και ακολούθως να συνεισφέρουν σε αυτή για παράδειγμα θέτοντας ερωτήσεις σχετικές με την ιδέα, προτείνοντας βελτιώσεις της ιδέας ή αλλαγές σύμφωνα με την προσωπική τους άποψη.
- Αξιολόγηση της ιδέας (Idea evaluation): Στο σενάριο αυτό σκοπός είναι η αξιολόγηση νέων ιδεών. Μια αγορά στήνεται με έναν αριθμό από νέες ιδέες και παίκτες (traders) που λειτουργούν ως εκτιμητές, οι οποίοι συναλλάσσονται μετοχές ιδεών (idea stocks) σε μια προσπάθεια αύξησης της αξίας του χαρτοφυλακίου των μετοχών τους. Από τις συναλλαγές των μετοχών μπορούμε να εντοπίσουμε τις πιο πολλά υποσχόμενες ιδέες. Επίσης, μπορεί να ζητείται και ανάδραση (feedback) από τους traders.

Για να υποστηρίξει τα προαναφερθέντα σενάρια, το IDeM περιέχει λειτουργίες για δύο ρόλους: τον συντονιστή της αγοράς (Market Administrator) και τον παίκτη (Trader). Ο Administrator είναι υπεύθυνος για το στήσιμο νέων αγορών, τη διαχείριση αγορών που ήδη υπάρχουν, την επιλογή του αρχικού συνόλου ιδεών που συμπεριλαμβάνονται σε μια αγορά και την πρόσκληση traders για συμμετοχή στην αγορά. Ο αντίστοιχος ρόλος οργανωτικά σε μια εταιρεία είναι κάποιο άτομο υπεύθυνο για τη διαδικασία διαχείρισης καινοτομίας. Οι Traders είναι υπάλληλοι της εταιρείας οι οποίοι συνεισφέρουν με ιδέες ή συμμετέχουν στον εμπλουτισμό και την αξιολόγηση των ιδεών.



Εικόνα 2-2 : Η αρχιτεκτονική του IDeM

Ως προς την αρχιτεκτονική του συστήματος, την οποία μπορούμε να δούμε και ακολούθως στην εικόνα 2.2, διακρίνουμε μια αρχιτεκτονική τριών στρωμάτων η οποία επιτρέπει να χρησιμοποιούνται διαφορετικά εργαλεία για την ανάπτυξη του συστήματος και, επίσης, καθιστά δυνατή κάποια πιθανή επέκταση με νέα χαρακτηριστικά.

Ο **μηχανισμός της αγοράς (Market Engine)** αποτελεί τον πυρήνα του IDeM. Είναι υπεύθυνος για να εκτελεί και να διαχειρίζεται τις συναλλαγές αγοράς και πώλησης. Όταν ένας παίκτης τοποθετεί μια παραγγελία για κάποια συγκεκριμένη ιδέα, αυτή αποθηκεύεται σε έναν πίνακα που ονομάζεται "book of orders". Το σύστημα, ανάλογα με τον αλγόριθμο που χρησιμοποιεί κάθε φορά, αξιολογεί ποιες παραγγελίες θα εκτελεσθούν και με ποια σειρά.

Διαμόρφωση της αγοράς και του γραφικού περιβάλλοντος (Market and GUI configuration): Το μέρος αυτό του συστήματος αφορά στη διαχείριση των ρυθμίσεων του συστήματος. Μία νέα αγορά μπορεί να στηθεί ακολουθώντας μια καθοδηγούμενη διαδικασία τριών βημάτων, κατά την οποία το σύστημα ζητά από τον Market Administrator ορισμένες ρυθμίσεις, δίνοντάς του έτσι τη δυνατότητα να διαμορφώσει ορισμένες παραμέτρους: την ημερομηνία κλεισίματος της αγοράς, τις ώρες λειτουργίας, τις ιδέες που θα μπουν στην αγορά, τους χρήστες που θα προσκληθούν να συμμετέχουν. Άλλες ρυθμίσεις έχουν να κάνουν με πόσα χρήματα και μετοχές ξεκινάνε οι χρήστες, αν θα επιτρέπεται η υποβολή νέων ιδεών από τους χρήστες και ο τρόπος χειρισμού αυτών, η υποβολή feedback για την αξιολόγηση και εμπλουτισμό των ιδεών, κα.

Η **διεπαφή των δεδομένων της αγοράς (Market Data Interface)** περιλαμβάνει web υπηρεσίες, έτσι ώστε οι εταιρικές εφαρμογές να μπορούν να έχουν εύκολη πρόσβαση σε αυτές στο μέλλον για την επεξεργασία τους. Τα δεδομένα που εκτίθενται περιλαμβάνουν την κατάταξη των traders με βάση την αξία του χαρτοφυλακίου τους, καθώς και την κατάταξη των συναλλαγών των μετοχών που γίνονται. Ακόμη, η διεπαφή περιλαμβάνει και στατιστικά στοιχεία που παρέχονται στους traders όπως η υψηλότερη και χαμηλότερη τιμή μιας μετοχής, η διακύμανση της τιμής με το χρόνο, η τιμή της τελευταίας συναλλαγής καθώς και η διακύμανση του όγκου συναλλαγών με το χρόνο. Οι Market Administrators μπορούν να δουν επιπλέον δεδομένα, σχετικά με τον όγκο συναλλαγών/συμμετεχόντων/χρόνου, οι μέσες τιμές των μετοχών και η κατάταξη των ιδεών.

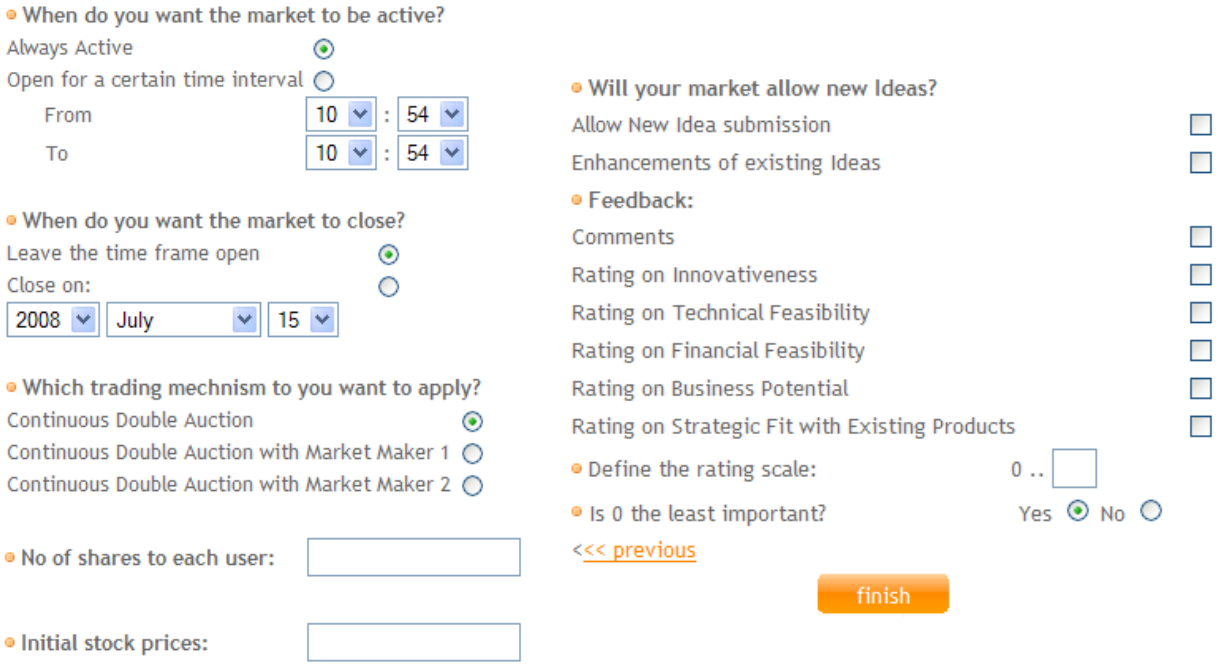
Η μονάδα ασφαλείας (**Security Module**) ελέγχει την πρόσβαση των χρηστών (λειτουργία πιστοποίησης) και τις σελίδες στις οποίες έχουν πρόσβαση οι χρήστες (λειτουργία εξουσιοδότησης). Ανάλογα με το όνομα χρήστη και τον κωδικό πρόσβασης που δίνονται, το σύστημα εισάγει τους χρήστες στο σύστημα ως traders ή ως administrators. Ο market administrator έχει τη δυνατότητα να δημιουργήσει ομάδες χρηστών με πρόσβαση σε ένα

υποσύνολο ενεργών αγορών. Η λειτουργία αυτή επιτρέπει να έχουμε αρκετές αγορές να τρέχουν συγχρόνως με διαφορετικές ομάδες χρηστών σε κάθε αγορά.

Τέλος, ο χειριστής των δεδομένων της αγοράς (**Market Data Handler**) είναι υπεύθυνος για την ανάκτηση και της αποθήκευση δεδομένων στο υποκείμενο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS). Κάθε χρήστης, ανάλογα με το ρόλο του, μπορεί να διαβάσει, να γράψει ή να διαγράψει πληροφορίες από το σύστημα βάσεων δεδομένων.

Το IDeM είναι αναπτυγμένο στην πλατφόρμα "Ruby on Rails" (RoR, www.rubyonrails.org) που χρησιμοποιείται για την ανάπτυξη web εφαρμογών. Με τη χρήση τεχνολογιών web 2.0 η προσοχή εστιάζεται στην επιχειρηματική λογική χωρίς να χρειάζεται η ενασχόληση με περαιτέρω προγραμματισμό λειτουργιών, που δεν είναι απαραίτητες. Το σχεδιαστικό πρότυπο "Model-View-Controller" στο οποίο βασίζεται η RoR, φάνηκε ιδιαίτερα χρήσιμο για την ανάπτυξη της λογικής αρχιτεκτονικής τριών στρωμάτων που βασίστηκε το IDeM. Η αποθήκευση των δεδομένων του IDeM γίνεται σε βάση δεδομένων MySql, ενώ ως web εξυπηρετητής επιλέχθηκε ο Apache Web Server. Τέλος, η μεριά του client πρέπει να διαθέτει έναν web browser με εγκατεστημένα JavaScript και macromedia flash.

Σε μια σύντομη παρουσίαση ροής του συστήματος, μπορούμε αρχικά να δούμε τον Market Administrator ο οποίος μπαίνει στο σύστημα και στήνει μια νέα αγορά και επιλέγει τους traders.

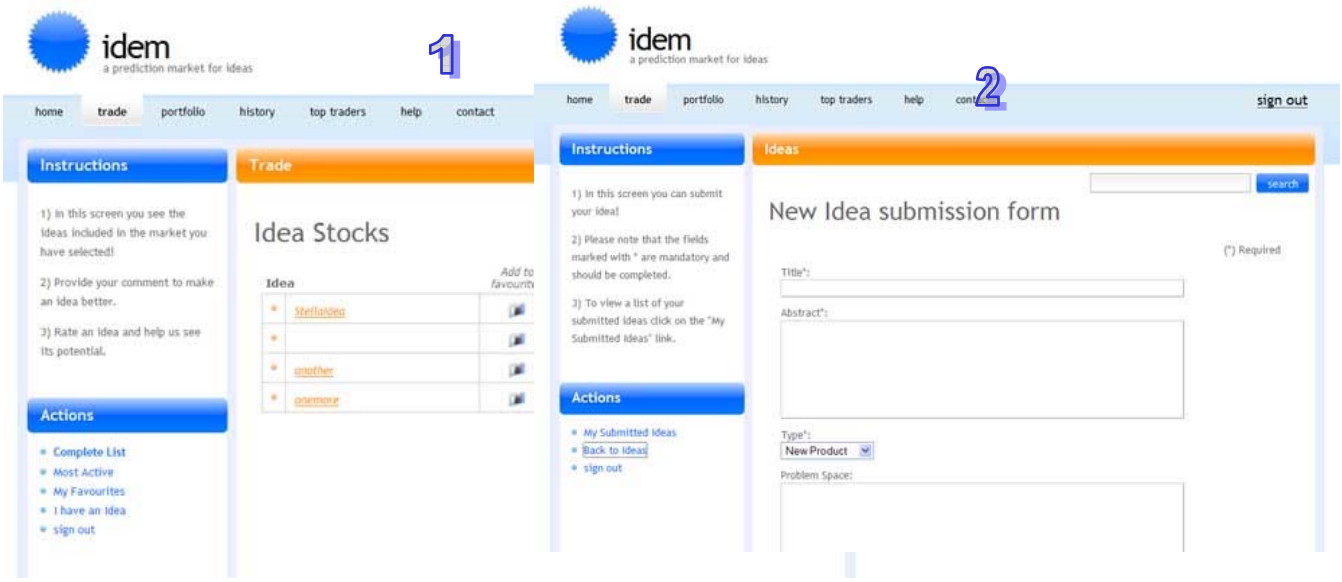


Εικόνα 2-3: Βήματα διαμόρφωσης της νέας αγοράς

Μετά από αυτό το βήμα οι traders μπορούν να μπαίνουν στο σύστημα και να τοποθετούν τις παραγγελίες τους. Η οθόνη των συναλλαγών (trading) , την οποία μπορούμε να δούμε στην εικόνα 2.4, παρέχει διάφορες πληροφορίες: Οι υψηλότερες, χαμηλότερες και τελευταίες τιμές συναλλαγών παρουσιάζονται στο τμήμα 1, ενώ στο τμήμα 2 μπορούμε να δούμε τη χρήση γραφημάτων για τη διακύμανση του όγκου και της τιμής της ιδέας. Οι χρήστες μπορούν να αγοράσουν και να πουλήσουν ιδέες στο τμήμα 3 της οθόνης, ενώ στο τμήμα 4 μπορούν να δουν τη σύντομη και την αναλυτική (κάνοντας κλικ) περιγραφή της ιδέας.

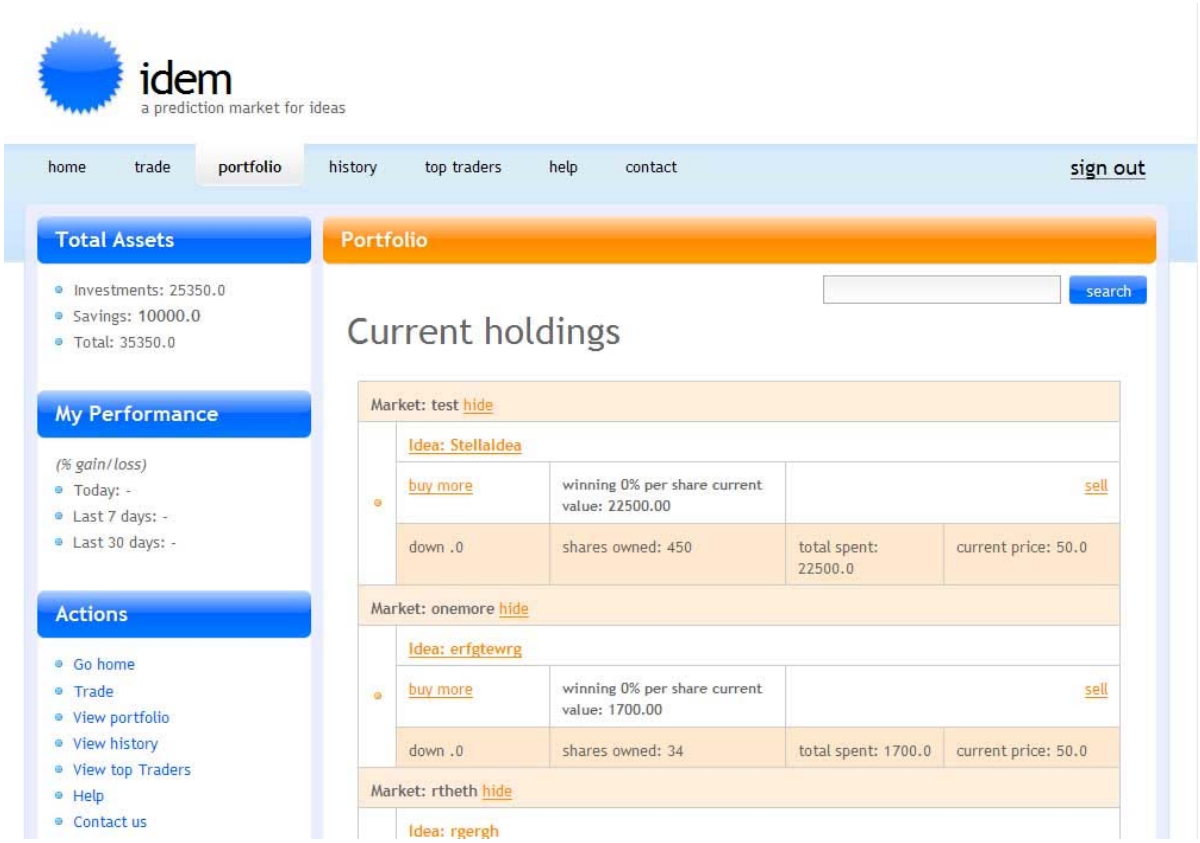
Εικόνα 2-4 : Η οθόνη για το trading

Όταν επιτρέπεται η υποβολή νέων ιδεών από τους χρήστες, αυτοί έχουν τη δυνατότητα πρόσβασης στην οθόνη 1 της εικόνας 2.5. Από εκεί μπορούν να βρουν το σύνδεσμο υποβολής νέας ιδέας (οθόνη 2) όπου πρέπει υποχρεωτικά να συμπληρώσουν τα πεδία του τίτλου, της περίληψης και της κατηγορίας της ιδέας. Άλλα δεδομένα μπορεί να είναι ο χώρος προβλήματος της ιδέας, τεχνικές προδιαγραφές, το πλαίσιο εφαρμογής της και άλλες επιπρόσθετες πληροφορίες.



Εικόνα 2-5 : Η υποβολή νέας ιδέας

Εάν είναι ενεργή η δυνατότητα προσθήκης feedback, ένας υπερσύνδεσμος οδηγεί τους traders στη σελίδα υποβολής feedback. Όπως μπορούμε να δούμε στην εικόνα 2.7, οι χρήστες έχουν πρόσβαση στο χαρτοφυλάκιό τους, όπου παρουσιάζεται μια λίστα από τις ιδέες-μετοχές που έχουν στην κατοχή τους. Σε αυτό το σημείο οι traders μπορούν να δουν τις προσφορές που έχουν κάνει για πώληση/αγορά μετοχών και δεν έχουν προχωρήσει σε συναλλαγές ακόμα, καθώς και να ακυρώσουν εάν θέλουν κάποια προσφορά.

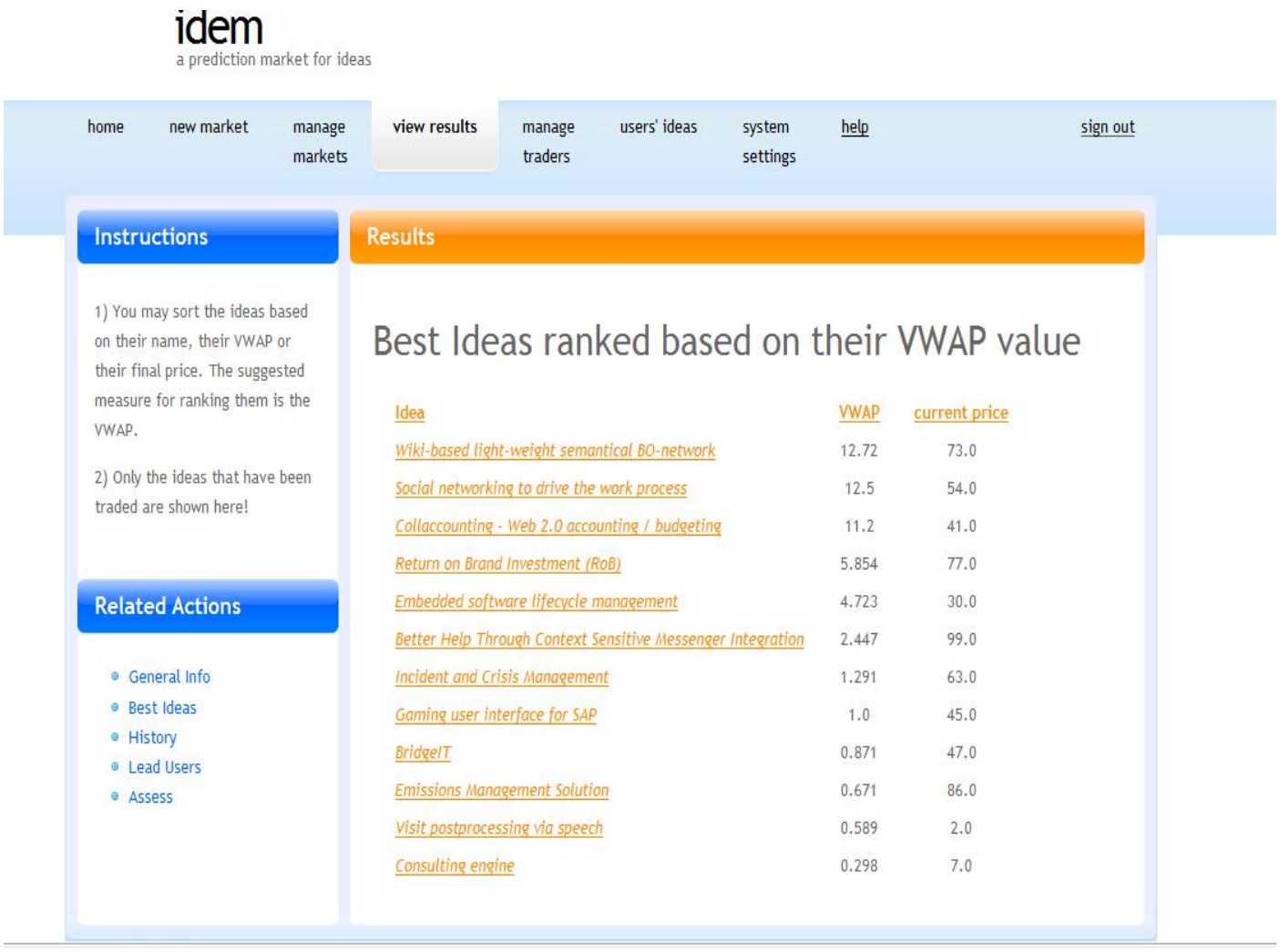


Εικόνα 2-6 : Η οθόνη του χαρτοφυλακίου του trader

Κατά τη λειτουργία της αγοράς οι administrator μπορούν να παρακολουθούν την πορεία της αγοράς και να βλέπουν ή να προσθέτουν νέες ιδέες που προτείνονται. Στο τέλος της αγοράς μπορούν να δουν μια λίστα κατάταξης όλων των ιδεών ως προς το μέτρο Volume Weighted Average Price (VWAP)

$$VWAP_i = \frac{\sum_{t=1}^{T_i} P_{t,i} Q_{t,i}}{\sum_{t=1}^{T_i} Q_{t,i}}$$

όπου τα $P_{t,i}$ και $Q_{t,i}$ συμβολίζουν την τιμή στην οποία έγινε κάθε συναλλαγή και τον αριθμό των μερισμάτων που συναλλάχθηκαν αντίστοιχα. Η κατάταξη αυτή αποτελεί το αποτέλεσμα της αξιολόγησης ιδεών από την αγορά.

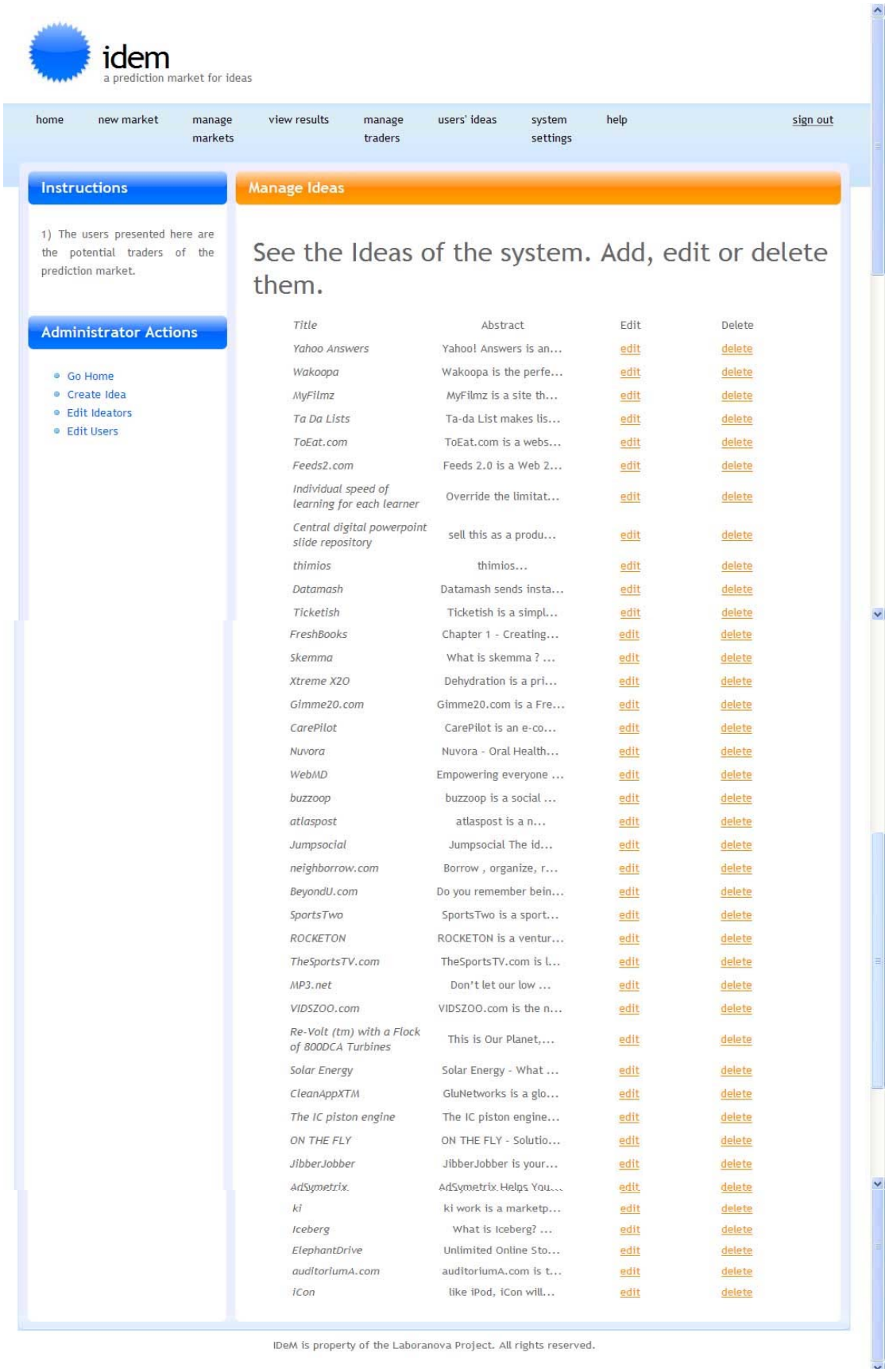


Εικόνα 2-7 : Η κατάταξη των ιδεών

2.3 Η ανάγκη ημιαυτόματης ομαδοποίησης ιδεών

Ανάμεσα στις ερευνητικές προοπτικές που προέκυψαν από την ανάπτυξη του IDeM, υπήρχε και το θέμα του μεγάλου αριθμού ιδεών. Στο περιβάλλον μιας μεγάλης εταιρείας ο αριθμός των ιδεών είναι ιδιαίτερα υψηλός (τάξης μεγέθους εκατοντάδων ιδεών). Εφόσον σε κάθε αγορά γίνεται επιλογή των ιδεών που μπαίνουν, το μεγάλο πλήθος ιδεών δεν επηρεάζει ιδιαίτερα τους traders, εφόσον κάθε φορά έχουν να δουν έναν περιορισμένο αριθμό ιδεών που έχουν μπει στην αγορά. Ο ρόλος του Market Administrator όμως επηρεάζεται δραματικά, αφού για την επιλογή των ιδεών που θα μπουν στην αγορά έχει να μελετήσει ένα πολύ μεγάλο αριθμό από ιδέες για να αποφασίσει. Το πρόβλημα αυτό θα μπορούσε να λυθεί με την ομαδοποίηση των ιδεών ανάλογα με το θεματικό τους περιεχόμενο, ώστε ο administrator να έχει μια πιο εποπτική εικόνα των ιδεών που υπάρχουν και να διευκολύνεται τόσο στη μελέτη των ιδεών όσο και στην επιλογή τους για τη δημιουργία μιας νέας αγοράς.

Λόγω του παραπάνω προβλήματος, αναπτύχθηκε η ιδέα της δημιουργίας κάποιας εφαρμογής ημιαυτόματης ομαδοποίησης ιδεών. Οι ιδέες υποβάλλονται από τους χρήστες σε μορφή κειμένου. Έτσι, επιλέξαμε να ασχοληθούμε με την ανάπτυξη κάποιου συστήματος ομαδοποίησης κειμένου, το οποίο θα είναι συμβατό με το IDeM και θα προσφέρει τη δυνατότητα ομαδοποίησης των κειμένων των ιδεών ως προς το θεματικό τους περιεχόμενο.



Εικόνα 2-8 : Μια απεικόνιση των ιδεών για τον administrator, που δείχνει την ανάγκη για ομαδοποίηση

3

Μέθοδοι και Εργαλεία Εξόρυξης Κειμένου & Επεξεργασίας Φυσικής Γλώσσας για την Ομαδοποίηση Κειμένων

Στο κεφάλαιο αυτό θα προσπαθήσουμε να τοποθετήσουμε τη διαδικασία της ομαδοποίησης των ιδεών-κειμένων σε ένα γενικότερο πλαίσιο. Συγκεκριμένα, θα ασχοληθούμε με τη θεωρία, τις μεθόδους, τις τεχνικές καθώς και εργαλεία του τομέα της Εξόρυξης Κειμένου καθώς και του τομέα της Επεξεργασίας Φυσικής Γλώσσας, με σκοπό να μελετήσουμε πώς μπορούν να βρουν εφαρμογή στην ομαδοποίηση αρχείων κειμένου.

3.1 Εισαγωγή στην Εξόρυξη Κειμένου

3.1.1 Τι είναι το Text Mining

Είναι γεγονός ότι ζούμε στην «Εποχή της Πληροφορίας». Η ταχεία ανάπτυξη του διαδικτύου (internet) και του Παγκόσμιου Ιστού (WWW – World Wide Web), καθώς και η εισαγωγή των πληροφοριακών συστημάτων σε υπηρεσίες και οργανισμούς τόσο για την εσωτερική τους λειτουργία όσο και για την εξυπηρέτηση του κοινού, έχουν ως αποτέλεσμα τη συνεχή παραγωγή, διακίνηση και αποθήκευση τεράστιου όγκου πληροφορίας σε τρομερές ταχύτητες (GB/hour) καθημερινά, μέσω δεδομένων διαφορετικού περιεχομένου, ακόμη και

διαφορετικού τύπου (κείμενα, εικόνες, audio, video). Συνεπώς παρατηρούμε να αυξάνεται με αλματώδη τρόπο το σύνολο των κειμένων τα οποία τις περισσότερες φορές δεν είναι δομημένα, και μπορεί να είναι γραμμένα σε διάφορους τύπους κειμένων (άρθρα, e-mail, δημοσιεύσεις, HTML κείμενα ιστοσελίδων, δεδομένα ηλεκτρονικού εμπορίου, κα) αλλά και σε διαφορετικές γλώσσες. Ωστόσο, η υπερφόρτωση της πληροφορίας κάνει όλο και πιο εμφανές το πρόβλημα της κατανόησης και του χειρισμού της (πχ αναζήτηση κάποιου στοιχείου, ανάλυση κειμένου, ερώτηση, κα) από τους ανθρώπους ενώ τα περισσότερα κείμενα δεν μπορούν να υποβληθούν σε αυτόματη επεξεργασία με κ`άποιο τυποποιημένο τρόπο, εξαιτίας της μη σύνδεσής τους με μεταδεδομένα (metadata: δεδομένα τα οποία χρησιμοποιούνται για την περιγραφή και αναφορά σε άλλα δεδομένα). Συνεπώς γίνεται εύκολα αντιληπτή η ανάγκη για ανάπτυξη αυτοματοποιημένων τεχνικών για την ανακάλυψη, ανάλυση και επεξεργασία πληροφοριών από κειμενικά δεδομένα.

Η ανακάλυψη γνώσης σε κείμενο (Knowledge Discovery in Text - KDT) καθώς και η εξόρυξη κειμένου (Text Mining – TM) περιλαμβάνουν αυτοματοποιημένες τεχνικές για την ανάλυση πολύ μεγάλων συλλογών από δεδομένα αλλά και την εξαγωγή χρήσιμων πληροφοριών από αυτά, οι οποίες βρίσκονται σήμερα στο επίκεντρο του ενδιαφέροντος τόσο από εμπορική όσο και από επιστημονική πλευρά. Χρησιμοποιώντας τεχνικές από την εξόρυξη δεδομένων (text mining), την μηχανική μάθηση (machine learning), τη στατιστική (statistics) την επεξεργασία φυσικής γλώσσας (natural language processing), την ανάκτηση πληροφορίας (information retrieval), την εξαγωγή πληροφορίας (information extraction) και τη διαχείριση γνώσης (knowledge management), οι τεχνικές αυτές προσπαθούν να επιλύσουν το πρόβλημα της μετατροπής των τεραστίων ποσοτήτων από δεδομένα, σε χρήσιμη γνώση. Καθώς δεν υπάρχει καθιερωμένο λεξιλόγιο για αυτό τον αναπτυσσόμενο ερευνητικό τομέα, συχνά απαντώνται διαφορετικοί όροι για να δηλώσουν το ίδιο πράγμα: Ανακάλυψη γνώσης σε κείμενο (Knowledge Discovery in Text), Κειμενική Εξόρυξη Δεδομένων (Text Data Mining), Εξόρυξη Κειμένου ή Εξόρυξη Γνώσης από Κείμενα (Text Mining).

Διαχωρίζοντας τον όρο knowledge discovery in text από τον όρο text mining, μπορούμε να πούμε ότι η εξόρυξη κειμένου αποτελεί ένα στάδιο της ανακάλυψης γνώσης σε κείμενο, η οποία είναι μια διαδικασία που περιλαμβάνει πολλά βήματα για την ανεύρεση χρήσιμης πληροφορίας από κείμενα, από την συλλογή των εγγράφων, την προ-επεξεργασία τους (ώστε να μετατραπούν σε κάποια επιθυμητή αναπαράσταση όπως XML, SGML κλπ), την εξαγωγή λεκτικών πληροφοριών σχετικών με το περιεχόμενο κάθε εγγράφου, την εξόρυξη κειμένου μέσω της δημιουργίας μεταδεδομένων (metadata creation) και της αναγνώρισης προτύπων και συσχετίσεων μεταξύ των δεδομένων, μέχρι και την απεικόνιση (οπτικοποίηση-visualization) της γνώσης που προκύπτει.

Η διαδικασία της εξόρυξης γνώσης από κείμενο (text mining) χρησιμοποιεί πολύ μεγάλα σύνολα κειμένων (γνωστά και ως corpora) που είναι αποθηκευμένα είτε στο διαδίκτυο είτε συμβατικά, και περιλαμβάνει την ανακάλυψη (discovery) προτύπων (patterns) ανάμεσα στα σύνολα δεδομένων (data sets) που περιλαμβάνονται στα κείμενα, που πριν δεν ήταν γνωστά, ισχύουν, είναι κατανοητά και πιθανώς χρήσιμα, καθώς και την ανάλυσή τους για να βρούμε μη αναμενόμενες συσχετίσεις ανάμεσα στα δεδομένα και να τα συνοψίσουμε με νέους τρόπους που είναι κατανοητοί και χρήσιμοι στους χρήστες.

Για να επεξηγήσουμε τον όρο «πρότυπο» που προαναφέρθηκε, μπορούμε να θεωρήσουμε τα δεδομένα μας ως ένα σύνολο γεγονότων F (πχ περιπτώσεις σε μια βάση δεδομένων). Το πρότυπο είναι ένας κανόνας E ο οποίος περιγράφει γεγονότα σε ένα υποσύνολο F_E toy F. Μπορεί να έχουμε είτε πρότυπα πρόβλεψης (predictive pattern), με σκοπό την πρόβλεψη ενός ή περισσότερων γνωρισμάτων (attributes) από αυτά που υπάρχουν στη βάση, είτε πρότυπα ενημέρωσης (informative pattern) τα οποία δεν επιλύουν κάποιο συγκεκριμένο πρόβλημα αλλά παρουσιάζουν στο χρήστη ενδιαφέροντα πρότυπα που θα έπρεπε να γνωρίζει.

Έτσι, το text mining εξετάζει μεγάλες συλλογές από έγγραφα (documents) μη δομημένων κειμένων προκειμένου να ανακαλύψει τη δομή καθώς και αυτονόητα «νοήματα» που κρύβονται μέσα στο κείμενο. Έτσι, όπως η εξόρυξη δεδομένων εντοπίζει συνδέσεις και συσχετίσεις που δεν ήταν προηγουμένως γνωστές ανάμεσα σε δομημένα δεδομένα, έτσι και η εξόρυξη δεδομένων βρίσκει συνδέσεις ανάμεσα σε κείμενα, τα οποία όμως αποτελούν μη δομημένα δεδομένα.

3.1.2 Στόχοι και Τεχνικές Εξόρυξης Κειμένου

Η εξόρυξη κειμένου στοχεύει στην εξαγωγή πληροφοριών από μεγάλο όγκο κειμένων οι οποίες μπορεί να φανούν χρήσιμες προς το χρήστη, μέσω της ανακάλυψης προτύπων ανάμεσα στις πληροφορίες και τα μεταδεδομένα που έχουν προκύψει από αυτά ύστερα από επεξεργασία των μη δομημένων δεδομένων τους.

Οι κυριότερες κατηγορίες των μεθόδων που χειρίζεται η εξόρυξη κειμένου είναι:

- Εξαγωγή χαρακτηριστικών γνωρισμάτων (Feature Extraction)
- Πλοήγηση με βάση το κείμενο (Text Based Navigation)
- Κατηγοριοποίηση, κατάταξη με επίβλεψη (Categorization, Supervised Classification)
- Ομαδοποίηση, μη επιβλεπόμενη κατάταξη (Clustering, Unsupervised Classification)
- Περιληπτική Παρουσίαση της Πληροφορίας (Summarization)
- Γλωσσικός προσδιορισμός (Language Identification) και απόδοση κειμένου στο συγγραφέα

- Συσχετίσεις (Associations)
- Απεικόνιση – Οπτικοποίηση (Visualization)

3.1.2.1 Εξαγωγή χαρακτηριστικών γνωρισμάτων (Feature Extraction)

Έχει ως στόχο τον προσδιορισμό γεγονότων και σχέσεων στο κείμενο, διακρίνοντας συχνά εάν κάποια ονομαστική φράση είναι πρόσωπο, θέση, οργανισμός ή άλλο διακριτό αντικείμενο. Οι αλγόριθμοι εξαγωγής χαρακτηριστικών περιλαμβάνουν την εξαγωγή ονόματος (εντοπίζονται εμφανίσεις ονομάτων στο κείμενο και καθορίζεται σε ποιο τύπο οντότητας αναφέρεται το όνομα), την εξαγωγή όρου μιας περιοχής (προσδιορισμός τεχνικών όρων σε ένα κείμενο) αναγνώριση συντμήσεων (προσδιορίζονται συντμήσεις και αρκτικόλεξα και αντιστοιχούνται στην πλήρη μορφή τους). Φυσικά αυτό περιλαμβάνει έπειτα και την επιλογή σημαντικών όρων και απόρριψη άλλων μη σημαντικών, καθώς και τον υπολογισμό της συχνότητας εμφάνισης των όρων, ενώ οι όροι πρέπει να βρίσκονται σε κανονική ή καθιερωμένη μορφή (πχ χωρίς επιπλέον καταλήξεις λόγω κλίσης της λέξης). Η διαδικασία αυτή μπορεί να χρησιμοποιεί λεξικά για τον προσδιορισμό μερικών όρων καθώς και γλωσσικά υποδείγματα για την ανίχνευση άλλων.

3.1.2.2 Αναζήτηση και Ανάκτηση (Search and Retrieval)

Περιλαμβάνει την αναζήτηση σε εσωτερικές συλλογές εγγράφων ή σε συλλογές που βρίσκονται στον Παγκόσμιο Ιστό. Κύριο χαρακτηριστικό αποτελεί η δυνατότητα, αφού αρχικά συνταχθεί ένα ευρετήριο, να προσφέρεται ένα αρκετά ευρύ φάσμα επιλογών αναζήτησης κειμένου, στις οποίες συμπεριλαμβάνονται οι βασικές επιλογές αναζήτησης (όπως η Boolean, η index-based, η βασισμένη σε οντολογίες, ή στην αριθμητική σειρά, η τμηματική αναζήτηση,) αλλά και πιο σύνθετες επιλογές αναζήτησης (όπως relevancy, έρευνα φυσικής γλώσσας, η αναζήτηση έννοιας, η ασαφής αναζήτηση, κα).

3.1.2.3 Κατηγοριοποίηση, κατάταξη με επίβλεψη (Categorization, Supervised Classification)

Η κατηγοριοποίηση είναι η διαδικασία της κατάταξης εγγράφων σε προκαθορισμένες κατηγορίες. Μας βοηθάει λοιπόν στο να προσδιορίσουμε ποια είναι τα κύρια θέματα μιας συλλογής εγγράφων. Οι κατηγορίες είτε έχουν διαμορφωθεί εξαρχής από τον προγραμματιστή είτε μπορούν να προσδιοριστούν από το χρήστη. Υπάρχουν δύο τρόποι για την κατηγοριοποίηση: ο πρώτος περιλαμβάνει τη δημιουργία ενός θησαυρού (thesaurus), δηλαδή ενός συνόλου που

περιλαμβάνει όρους σχετικούς με το θέμα κάθε κατηγορίας καθώς και συσχετίσεις μεταξύ αυτών των όρων (πχ διευρυμένους όρους, κοντινότερους όρους, συνώνυμα, σχετικούς όρους) και τελικά τον ορισμό του θέματος του κειμένου με βάση τη συχνότητα των όρων σχετικών με το θέμα που υπάρχουν στο έγγραφο. Ο δεύτερος τρόπος περιλαμβάνει την εκπαίδευση (training) του εργαλείου κατηγοριοποίησης με κάποια δείγματα από τα έγγραφα, τη στατιστική ανάλυση λεκτικών προτύπων (linguistic patterns) όπως είναι οι λεξικολογικές συγγένειες, οι συχνότητες λέξεων των εγγράφων προς εκπαίδευση, το χωρισμό αυτών των προτύπων σε κατηγορίες (με στατιστικό τρόπο), και τέλος την ταξινόμηση των υπόλοιπων εγγράφων. Η δεύτερη προσέγγιση είναι προτιμότερη όταν έχουμε να κάνουμε με μεγάλους τομείς, καθώς τότε είναι αρκετά δύσκολο να δημιουργηθεί κάποιος θησαυρός εννοιών.

3.1.2.4 Ομαδοποίηση, μη επιβλεπόμενη κατάταξη (Clustering, Unsupervised Classification)

Μία ομάδα (cluster) είναι μια συλλογή από σχετικά έγγραφα, και η ομαδοποίηση (clustering) είναι η διαδικασία της δημιουργίας ομάδων εγγράφων βάσει κάποιου κριτηρίου ομοιότητας, αυτόματα χωρίς να έχουμε προσδιορίσει από πριν τις κατηγορίες. Η ομαδοποίηση κειμένων είναι χρήσιμη για τον προσδιορισμό κρυμμένων ομοιοτήτων, για να διευκολύνει τη διαδικασία του να βρούμε παρόμοιες ή σχετικές πληροφορίες, ενώ επιπλέον μπορούμε όταν εξερευνούμε μια καινούρια συλλογή δεδομένων να έχουμε μια γενική επισκόπηση της συλλογής. Οι πιο γνωστοί αλγόριθμοι που χρησιμοποιούνται είναι ιεραρχικοί (hierarchical), διαχωριστικοί (partitional), δυαδικοί σχεσιακοί (binary relational) και ασαφείς (fuzzy). Ο πιο σημαντικός παράγοντας στη λειτουργία της ομαδοποίησης είναι το μέτρο ομοιότητας που χρησιμοποιεί ο εκάστοτε αλγόριθμος , καθώς υπάρχουν διάφοροι τύποι μέτρων όπως η θεώρηση λέξεων οι ο οποίες εμφανίζονται συχνά μαζί ως κοινά χαρακτηριστικά, ενώ ένας άλλος τύπος μπορεί να περιλαμβάνει χαρακτηριστικά γνωρίσματα που έχουν εξαχθεί (πχ το όνομα ενός προσώπου).

3.1.2.5 Περιληπτική Παρουσίαση της Πληροφορίας (Summarization)

Αποτελεί την εξαγωγή της περίληψης ενός κειμένου, δηλαδή τη μείωση του μεγέθους του κειμένου διατηρώντας όμως τα βασικά στοιχεία του περιεχομένου του. Σε αυτή τη λειτουργία ο χρήστης έχει συνήθως τη δυνατότητα να καθορίσει διάφορες παραμέτρους, όπως το πλήθος των λέξεων που θα εξαχθούν ή το ποσοστό επί του συνολικού κειμένου που θα αποτελεί την περίληψη.

3.1.2.6 Γλωσσικός προσδιορισμός (Language Identification) και απόδοση κειμένου στο συγγραφέα

Ένα εργαλείο language identification μπορεί να προσδιορίσει σε ποια γλώσσα είναι γραμμένο ένα κείμενο, ή και τι ποσοστό του κειμένου είναι γραμμένο σε κάθε γλώσσα, εάν αυτό είναι γραμμένο σε περισσότερες. Επιπλέον, υπάρχει η δυνατότητα προσδιορισμού του συγγραφέα στον οποίο ανήκει το κείμενο, χρησιμοποιώντας τεχνικές data mining.

3.1.2.7 Συσχετίσεις (Associations)

Στην ανάλυση συσχετίσεων αναγνωρίζονται σχέσεις μεταξύ χαρακτηριστικών γνωρισμάτων που έχουν εξαχθεί από τη συλλογή εγγράφων, και ορίζεται ένα πρότυπο με τη χρήση μιας αντικειμενικής συσχέτισης. Το πρότυπο αυτό, εκφράζει έναν κανόνα που αναφέρει ότι αν βρεθεί η υπολέξη που περιέχεται στο πρότυπο, ακολουθούμενη από μία άλλη δεδομένη υπολέξη, σε συγκεκριμένη απόσταση μεταξύ τους, τότε η αντικειμενική συνθήκη θα διατηρηθεί με μεγάλη πιθανότητα. Οι κανόνες αυτοί είναι πολύ ευέλικτοι για την περιγραφή των τοπικών ομοιοτήτων που περιέχονται στα δεδομένα του κειμένου.

3.1.2.8 Απεικόνιση – Οπτικοποίηση (Visualization)

Το visualization χρησιμοποιεί την εξαγωγή χαρακτηριστικών γνωρισμάτων και το ευρετήριο βασικών όρων για να κατασκευάσει μια γραφική αναπαράσταση μιας συλλογής εγγράφων. Η προσέγγιση αυτή βοηθάει το χρήστη να αναγνωρίζει πολύ γρήγορα τα κύρια θέματα και τις βασικές έννοιες των κειμένων, με βάση τη σπουδαιότητα (πχ μέγεθος) αυτών στην αναπαράσταση.

3.1.3 Μέθοδοι text mining

Ενώ γενικά η εξόρυξη κειμένου περιέχει μεθόδους από διάφορα τεχνολογικά πεδία, θα μπορούσαμε να χωρίσουμε αυτές τις μεθόδους σε δύο κατηγορίες:

Η πρώτη κατηγορία περιλαμβάνει μεθόδους βασισμένες στην απόδοση. Ενδιαφέρει δηλαδή περισσότερο η αποτελεσματική συμπεριφορά του συστήματος και όχι απαραίτητα τα μέσα με τα οποία λαμβάνεται αυτή η συμπεριφορά. Στην κατηγορία αυτοί περιλαμβάνονται διάφορες στατιστικές μέθοδοι (που στηρίζονται συνήθως σε ένα σαφές θεμελιώδες πρότυπο πιθανότητας) καθώς και τα νευρωνικά δίκτυα (neural networks: συστήματα στα οποία οι διασυνδέσεις διαμορφώνονται όπως οι νευρώνες του εγκεφάλου, και οι οποίες μπορούν να αλλάξουν δυναμικά).

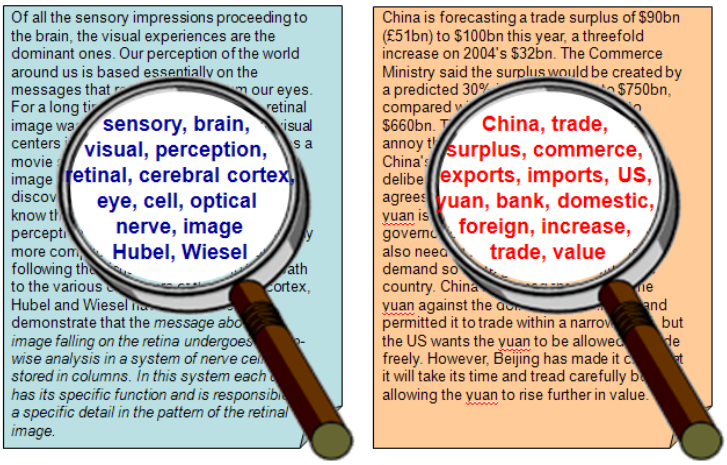
Η δεύτερη κατηγορία περιλαμβάνει μεθόδους βασισμένες στη γνώση. Χρησιμοποιούν δηλαδή σαφείς αντιπροσωπεύσεις της γνώσης όπως οι έννοιες των λέξεων, οι σχέσεις μεταξύ των γεγονότων και των κανόνων για τα συμπεράσματα στις ιδιαίτερες περιοχές. Τέτοια συστήματα περιλαμβάνουν τους κανόνες διεξαγωγής συμπεράσματος, τις λογικές προτάσεις, τα σημασιολογικά δίκτυα (πχ ταξινομήσεις, οντολογίες), κανόνες ταιριάσματος των patterns κα.

Η επιλογή μεταξύ ενός στατιστικά προσανατολισμένου ή ενός βασισμένου στη γνώση εργαλείου εξαρτάται από την περιοχή στην οποία ενδιαφερόμαστε να κάνουμε εξόρυξη δεδομένων. Για παράδειγμα, για περιοχές που δεν αλλάζουν συχνά έννοιες και κανόνες, όπως είναι για παράδειγμα τα οικονομικά και η πολιτική, θα προτιμούσαμε κάποιον αλγόριθμο βασισμένο στη γνώση. Από την άλλη, σε μια περιοχή όπως η γενετική, η οποία αλλάζει συνεχώς έννοιες λόγω της ταχείας εξέλιξης του ερευνητικού αυτού τομέα, είναι προτιμότερο να χρησιμοποιηθεί κάποιο εργαλείο βασισμένο στην απόδοση.

3.1.4 Αναπαράσταση Κειμένου στην Εξόρυξη Κειμένου

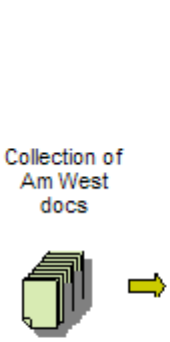
Αφού μελετήσαμε τεχνικές και μεθόδους της εξόρυξης κειμένου, στη συνέχεια θα ασχοληθούμε με τον τρόπο αναπαράστασης κειμένου στη διαδικασία του text mining. Λόγω της συχνής έλλειψης κάποιας δομής στα αρχεία κειμένων, είναι προφανής η ανάγκη εύρεσης μια αναπαράστασης για την αντιπροσώπευση των στοιχείων-όρων των κειμένων, έτσι ώστε να είναι δυνατή η μετέπειτα επεξεργασία τους.

Όταν έχουμε μια συλλογή από αρχεία κειμένου, μπορούμε να θεωρήσουμε καθένα από αυτά ως ένα bag-of-words, μια «σακούλα» η οποία περιλαμβάνει όλες τις λέξεις που βρίσκονται στο κείμενο.



Εικόνα 3-1 : Κάθε αρχείο αποτελεί ένα bag-of-words

Ο πιο συνήθης τρόπος αναπαράστασης ενός κειμένου είναι η αναπαράσταση διανύσματος (vector representation), η οποία προέρχεται από τα συστήματα ανάκτησης πληροφορίας (information retrieval). Έτσι, κάθε text document από το σύνολο κειμένων που έχουμε είναι και ένα διάνυσμα όρων (term vector) στο οποίο κάθε όρος αποτελεί ένα μοναδικό ανεξάρτητο χαρακτηριστικό (feature). Κάθε στοιχείο σε αυτό το διάνυσμα έχει και μια τιμή η οποία αντιστοιχεί στην εμφάνιση του όρου μέσα στο κείμενο.

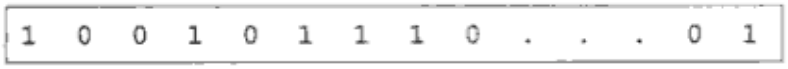


	water	rights	cattle	hunting	land	use	dust	maize	churnash	navajo	reservation	dispossession	war	disease
doc1	1	1			2									
doc2			1	1	1									1
doc3	2	1	1		1	1	1							
doc4			2		1						1			
doc5	1	1	1	1	1	1	1							
doc6		1		1										
doc7									1	1	1	2	1	
doc8					1			1	1					
doc9										1	1	1		
d10									1		1			1
d11	1				1				1	1				
d12									1	1	1	2	2	1

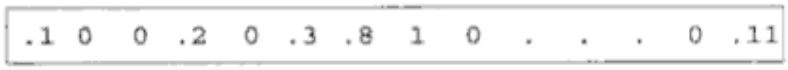
Εικόνα 3-2 : Από το bag-of-words στη δημιουργία word vectors

Με βάση αυτό μπορούμε να διακρίνουμε διάφορα μοντέλα διανυσματικής αναπαράστασης των κειμένων:

Στο λογικό μοντέλο (Boolean model), κάθε έγγραφο αναπαρίσταται από ένα σύνολο λογικών τιμών κάθε μία από τις οποίες δηλώνει εάν ένας συγκεκριμένος όρος εμφανίζεται στο έγγραφο: συνήθως η τιμή 1 σημαίνει ότι εμφανίζεται και η τιμή 0 σημαίνει απουσία του συγκεκριμένου όρου από το κείμενο. Τα πλεονεκτήματα του λογικού μοντέλου είναι η ευκολία και η ταχύτητα λειτουργιών ερώτησης, αναζήτησης, κα, εφόσον χρησιμοποιούνται λογικές πράξης AND, OR, NOT κλπ, και η δυνατότητα χρησιμοποίησης της Boolean άλγεβρας στο Boolean model. Ωστόσο, το λογικό μοντέλο συνεπάγεται ότι η απάντηση στο κατά πόσον είναι σχετικό ένα κείμενο με ένα συγκεκριμένο όρο (και κατ' επέκταση θέμα) είναι μια δυαδική (binary) απόφαση, ενώ επιπλέον μία λογική τιμή για κάθε χαρακτηριστικό δεν μπορεί να αποδώσει κατά πόσο σημαντική είναι η παρουσία μίας λέξης σε ένα κείμενο, γεγονός το οποίο συχνά μπορεί να οδηγήσει σε λάθος συμπεράσματα.



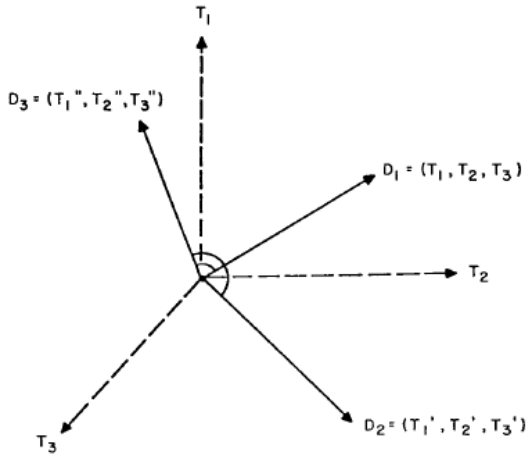
Document Vector - Boolean Model



Document Vector - Term-Weighted Model

Εικόνα 3-3 : Παράδειγμα ενός document vector σε boolean μορφή (πάνω) και σε word frequency μορφή (κάτω)

Στο μοντέλο διανυσματικού χώρου (vector space model – VSM) τα αρχεία αναπαρίστανται ως διανύσματα σε ένα πολυδιάστατο Ευκλείδειο χώρο. Κάθε άξονας στο χώρο αντιστοιχεί σε ένα χαρακτηριστικό (attribute), δηλαδή σε έναν όρο/λέξη, με αποτέλεσμα η συντεταγμένη κάθε διανύσματος ως προς έναν άξονα να χαρακτηρίζει την εμφάνιση του όρου (στον οποίο αντιστοιχεί ο άξονας) στο συγκεκριμένο διάνυσμα-αρχείο κειμένου, και μάλιστα να αποτελεί ένα «βάρος» του όρου (term weight) ως προς το συγκεκριμένο κείμενο (πόσο σημαντικός θεωρείται δηλαδή ο όρος για το κείμενο). Τα βάρη που χρησιμοποιούνται για κάθε attribute είναι πραγματικές τιμές και μπορεί να είναι είτε απλά η συχνότητα εμφάνισης της λέξης (word frequency), είτε άλλες τιμές που θα μελετήσουμε ακολούθως. Τελικά, μια συλλογή εγγράφων αναπαρίσταται από ολόκληρο το διανυσματικό χώρο.



Εικόνα 3-4 : Το vector space model

Ας δούμε τώρα εκτενέστερα τα βάρη (weights) που χρησιμοποιούνται για τις τιμές των συντεταγμένων (που αντιστοιχούν σε όρους) στο Vector Space Model. Θα θεωρήσουμε ότι έχουμε τη συντεταγμένη του αρχείου d που αντιστοιχεί στον άξονα του όρου t.

Καταρχάς, ορίζουμε τις ακόλουθες τιμές για τους όρους (terms) και τα αρχεία (documents):

- Term Frequency - $TF(d,t)$: Είναι η συχνότητα του όρου, πόσες φορές ($n(d,t)$) δηλαδή ο όρος t εμφανίζεται στο αρχείο d.
- Document Frequency - $DF(t)$: Εκφράζει πόσα κείμενα από τη συλλογή που έχουμε περιέχουν τον όρο t.
- D: είναι ο αριθμός των αρχείων που συγκροτούν τη συλλογή κειμένων που έχουμε (άρα και ο αριθμός των διανυσμάτων)
- Inverse Document Frequency - $IDF(t)$: εκφράζει την «σπανιότητα» (scarcity) του όρου μέσα στη συλλογή κειμένων. Έχει διάφορους τρόπους υπολογισμού, από τους οποίους δύο συνήθεις είναι $IDF(t) = \log\left(\frac{D}{DF(t)}\right)$ καθώς και $IDF(t) = \log\left(\frac{1+D}{DF(t)}\right)$ ή $IDF(t) = \log\left(\frac{D - DF(t)}{DF(t)}\right)$

Μπορούμε λοιπόν τώρα να διακρίνουμε τους διάφορους τρόπους απόδοσης βάρους w σε κάθε όρο (term weighting), και άρα υπολογισμού της τιμής της συντεταγμένης.

Ένας πρώτος τρόπος είναι η θεώρηση $w(d,t) = TF(d,t)$, έτσι ώστε κάθε διάνυσμα να είναι της μορφής $\mathrm{d}_{\mathrm{tf}} = (\mathrm{tf}_1, \mathrm{tf}_2, \mathrm{tf}_3, ..., \mathrm{tf}_n)$. Η πιο απλή μορφή για το term frequency είναι τα term counts, ο αριθμός δηλαδή εμφάνισης μιας λέξης σε κάθε κείμενο ($TF(d,t) = n(d,t)$). Ωστόσο συνήθως υπόκειται σε κάποια κανονικοποίηση (length normalization) έτσι ώστε να μειώνεται ο θόρυβος που προκαλείται από το μέγεθος κειμένων τα οποία εκ των πραγμάτων θα εμφανίζουν περισσότερους όρους με μεγαλύτερη συχνότητα. Έτσι, υπάρχουν ποικίλοι τρόποι υπολογισμού του term frequency όπως :

$$TF(d,t) = \frac{n(d,t)}{\max_t n(d,t)} \quad \text{ή} \quad TF(d,t) = 1 + \log n(d,t)$$

Παρόλα' αυτά, δεν είναι όλοι οι όροι εξίσου σημαντικοί μέσα σε μια συλλογή κειμένων. Για παράδειγμα λέξεις που εμφανίζονται συνέχεια όπως άρθρα, αντωνυμίες κλπ θα έχουν πολύ μεγάλη συχνότητα και θα αποτελούν θόρυβο για την εξακρίβωση των σημαντικών όρων που καθορίζουν το περιεχόμενο ενός κειμένου. Για το λόγο αυτό, θεωρούμε ότι η σπανιότητα (scarcity) ενός όρου μέσα στη συλλογή κειμένων αποτελεί ένα μέτρο για τη σημαντικότητα του όρου. Θεωρούμε λοιπόν ότι η σημαντικότητα είναι αντιστρόφως ανάλογη της εμφάνισης του όρου, και εισάγουμε τον όρο του inverse document frequency στον υπολογισμό του βάρους: $w(d,t) = TF(d,t) * IDF(t)$

Συνεπώς, όροι οι οποίοι εμφανίζονται σε πάρα πολλά αρχεία κειμένου λαμβάνουν μικρό βάρος, ενώ πιο σπάνιοι όροι οι οποίοι εμφανίζονται σε λίγα αρχεία λαμβάνουν μεγάλο βάρος, και θα μπορούσαμε να πούμε λοιπόν ότι περισσότερο ενδιαφέρον παρουσιάζουν όροι οι οποίοι ούτε είναι υπερβολικά συχνοί ούτε υπερβολικά σπάνιοι.

Ο υπολογισμός του term weighting με την προσέγγιση $TF - IDF$ είναι από τους πιο συνήθεις στον τομέα της εξόρυξης δεδομένων, και υπολογίζεται με διάφορους τρόπους, σε σχέση με τους εκάστοτε τύπους που χρησιμοποιούνται για τα μέτρα TF και IDF όπως είδαμε και προηγουμένως.

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$

Query, Q: "gold silver truck"
D_1: "Shipment of gold damaged in a fire"
D_2: "Delivery of silver arrived in a silver truck"
D_3: "Shipment of gold arrived in a truck"
$D = 3$; $IDF = \log(D/df_i)$

		Counts, tf_i						Weights, $w_i = tf_i * IDF_i$			
Terms	**Q**	**D_1**	**D_2**	**D_3**	**df_i**	**D/df_i**	**IDF_i**	**Q**	**D_1**	**D_2**	**D_3**
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761

Εικόνα 3-5 : Παράδειγμα υπολογισμού του διανυσματικού μοντέλου με βάση το βάρος TF-IDF

Υπενθυμίζεται ωστόσο και πάλι η ανάγκη για κανονικοποίηση της τιμής του βάρους, καθώς τα κείμενα μεγάλου μήκους τείνουν να έχουν μεγαλύτερες συχνότητες λέξεων καθώς και περισσότερους όρους. Υπάρχουν διάφοροι τρόποι κανονικοποίησης ως προς το μήκος των αρχείων (document length normalization), όπως για παράδειγμα με πολλαπλασιασμό του term frequency με κάποιο άλλο όρο, ή κανονικοποίηση της απόστασης μεταξύ των διανυσμάτων, την οποία και θα μελετήσουμε στην επόμενη ενότητα.

Συνοπτικά, μπορούμε να πούμε ότι για τον υπολογισμό του vector space model στο οποίο αντιστοιχεί μια συλλογή αρχείων, θα πρέπει αρχικά να γίνει μια προ-επεξεργασία των κειμένων: να αναγνωρισθούν δηλαδή οι λέξεις από τις οποίες αποτελείται κάθε κείμενο (bag-of-words) και μάλιστα, για να βελτιστοποιηθεί η διαδικασία εύρεσης των σημαντικών όρων κάθε κειμένου, οι λέξεις θα πρέπει να υπόκεινται σε κάποια επεξεργασία όπως αφαίρεση

πολύ κοινών λέξεων οι οποίες δεν έχουν νοηματική αξία (άρθρα αντωνυμίες, κλπ), η εύρεση λέξεων αντιστοιχούν στο ίδιο θέμα αλλά έχουν διαφορετική μορφή (πχ παράγωγα της ίδιας λέξης, και η εύρεση τελικά όρων που είναι οι πιο αντιπροσωπευτικοί για κάθε κείμενο ξεχωριστά. Τη διαδικασία αυτή της προ-επεξεργασίας θα την εξετάσουμε αναλυτικά σε επόμενη ενότητα. Μετά από αυτό το βήμα (document indexing) προχωράμε στο βήμα της ανάθεσης βάρους σε κάθε όρο για κάθε κείμενο (term weighting) σε όλη τη συλλογή που έχουμε, ώστε κάθε βάρος να υποδηλώνει πόσο σημαντικός θεωρείται ο εκάστοτε όρος για το αντίστοιχο κείμενο.

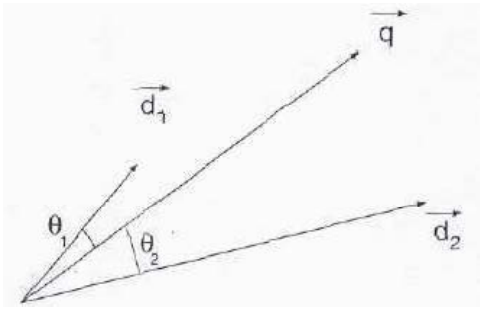
Τέλος, αναφέρονται ως μειονεκτήματα της μεθόδου του Vector Space Model το γεγονός ότι είναι αρκετά αργό ως προς το χρόνο επεξεργασίας του λόγω της πληθώρας υπολογισμών που απαιτούνται, δεν εξυπηρετεί ιδιαίτερα την ενημέρωση αλλαγών στα κείμενα εφόσον για κάθε όρο προστίθεται ένας επιπλέον άξονας και πρέπει να γίνουν υπολογισμοί της συντεταγμένης για όλα τα διανύσματα στο χώρο, ενώ τέλος η πολυδιάστατη μορφή του απαιτεί κόστος μνήμης και χαμηλή ταχύτητα σε υπολογισμούς. Ως πιθανές λύσεις προτείνονται η χρήση συνόλων λέξεων-κλειδιά (keyword-sets) για την αναπαράσταση ενός αρχείου, η οποία θα μειώνει το πλήθος των διαστάσεων, καθώς και η χρήση n-άδων λέξεων (τα λεγόμενα n-grams) δηλαδή ακολουθιών από n λέξεις (πχ το World Wide Web είναι ένα 3-gram) οι οποίες θα μπορούσαν να παρέχουν περισσότερη νοηματική πληροφορία για τα κείμενα από όσο μπορούν οι λέξεις μόνες τους.

3.1.5 Υπολογισμός της ομοιότητας μεταξύ αρχείων κειμένου στο μοντέλο διανυσματικού χώρου

Στην εξόρυξη κειμένου είναι αρκετά συνήθης η προσπάθεια εύρεσης νοηματικής ομοιότητας των αρχείων κειμένου με κάποια θεματική περιοχή (πχ στη διάρκεια της κατηγοριοποίησης) είτε και των αρχείων μεταξύ τους (όπως στην ομαδοποίηση αρχείων κειμένου).

Δεδομένου ότι έχουμε αναπαράσταση των αρχείων κειμένου με διανύσματα, η σύγκριση της ομοιότητας μεταξύ τους ανάγεται στην σύγκριση μεταξύ των διανυσμάτων στα οποία αντιστοιχούν στο μοντέλο διανυσματικού χώρου.

Στο ακόλουθο σχήμα, θεωρούμε δύο διανύσματα d_1 και d_2 στο μοντέλο του διανυσματικού χώρου τα οποία αντιστοιχούν σε αρχεία κειμένου, καθώς και το διάνυσμα q το οποίο αντιστοιχεί σε κάποιο query (ένα σύνολο από όρους).



Εικόνα 3-6 : Σύγκριση εγγράφων στο Vector Space Model

Μπορούμε εύκολα να παρατηρήσουμε λοιπόν ότι θεωρώντας ως μέτρο σύγκρισης την ευκλείδεια απόσταση, το έγγραφο d2 είναι πιο «κοντά» στην ερώτηση (query), ενώ αν θεωρήσουμε ως μέτρο το συνημίτονο της γωνίας δύο διανυσμάτων, το έγγραφο d1 είναι πιο κοντά στο q.

Αν θεωρήσουμε ως μέτρο σύγκρισης της ομοιότητας μεταξύ των αρχείων την απόσταση μεταξύ των διανυσμάτων τους στο χώρο, μπορούμε εύκολα να συμπεράνουμε ότι όσο μεγαλύτερη είναι η απόσταση μεταξύ των διανυσμάτων τόσο πιο ανόμοια είναι τα έγγραφα μεταξύ τους. Από την άλλη, αν θεωρήσουμε ως μέτρο ομοιότητας το πόσο σημαντικά είναι τα έγγραφα μεταξύ τους (δίνοντας τιμές από 0 έως 1), τότε συμπεραίνουμε ότι όσο μεγαλύτερη είναι η τιμή του μέτρου τόσο μεγαλύτερη θα είναι η ομοιότητα των εγγράφων.

Παρατηρούμε συνεπώς ότι υπάρχουν ποικίλα μέτρα για τη σύγκριση της ομοιότητας μεταξύ των term vectors, και ανάλογα με τη φύση των παρατηρήσεων (data sets) του μοντέλου που έχουμε κατασκευάσει θα πρέπει να επιλέξουμε το πιο ιδανικό μέτρο ομοιότητας (αν για παράδειγμα έχουμε binary δεδομένα δεν είναι ιδιαίτερα αποδοτική η χρήση της Ευκλείδειας απόστασης).

Επισημαίνουμε και πάλι ότι συχνά η ομοιότητα των εγγράφων ή όρων είναι αντίστοιχη της απόστασης, αυτό όμως δε συμβαίνει πάντα γι' αυτό και πρέπει να έχουμε υπόψη μας τι υπολογίζει κάθε φορά το μέτρο που χρησιμοποιούμε. Συγκεκριμένα, για να μπορεί να χρησιμοποιηθεί η απόσταση θα πρέπει η μετρική που χρησιμοποιούμε να έχει τις ιδιότητες που έχουν και τα διανύσματα στον Ευκλείδειο χώρο: Δεδομένης μιας συλλογής από αρχεία S, εάν το d: SxS→R είναι ένα μέτρο απόστασης, πρέπει να ικανοποιεί τις ακόλουθες προδιαγραφές:

$$d(x,x) = 0$$

$$d(x,y) \geq 0 \quad \text{όταν} \quad \text{x} \neq \text{y}$$

$$d(x,y) = d(y,x) \quad (\text{συμμετρία})$$

$$d(x,z) \leq d(x,y) + d(y,z) \quad (\text{τριγωνική ανισότητα})$$

Στη γενικότερη περίπτωση θα χρησιμοποιούμε τον όρο της ομοιότητας (SIM), για την οποία δεν ισχύουν πάντα οι ιδιότητες της απόστασης.

Ας δούμε τώρα μερικά από τα πιο γνωστά μέτρα ομοιότητας που χρησιμοποιούνται όταν έχουμε να κάνουμε με ποσοτικά δεδομένα (πχ συχνότητα εμφάνισης λέξεων, κλπ) ενός term-weighted μοντέλου:

- **L1 norm**: $SIM(X_j, X_k) = \sum_{i=1}^{n} \left| x_{ij} - x_{ik} \right|$, το άθροισμα δηλαδή των απόλυτων τιμών των διαφορών ανάμεσα στα δύο διανύσματα των εγγράφων.

- **Euclidean distance** (L2 norm): $SIM(X_j, X_k) = \sqrt{\sum_{i=1}^{n} \left(x_{ij} - x_{ik} \right)^2}$, δηλαδή η ευκλείδεια απόσταση η οποία ορίζεται ως η ρίζα του αθροίσματος των τετραγώνων. Όσο μικρότερη είναι η απόσταση μεταξύ των διανυσμάτων (όσο πιο κοντά βρίσκονται δηλαδή μέσα στο διανυσματικό χώρο) τόσο πιο όμοια θεωρούνται τα αντίστοιχα έγγραφα μεταξύ τους.

- **Cosine distance**: $SIM(X_j, X_k) = \frac{\sum_{i=1}^{n} \left(x_{ij} \times x_{ik} \right)}{\sqrt{\sum_{i=1}^{n} \left(x_{ij} \right)^2} \times \sqrt{\sum_{i=1}^{n} \left(x_{jk} \right)^2}}$, δηλαδή το συνημίτονο της γωνίας μεταξύ των δύο διανυσμάτων. Επειδή όσο πιο κοντά είναι η τιμή ενός συνημίτονου στην τιμή 1 τόσο μεγαλύτερη είναι η γωνία, μπορούμε να θεωρήσουμε ότι όσο πιο κοντά στην τιμή 1 είναι το cosine distance τόσο πιο κοντά είναι μεταξύ τους τα term vectors και τόσο ομοιότερα μεταξύ τους τα αντίστοιχα έγγραφα.

- **Cluster**: $SIM(X_j, X_k) = \frac{\sum_{i=1}^{n} \left(x_{ij} \times x_{ik} \right)}{\sqrt{\sum_{i=1}^{n} x_{ij}}}$ και $SIM(X_k, X_j) = \frac{\sum_{i=1}^{n} \left(x_{ik} \times x_{ij} \right)}{\sqrt{\sum_{i=1}^{n} x_{ik}}}$, που αποτελεί ένα ενδιαφέρον μέτρο ως προς το ότι δεν ισχύει η συμμετρική ιδιότητα, αν δηλαδή πάρουμε δύο λέξεις θα έχουν διαφορετική ομοιότητα μεταξύ τους ανάλογα με τη σειρά που τις πήραμε.

Αξίζει να σημειωθεί ότι η επιλογή του μέτρου ομοιότητας εξαρτάται σημαντικά από τη φύση των δεδομένων. Εάν για παράδειγμα μετράμε τα βάρη των όρων με βάση το TF-IDF χωρίς να λαμβάνουμε υπόψη το μέγεθος του αρχείου, και χρησιμοποιήσουμε την Ευκλείδεια απόσταση ως μέτρο ομοιότητας, τότε το γεγονός ότι αρχεία μεγαλύτερου μήκους τείνουν να

εμφανίζουν περισσότερους όρους και μεγαλύτερες συχνότητες εμφάνισης των όρων θα αποτελεί ένα είδος θορύβου για την εύρεση της ομοιότητας μεταξύ των αρχείων. Σε αυτή την περίπτωση ένα μέτρο που προτείνεται είναι για παράδειγμα το cosine similarity (cosine distance) καθώς με τον υπολογισμό του συνημίτονου επιτελείται κανονικοποίηση ως προς το μήκος των κειμένων (document length normalization).

Ας δούμε τώρα μέτρα που χρησιμοποιούνται όταν έχουμε να κάνουμε με δυαδικά δεδομένα (τιμή 1 για εμφάνιση της λέξης σε ένα κείμενο, τιμή 0 για απουσία της λέξης από το κείμενο):

Προκειμένου να μετρηθεί η ομοιότητα μεταξύ των αντικειμένων συγκρίνουμε πάντα τα ζευγάρια των παρατηρήσεων (έγγραφα ή όροι) *(Xj,Xk)* όπου:

$$X_j^T = (x_{1j},...,x_{nj}) \qquad X_k^T = (x_{1k},...,x_{nk}) \qquad x_{ij},.x_{ik} \in [0,1]$$

Υπάρχουν τέσσερις περιπτώσεις:

$$x_{ij} = .x_{ik} = 1 \qquad x_{ij} = 0,\ \ x_{ik} = 1 \qquad x_{ij} = 1,\ \ x_{ik} = 0 \qquad x_{ij} = .x_{ik} = 0$$

Και έτσι έχουμε:

$$a = \sum_{i=1}^{n} l(x_{ij} = .x_{ik} = 1)$$

$$b = \sum_{i=1}^{n} l(x_{ij} = 0,\ \ x_{ik} = 1)$$

$$c = \sum_{i=1}^{n} l(x_{ij} = 1,\ \ x_{ik} = 0)$$

$$d = \sum_{i=1}^{n} l(x_{ij} = .x_{ik} = 0)$$

Οπότε τρία γνωστά μέτρα ομοιότητας είναι:

- **Dot product coefficient**: $SIM(X_j, X_k) = \frac{a+d}{n}$, που ορίζεται ως ο αριθμητικός μέσος των όμοιων συντεταγμένων των δύο διανυσμάτων, δηλαδή το ποσοστό των όμοιων συντεταγμένων των διανυσμάτων σε σύνολο n.

- **Cluster (binary)**: $SIM(X_j, X_k) = \frac{\sum_{i=1}^{n}(x_{ij} \times x_{ik})}{\sqrt{\sum_{i=1}^{n} x_{ij}}} = \frac{a}{a+c}$ και

$$SIM(X_k, X_j) = \frac{\sum_{i=1}^{n}(x_{ik} \times x_{ij})}{\sqrt{\sum_{i=1}^{n} x_{ik}}} = \frac{a}{a+b}$$

- **Cosine**: $SIM(X_j, X_k) = \frac{\sum_{i=1}^{n}(x_{ij} \times x_{ik})}{\sqrt{\sum_{i=1}^{n}(x_{ij})^2} \times \sqrt{\sum_{i=1}^{n}(x_{jk})^2}} = \frac{a}{\sqrt{(a+c)\times(a+b)}}$

Τέλος, κλείνουμε την ενότητα αυτή δίνοντας τον τρόπο υπολογισμού του κέντρου μιας ομάδας αρχείων (**cluster centroid**) στο μοντέλο του διανυσματικού χώρου:

Δεδομένου ενός συνόλου S από αρχεία κειμένου και τις αντίστοιχες διανυσματικές αναπαραστάσεις τους:

$$S = (d_1^T, d_2^T, d_3^T, \ ..., \ d_m^T)$$

το κεντρικό διάνυσμα c ορίζεται ως

$$c = \frac{\sum_{d=1}^{m} d}{|S|}$$

το οποίο βρίσκεται υπολογίζοντας τον μέσο όρο των βαρών όλων των terms στα αρχεία του συνόλου S.

3.2 Ομαδοποίηση Κειμένων

Θα παρουσιάσουμε τώρα μία από τις τεχνικές του text mining, η οποία αποτέλεσε και αντικείμενο της παρούσας εργασίας, την τεχνική της ομαδοποίησης (clustering).

3.2.1 Η έννοια της ομαδοποίησης

Η κατάταξη των αντικειμένων σε κλάσεις με βάση ομοιότητες που παρουσιάζουν αποτελεί μια πολύ συχνή πρακτική σε πολλούς τομείς, γι' αυτό και αποτελεί ένα αρκετά μεγάλο επιστημονικό πεδίο.

Ομαδοποίηση (clustering) ή Συσταδοποίηση ή Ανάλυση Ομαδοποίησης (Cluster Analysis) όπως αλλιώς ονομάζεται η εύρεση ομάδων αντικειμένων έτσι ώστε τα αντικείμενα σε κάθε ομάδα να είναι όμοια μεταξύ τους (ή να σχετίζονται) και διαφορετικά (ή μη σχετιζόμενα) από τα αντικείμενα των άλλων ομάδων.

Παραδείγματα στα οποία απαντάται η διαδικασία της ομαδοποίησης υπάρχουν σε κάθε τομέα: εύρεση ομάδων πελατών μιας εταιρείας που εμφανίζουν παρόμοια συμπεριφορά, ομαδοποίηση γονιδίων που έχουν την ίδια λειτουργία στον τομέα της γενετικής, ομαδοποίηση μετοχών που παρουσιάζουν παρόμοια διακύμανση τιμών, ομαδοποίηση weblog για εύρεση παρόμοιων προτύπων προσπέλασης, ομαδοποίηση σχετιζόμενων αρχείων για browsing, ομαδοποίηση κειμένων, ομαδοποίηση ασθενειών με βάση τα χαρακτηριστικά τους, κα.

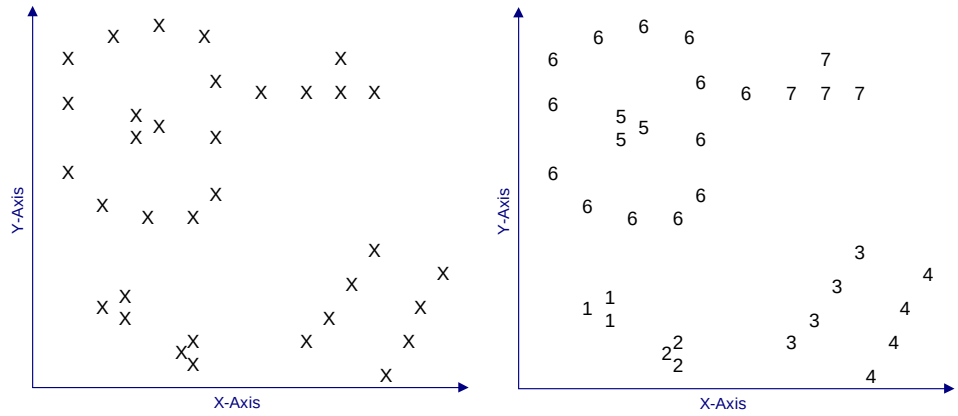
Εάν θεωρήσουμε ότι τα αντικείμενα αναπαρίστανται από στοιχεία στο χώρο, μπορούμε να θεωρήσουμε ως πυκνότητα (density) μιας ομάδας αντικειμένων εκείνη την ιδιότητα που καθορίζει την ομάδα ως ένα σχετικά παχύ σμήνος από σημεία σε ένα χώρο, συγκρινόμενη με άλλες περιοχές του χώρου που μπορεί να έχουν λιγότερα αν όχι και καθόλου σημεία.

Εικόνα 3-7 - Παραδείγματα ομάδων στο χώρο

Έτσι, μπορούμε να δώσουμε τον ορισμό που έδωσε Everrit για την ομάδα: **Ομάδα (cluster)** είναι μια συνεχής περιοχή του χώρου που περιέχει μια σχετικά υψηλή πυκνότητα από σημεία και χωρίζεται από άλλες περιοχές με σχετικά μεγάλη πυκνότητα σημείων, με περιοχές που έχουν χαμηλή πυκνότητα σημείων.

3.2.2 Η διαδικασία του clustering

Μπορούμε λοιπόν να ορίσουμε την ομαδοποίηση (clustering) ως την οργάνωση μιας συλλογής από δείγματα-στοιχεία (patterns) σε ομάδες (clusters) με βάση κάποιο μέτρο ομοιότητας. Τα στοιχεία συνήθως τα περιγράφουμε με τη χρήση διανυσμάτων τιμών ή κάποιων μέτρων, ενώ επίσης μπορούμε και να τα αναπαραστήσουμε ως σημεία σε έναν πολυδιάστατο χώρο.



(α) Τα δεδομένα ως σημεία στο χώρο (β) Η ομαδοποίηση των δεδομένων

Εικόνα 3-8 - Ομαδοποίηση στοιχείων τα οποία αναπαρίστανται ως σημεία στο χώρο

Στοιχεία τα οποία ανήκουν στην ίδια ομάδα παρουσιάζουν μεγαλύτερη ομοιότητα από ότι στοιχεία που ανήκουν σε διαφορετικές ομάδες. Καθώς η ομαδοποίηση σήμερα αποτελεί σημαντικό ερευνητικό πεδίο, έχει αναπτυχθεί μια μεγάλη γκάμα από τεχνικές για την αναπαράσταση των δεδομένων, έκφρασης της ομοιότητας μεταξύ στοιχείων και ομαδοποίησης των δεδομένων, με αποτέλεσμα να υπάρχει πληθώρα μεθόδων ομαδοποίησης.

Σημειώνουμε ότι τα κριτήρια ομοιότητας (similarity) που μπορούν να χρησιμοποιηθούν για να εξακριβωθεί κατά πόσον κάποια αντικείμενα έχουν αρκετά κοινά γνωρίσματα ώστε να θεωρούνται μέλη της ίδιας ομάδας, θα διαφέρουν ανάλογα με τα είδη των γνωρισμάτων των αντικειμένων, που αναφέρθηκαν στην προηγούμενη ενότητα.

Τα βήματα που ακολουθούνται συνήθως στη διαδικασία του clustering είναι τα ακόλουθα:

1) Pattern representation: αναπαράσταση των στοιχείων, που μπορεί να συνδυάζεται με την επιλογή μέρους των χαρακτηριστικών των στοιχείων ή και την παραγωγή νέων χαρακτηριστικών
2) Similarity measure definition: ορισμός του μέτρου ομοιότητας μεταξύ των στοιχείων
3) Clustering: η καθεαυτή διαδικασία της ομαδοποίησης, με εφαρμογή κάποιου αλγορίθμου ομαδοποίησης
4) Data abstraction: αφαίρεση δεδομένων όταν χρειάζεται
5) Assessment of output: προσδιορισμός και εκτίμηση του αποτελέσματος.

3.2.2.1 Ορισμοί και Συμβολισμοί

Στοιχείο (pattern) ή διάνυσμα χαρακτηριστικών x είναι ένα απλό δεδομένο το οποίο υπόκειται σε επεξεργασία από τον αλγόριθμο ομαδοποίησης. Αποτελείται από έναν αριθμό d χαρακτηριστικών και συμβολίζεται με $x=(x_1,x_2,\ldots,x_d)$.

Χαρακτηριστικό ή γνώρισμα (feature, attribute) καλείται κάθε μέρος x_i του στοιχείου x.

Η διάσταση (dimensionality) του κάθε στοιχείου καθώς και του χώρου των δεδομένων είναι ο αριθμός d των χαρακτηριστικών.

Ένα σύνολο από στοιχεία (pattern set) ορίζεται ως $X =\{ x_1,x_2,\ldots,x_d \}$.

Η κλάση (class) αποτελεί μια ομάδα στοιχείων με κοινά ή όμοια χαρακτηριστικά. Οι αλγόριθμοι clustering προσπαθούν να δημιουργήσουν σύνολα στοιχείων τα οποία λογικά αναπαριστούν κλάσεις.

Το μέτρο της απόστασης (distance measure), όπως αναφέρθηκε και στην ενότητα 3.1.5, είναι ένα μέτρο ορισμένο στο χώρο των χαρακτηριστικών στοιχείων και φανερώνει το πόσο όμοια ή διαφορετικά είναι δύο στοιχεία μεταξύ τους.

3.2.2.2 Αναπαράσταση των στοιχείων, εισαγωγή και εξαγωγή χαρακτηριστικών

Η αναπαράσταση των στοιχείων αφορά στον αριθμό των κλάσεων, τον αριθμό των διαθέσιμων στοιχείων, στον αριθμό και τύπο των χαρακτηριστικών τα οποία ενδιαφέρουν τον clustering αλγόριθμο. Σε αυτή τη φάση επιλέγονται τα χαρακτηριστικά των στοιχείων που θεωρούνται ως καταλληλότερα για να χρησιμοποιηθούν στη διαδικασία της ομαδοποίησης, ενώ επίσης δημιουργούνται άλλα που μπορεί να κρίνεται ότι είναι πιο ενδιαφέροντα.

Τα γνωρίσματα μπορούν να διαχωριστούν στις ακόλουθες κατηγορίες:

- Ποιοτικά ή Κατηγορικά
  - Nominal: Οι τιμές είναι απλώς διαφορετικά ονόματα (αναγνωριστικά) με αρκετή πληροφορία ώστε να γίνει διάκριση ανάμεσά τους (=, ≠) (συμπεριλαμβάνονται και οι δυαδικές μεταβλητές 0-1). Παράδειγμα: ταχυδρομικός κώδικας, χρώμα ματιών, φύλο
  - Διάταξης-Cardinal: Οι τιμές περιέχουν πληροφορία διάταξης (<,>). Παράδειγμα: Ποιότητα υλικού (καλή, πιο καλή, άριστη), αριθμοί στις διευθύνσεις
- Ποσοτικά ή Αριθμητικά
  - Διαστήματος-Interval: Έχει σημασία η διαφορά μεταξύ δύο τιμών, υπάρχει μονάδα μέτρησης (+,-). Παράδειγμα: Θερμοκρασία σε Celsius ή Fahrenheit
  - Ratio: Έχει σημασία και ο λόγος μεταξύ δύο τιμών (*, /). Παράδειγμα: Νομισματικές ποσότητες, ηλικία, θερμοκρασία σε Kelvin, ηλικία, μήκος

Αφού εξαχθούν, παραχθούν και επιλεγούν τα καταλληλότερα χαρακτηριστικά, πραγματοποιείται η βέλτιστη αναπαράσταση για τα στοιχεία που θα επεξεργαστεί ο αλγόριθμος της ομαδοποίησης.

3.2.2.3 Ορισμός μέτρου ομοιότητας

Όπως έχει προαναφερθεί, το μέτρο ομοιότητας μεταξύ των στοιχείων καθορίζεται από μια συνάρτηση απόστασης όπως είναι για παράδειγμα η ευκλείδεια απόσταση.

3.2.2.4 Ομαδοποίηση

Η διαδικασία του clustering μπορεί να πραγματοποιηθεί διάφορους τρόπους, και να έχει ένα απόλυτα καθορισμένο αποτέλεσμα (ξένες μεταξύ τους κλάσεις) είτε αφηρημένο (fuzzy), στο οποίο κάποια στοιχεία μπορεί να ανήκουν σε περισσότερες από μία κλάσεις. Υπάρχουν διάφορες τεχνικές clustering, τις οποίες θα μελετήσουμε στην επόμενη ενότητα.

3.2.2.5 Αφαίρεση δεδομένων

Κατά την αφαίρεση δεδομένων, το σύνολο των δεδομένων αποκτά μια απλή αναπαράσταση, τέτοια ώστε οι κλάσεις να είναι καθορισμένες με τρόπο σαφή και κατανοητό για την επεξεργασία των αποτελεσμάτων και την εξαγωγή συμπερασμάτων από τους χρήστες, όσο και για να είναι δυνατή η μετέπειτα αυτοματοποιημένη επεξεργασία των δεδομένων. Συνήθως στην αφαίρεση δεδομένων στο clustering κάθε κλάση αναπαρίσταται συνοπτικά με τη βοήθεια του κεντρικού στοιχείου (centroid) το οποίο χρησιμοποιείται ως αντιπρόσωπο στοιχείο της κλάσης.

3.2.2.6 Προσδιορισμός και εκτίμηση του αποτελέσματος

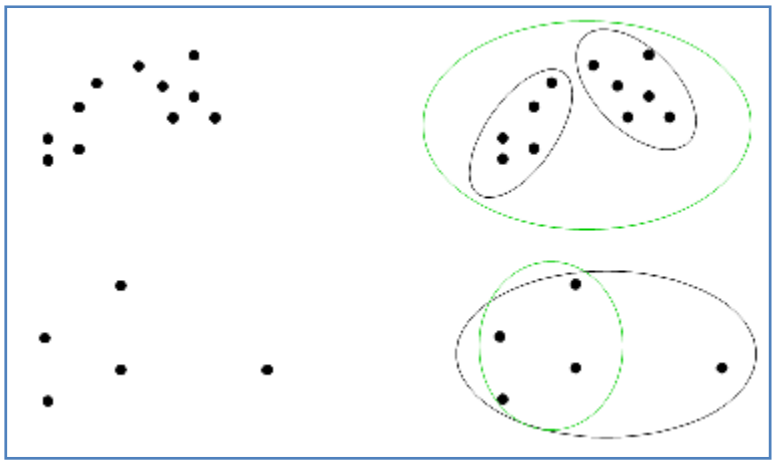
Στο τέλος του clustering γίνεται αξιολόγηση της διαδικασίας που ακολουθήθηκε και εκτίμηση του αποτελέσματος, ώστε να διευκρινιστεί κατά πόσον οι κλάσεις που δημιουργήθηκαν έχουν νόημα ή η ομαδοποίηση έγινε με τυχαίο τρόπο.

3.2.3 Αλγόριθμοι ομαδοποίησης

Οι τεχνικές ομαδοποίησης μπορούν να χωριστούν σε δύο κύριες κατηγορίες:

- στη Διαχωριστική Ομαδοποίηση (Partitional Clustering), στην οποία πραγματοποιείται ένας διαμερισμός των αντικειμένων σε μη επικαλυπτόμενα (non-overlapping) υποσύνολα (clusters) τέτοιος ώστε κάθε αντικείμενο να ανήκει σε ένα ακριβώς υποσύνολο, και

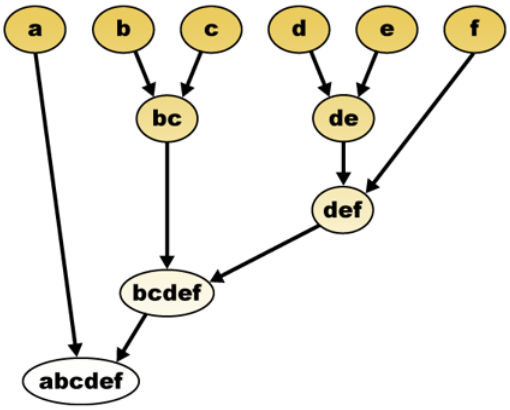
- στην Ιεραρχική Ομαδοποίηση (Hierarchical Clustering) στην οποία δημιουργούμε ένα σύνολο από εμφωλευμένα (nested) clusters, επιτρέποντας έτσι μια ομάδα να έχει υπο-ομάδες οργανωμένες σε ένα ιεραρχικό δέντρο.



(α) Αρχικά σημεία **(β) Ομαδοποίηση**

Εικόνα 3-9 : Διαχωριστική (β – πάνω) και Ιεραρχική ομαδοποίηση (β-κάτω)

Κάθε ιεραρχικός αλγόριθμος δημιουργεί μια ακολουθία από διαμερίσεις τμημάτων με μία μοναδική ομάδα στην κορυφή της δενδρικής ακολουθίας. Κάθε επίπεδο δημιουργείται από τη συγχώνευση δύο ομάδων του κατώτερου επιπέδου (από κάτω προς τα πάνω) ή την διαίρεση μιας μεγαλύτερης ομάδας σε μικρότερες (από πάνω προς τα κάτω). Η ομαδοποίηση των αντικειμένων πραγματοποιείται χρησιμοποιώντας ήδη υπάρχουσες ομάδες, σε πολυπλοκότητα τετραγωνικού χρόνου. Οι ιεραρχικοί αλγόριθμοι μπορούν να εφαρμόζονται χωρίς περιορισμό σε οποιοδήποτε είδος δεδομένων και είναι κατάλληλοι για μεγάλο όγκο δεδομένων. Παράγουν ομάδες με υψηλή ποιότητα και υπάρχει μεγάλη ανομοιογένεια μεταξύ των παραγόμενων ομάδων, ενώ ο χρήστης έχει τη δυνατότητα να αποφασίσει σε ποιο σημείο θα κόψει την παραγωγή του δένδρου. Οι ιεραρχικοί αλγόριθμοι δεν μπορούν να χειριστούν δεδομένα με πολύ θόρυβο επειδή οι αποφάσεις για τη συγχώνευση δύο ομάδων είναι τελικές (δεν υπάρχει επιστροφή σε προηγούμενη κατάσταση). Οι ιεραρχικοί αλγόριθμοι χωρίζονται στους ιεραρχικά συσσωρευτικούς (hierarchical agglomerative) και στους ιεραρχικά διαιρετικούς (hierarchical divisive) αλγορίθμους.



Εικόνα 3-10 : Ένα ιεραρχικό δένδρο ομαδοποίησης

Μερικοί γνωστοί ιεραρχικοί αλγόριθμοι:

- Κοντινότερος γείτονας (nearest neighbor)
- Ο απώτατος γείτονας (farthest neighbor)
- Το ελάχιστο συνδετικό δέντρο (minimum spanning tree)

Οι διαχωριστικοί αλγόριθμοι, σε αντίθεση με τους ιεραρχικούς αλγορίθμους, διαμερίζουν τα δεδομένα μόνο σε ένα σημείο. Έτσι εάν πρέπει να δημιουργηθούν *K* ομάδες με αντικείμενα ο αλγόριθμος κατάτμησης παράγει αυτά τα αντικείμενα αμέσως. Τα αντικείμενα αποδίδονται αυτόματα σε ομάδες με πολυπλοκότητα γραμμικού χρόνου. Οι διαχωριστικοί αλγόριθμοι εφαρμόζονται κυρίως σε δεδομένα που έχουν την έννοια της διαχώρισης, και λόγω της επαναληπτικής τους εκτέλεσης πολλές αποφάσεις που παίρνονται για τα δεδομένα μπορούν να ανακληθούν εάν κριθεί απαραίτητο. Η ποιότητα των παραγόμενων ομάδων εξαρτάται από το αρχικό σύνολο των αντικειμένων και συνήθως οι ομάδες βρίσκονται κοντά ως προς την ομοιότητά τους. Ο αριθμός των ομάδων είναι προκαθορισμένος από την αρχή, ενώ τέλος αντιμετωπίζουν το πρόβλημα του τοπικού ελαχίστου.

Μερικοί διαχωριστικοί αλγόριθμοι:

- Διανυσματικοί μηχανισμοί υποστήριξης (Support Vector Machines – SVM).
- Νευρωνικό Δίκτυο (Neural Network – Nnet).
- K-means αλγόριθμος
- Κατηγοριοποίηση με τη μέθοδο Naïve Bayes (Naïve Bayes classifier – NB).

3.2.4 Text mining και ομαδοποίηση

Επιστρέφοντας στην εξόρυξη κειμένου, υπενθυμίζουμε ότι η ομαδοποίηση κειμένων (clustering) αποτελεί μία από τις τεχνικές text mining, με την οποία δημιουργούμε ομάδες εγγράφων βάσει κάποιου κριτηρίου ομοιότητας, αυτόματα χωρίς να έχουμε προσδιορίσει από πριν τις κατηγορίες.

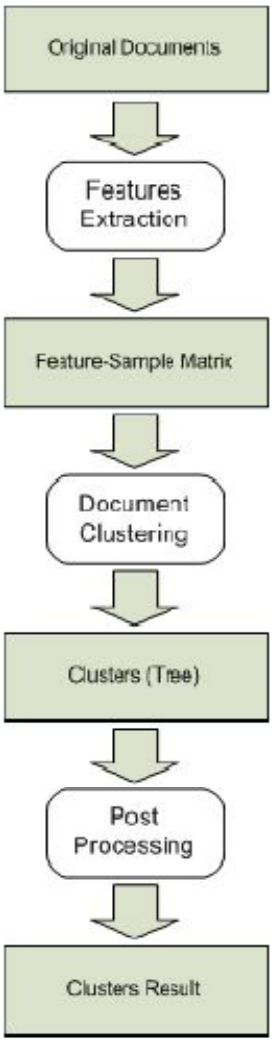
Χρησιμοποιώντας το vector space model, κάθε αρχείο αντιστοιχεί σε ένα διάνυσμα με συντεταγμένες τα βάρη που αντιστοιχούν σε κάθε όρο (άξονα στο χώρο) που εμφανίζεται στη συλλογή αρχείων που έχουμε. Αντιλαμβανόμαστε λοιπόν ότι η διαδικασία ομαδοποίησης των αρχείων κειμένου αντιστοιχίζεται απόλυτα στις μεθόδους ομαδοποίησης που αναφέρθηκαν στην προηγούμενη ενότητα.

Συγκεκριμένα, διακρίνουμε δύο προσεγγίσεις ομαδοποίησης αρχείων κειμένου: την ομαδοποίηση που βασίζεται σε λέξεις κλειδιά (keyword-based clustering) και την ομαδοποίηση που βασίζεται σε αρχεία (document based clustering). Οι δύο προσεγγίσεις διαφέρουν ως προς τα χαρακτηριστικά με βάση τα οποία ομαδοποιούνται τα αρχεία.

Οι αλγόριθμοι document-based clustering εφαρμόζονται κυρίως στο document vector space model στο οποίο κάθε στοιχείο παρουσιάζει το βάρος του όρου (term weighting) στο αντίστοιχο αρχείο. Έτσι, ένα αρχείο τοποθετείται σε ένα σημείο δεδομένων (data point) σε έναν ιδιαίτερα πολύ-διάστατο χώρο στον οποίο κάθε όρος είναι και ένας άξονας. Σε αυτό το χώρο η απόσταση μεταξύ των σημείων μπορεί να υπολογιστεί και να συγκριθεί. Σημεία δεδομένων τα οποία βρίσκονται κοντά μεταξύ τους μπορούν να συγχωνευθούν και να ομαδοποιηθούν στην ίδια ομάδα, ενώ στοιχεία σε μεγάλη απόσταση μεταξύ τους απομονώνονται σε διαφορετικές ομάδες. Συνεπώς τα αντίστοιχα αρχεία ομαδοποιούνται και χωρίζονται. Καθώς το document-based clustering βασίζεται στην «απόσταση μεταξύ των αρχείων» (document distance), είναι ιδιαίτερα σημαντικό να τοποθετούνται τα αρχεία στον κατάλληλο χώρο και να εφαρμόζονται σε αυτά οι κατάλληλες μέθοδοι υπολογισμού απόστασης.

Ου αλγόριθμοι keyword-based clustering επιλέγουν μόνο συγκεκριμένα γνωρίσματα (features) και βασισμένοι σε αυτό το σχετικά περιορισμένο πλήθος γνωρισμάτων δημιουργούν τα clusters. Αυτά τα συγκεκριμένα γνωρίσματα επιλέγονται επειδή θεωρούνται ως τα ουσιώδη γνωρίσματα μεταξύ των αρχείων, τα οποία απαντώνται σε παρόμοια αρχεία και είναι σπάνια σε ανόμοια αρχεία. Συνεπώς, για έναν keyword-based clustering αλγόριθμο είναι ιδιαίτερα σημαντικό το βήμα της επιλογής των πιο σημαντικών γνωρισμάτων.

Ας δούμε λοιπόν τα βήματα τα οποία περιλαμβάνει η διαδικασία της ομαδοποίησης αρχείων κειμένου:



Εικόνα 3-11 : Βήματα της διαδικασίας Text Clustering

Feature Extraction: Το πρώτο βήμα είναι η εξαγωγή των γνωρισμάτων (feature extraction). Παίρνεται ως είσοδος το αρχικό σύνολο με τα κείμενα, και υποβάλλονται όλα αυτά τα ακατέργαστα κείμενα σε επεξεργασία (γι' αυτό και τη διαδικασία αυτή την ονομάζουμε και προ-επεξεργασία των αρχείων – preprocessing) προκειμένου να αναλυθούν και να επιλεγούν τα σχετικά χαρακτηριστικά που μπορεί να περιγράφουν αυτά τα αρχεία. Η έξοδος της διαδικασίας feature extraction είναι συνήθως ένας πίνακας όρων-εγγράφων, στον οποίο κάθε στήλη αντιστοιχεί σε ένα έγγραφο και κάθε γραμμή δηλώνει ένα χαρακτηριστικό.

Η ποιότητα της εξαγωγής γνωρισμάτων έχει μεγάλη επίπτωση στην αποτελεσματικότητα των μετέπειτα clustering αλγορίθμων. Τα παρόμοια έγγραφα θα πρέπει να βρίσκονται κοντά στον χώρο γνωρισμάτων και τα ανόμοια έγγραφα θα πρέπει να βρίσκονται μακριά μεταξύ τους. Εάν έχουν απορριφθεί χρήσιμα και σημαντικά γνωρίσματα και έχουν συμπεριληφθεί άσχετα γνωρίσματα, υπάρχει αταξία στην απόσταση μεταξύ αρχείων, και όσο καλοί κι αν είναι οι

αλγόριθμοι ομαδοποίησης, δε μπορούν να ομαδοποιήσουν τα αρχεία με λανθασμένη απόσταση. Τα βασικά στοιχεία που συμπεριλαμβάνονται στην εξαγωγή γνωρισμάτων είναι η αφαίρεση των stop-words, το stemming, η απόδοση βάρους στους όρους για κάθε αρχείο (term weighting), η εξαγωγή των σημαντικών γνωρισμάτων (key feature extraction) και η δημιουργία του πίνακα. Θα δούμε σε επόμενη ενότητα αναλυτικά τις διαδικασίες αυτές που λαμβάνουν μέρος στην προ-επεξεργασία των αρχείων κειμένου πριν την ομαδοποίησή τους.

Document Clustering: Το δεύτερο βήμα είναι η ομαδοποίηση των αρχείων κειμένου, στην οποία εφαρμόζονται αλγόριθμοι ομαδοποίησης προκειμένου να πάρουμε ένα clusters map, μια αναπαράσταση δηλαδή των ομάδων που έχουν δημιουργηθεί. Το clusters map όχι μόνο μπορεί να δείξει σε ποια ομάδα ανήκει ένα αρχείο, αλλά μπορεί επίσης να περιγράφει τη σχέση μεταξύ των clusters. Η είσοδος σε αυτό το βήμα είναι ο πίνακας όρων-εγγραφών. Με αυτό τον πίνακα, ένα αρχείο αναπαρίσταται ως ένα data point στον πολυδιάστατο χώρο.

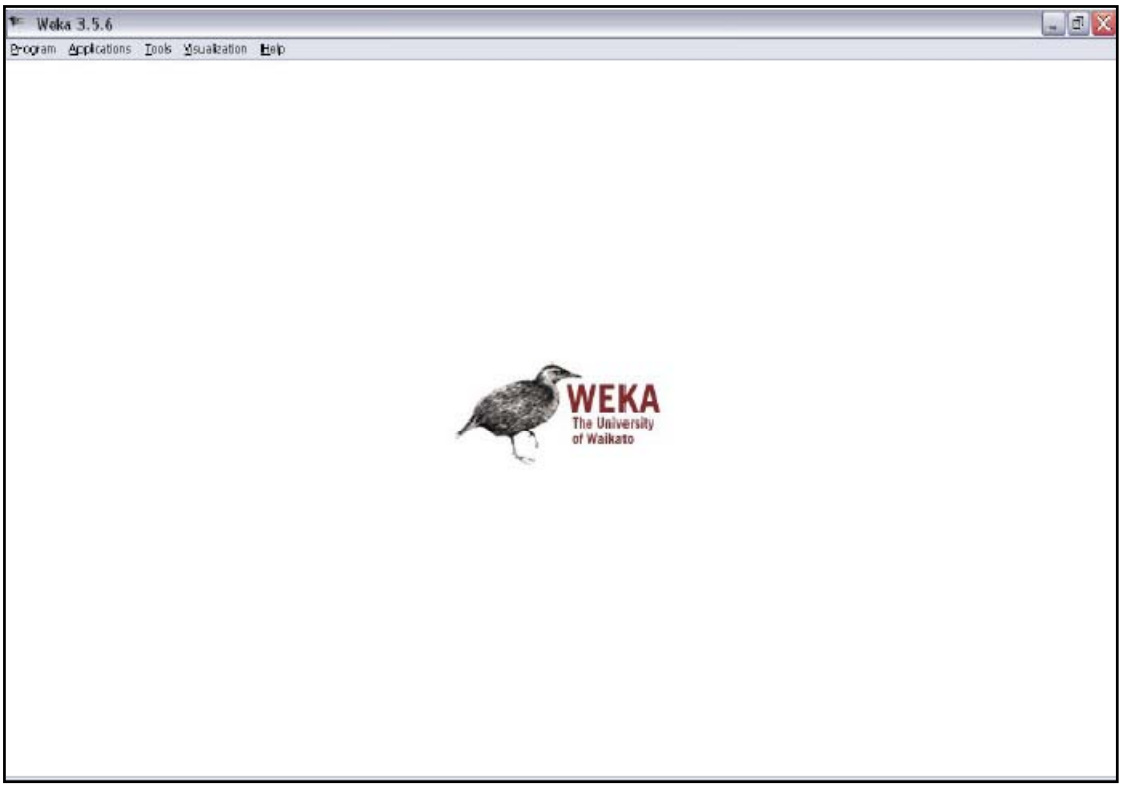
Συνήθως το βήμα της ομαδοποίησης βασίζεται σε στατιστικούς ή μαθηματικούς υπολογισμούς χωρίς επαγγελματική γνώση. Κάθε όρος(γνώρισμα) χάνει το νόημά του και αντιπροσωπεύει απλά μια διάσταση στο χώρο.

Post Processing: Για διαφορετικές εφαρμογές υπάρχουν διαφορετικοί τρόποι να γίνει η μετέπειτα επεξεργασία. Μια κοινή μέθοδος περιλαμβάνει την επιλογή ενός κατάλληλου ορίου (threshold) για την παραγωγή του τελικού αποτελέσματος ομαδοποίησης. Μετά την ομαδοποίηση αρχείων παίρνουμε ένα βασικό cluster map στο οποίο οι ομάδες είναι οργανωμένες είτε σαν ένα δέντρο είτε με επίπεδο τρόπο. Έτσι, οι αλγόριθμοι της μετέπειτα επεξεργασίας (post processing algorithms) εφαρμόζονται προκειμένου να βρεθεί η σωστή σχέση μεταξύ των clusters.

3.3 Ένα εργαλείο εξόρυξης γνώσης από δεδομένα: WEKA.

Το WEKA (Waikato Environment for Knowledge Analysis) είναι ένα πρόγραμμα ανάπτυξης εφαρμογών μηχανικής μάθησης και εξόρυξης γνώσης από δεδομένα (data mining), το οποίο αναπτύχθηκε στο τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου του Waikato της Νέας Ζηλανδίας. Πρόκειται για πακέτο λογισμικού ανοιχτού κώδικα υλοποιημένο σε Java, και χρησιμοποιείται ευρέως τόσο για ερευνητικούς και εκπαιδευτικούς λόγους όσο και για εφαρμογές που σχετίζονται με τον τομέα της εξόρυξης δεδομένων.

Παρέχει μια ολοκληρωμένη συλλογή από υλοποιήσεις αλγορίθμων μηχανικής μάθησης και εξόρυξης δεδομένων, η οποία περιλαμβάνει όλες τις γνωστές μεθόδους φιλτραρίσματος, επιλογής χαρακτηριστικών, ταξινόμησης, εύρεσης κανόνων συσχέτισης (association), κατηγοριοποίησης (categorization) και ομαδοποίησης (clustering), καθώς και μηχανισμούς για προ-επεξεργασία δεδομένων (preprocessing) και μετέπειτα επεξεργασία αποτελεσμάτων (post-processing).



Εικόνα 3-12 : Το γραφικό περιβάλλον του Weka 3.5

Το λογισμικό του WEKA παρουσιάζεται σε διάφορες εκδόσεις, καθώς αναπτύσσεται συνεχώς, με πιο πρόσφατη αυτή τη στιγμή την έκδοση 3.5. Οι ρουτίνες είναι υλοποιημένες σαν classes και ταξινομημένες σε packages, ενώ περιέχεται και ένα αναλυτικό γραφικό περιβάλλον (GUI Interface) . Οι χρήστες έχουν τη δυνατότητα να χρησιμοποιήσουν τις υλοποιήσεις τον αλγορίθμων τόσο από τη γραμμή εντολών όσο και από το γραφικό περιβάλλον του WEKA, ενώ οι προγραμματιστές έχουν τη δυνατότητα να χρησιμοποιούν τις υλοποιήσεις αυτές καλώντας τις αντίστοιχες κλάσεις του WEKA από τα δικά τους προγράμματα. Έτσι, το WEKA μπορεί κάλλιστα να αποτελέσει μια βιβλιοθήκη υλοποιήσεων αλγορίθμων εξόρυξης δεδομένων, οι κλάσεις της οποίας μπορούν να χρησιμοποιούνται για τη δημιουργία νέων προγραμμάτων.

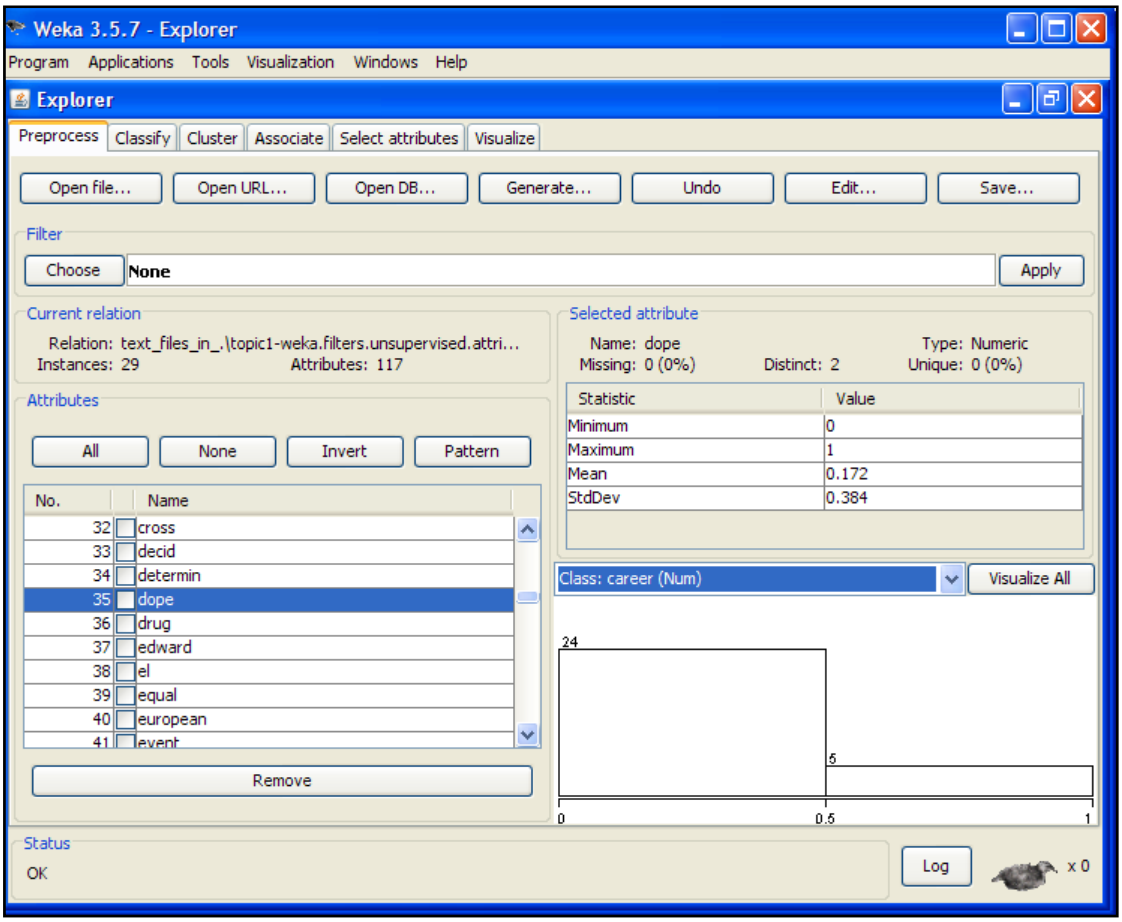
Όσον αφορά στη μορφή των δεδομένων, το WEKA χρησιμοποιεί flat text files (αρχεία τα οποία περιέχουν εγγραφές, στα οποία κάθε εγγραφή περιγράφεται σε μία μόνο γραμμή, τα δε πεδία των εγγραφών έχουν σταθερό πλάτος και χωρίζονται με κενά) για την περιγραφή των

δεδομένων. Η είσοδος στο WEKA δίνεται ως σύνολο δεδομένων (data set), μέσω διαφόρων μορφών αρχείων όπως CSV (comma separated values: *.csv), binary serialized instances (*.bsi) κα, με περισσότερο προτιμητέα και εξυπηρετική τη μορφή ARFF (*.arff) η οποία παράγεται μάλιστα από το ίδιο το WEKA. Τα δεδομένα μπορούν επίσης να διαβαστούν και από μία ηλεκτρονική διεύθυνση ή από μια βάση δεδομένων (με χρήση JDBC) . Ένα παράδειγμα ενός data set αρχείου .arff θα έμοιαζε κάπως έτσι:

```
@relation heart-disease-simplified
@attribute age numeric
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}
@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...
```

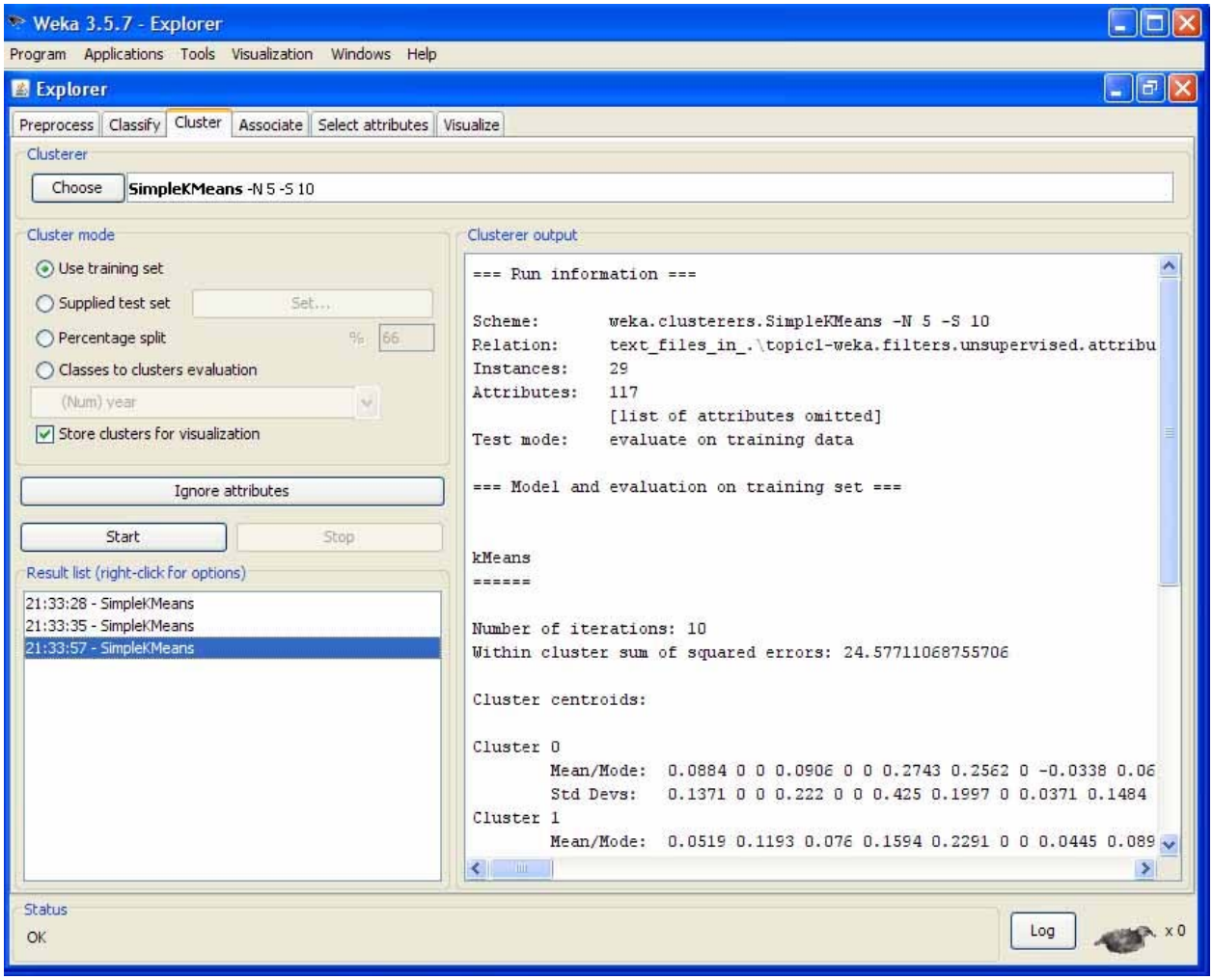
Ας δούμε τώρα πώς μπορεί να χρησιμοποιηθεί το WEKA στην ομαδοποίηση δεδομένων.

Ανοίγοντας το πρόγραμμα, μέσω του μενού Application→Explorer→Open file δίνεται η δυνατότητα να επιλεγεί ένα σύνολο δεδομένων (data set, αρχείο .arff), στο οποίο μπορούμε να εφαρμόσουμε τεχνικές που αφορούν σε προ-επεξεργασία (preprocess), κατηγοριοποίηση (classify), ομαδοποίηση (cluster), συσχέτιση (associate), επιλογή χαρακτηριστικών (select attributes) και οπτικοποίηση (visualize). Μάλιστα, αφού επιλέξουμε το σύνολο δεδομένων, μπορούμε να παρατηρήσουμε ότι στο κάτω δεξιά μέρος του παραθύρου εμφανίζονται γραφικά τα δεδομένα για καθένα από τα γνωρίσματα (attributes) ξεχωριστά καθώς και στατιστικές πληροφορίες για αυτά.



Εικόνα 3-13 : Η καρτέλα Explorer του Weka

Αφού έχει επιλεγεί ένα σύνολο δεδομένων, είναι δυνατόν να γίνει ομαδοποίηση (clustering), μεταβαίνοντας στην καρτέλα cluster και επιλέγοντας έναν αλγόριθμο μα βάση τον οποίο γίνεται η ομαδοποίηση. Πατώντας το κουμπί start μπορεί να ξεκινήσει η εκτέλεση του αλγορίθμου.

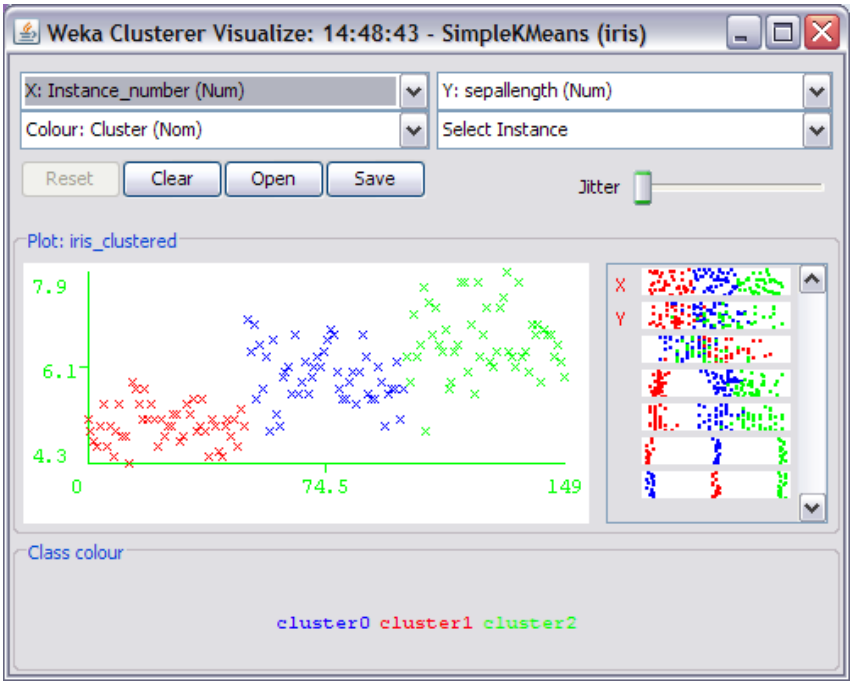


Εικόνα 3-14 : Η καρτέλα cluster του Weka

Οι αλγόριθμοι ομαδοποίησης που μπορεί έχουν υλοποιηθεί στον κώδικα του WEKA είναι οι:

- Cobweb
- DBScan
- EM
- Farthest First
- OPTICS
- SimpleKmeans(K-means)
- Xmeans

Τα αποτελέσματα της ομαδοποίησης εμφανίζονται στο δεξί μέρος της καρτέλας, ενώ επίσης παρέχεται η δυνατότητα οπτικοποίησης (visualize) του αποτελέσματος, με εμφάνιση της γραφικής παράστασης των δεδομένων με βάση τις ομάδες που προέκυψαν από το clustering.



Εικόνα 3-15 : Οπτικοποίηση του αποτελέσματος της ομαδοποίησης

Όπως μπορούμε πολύ εύκολα να διαπιστώσουμε, το Weka μπορεί να χρησιμοποιηθεί για την ομαδοποίηση αρχείων κειμένου χρησιμοποιώντας την εφαρμογή Cluster που προαναφέραμε. Η διαδικασία που ακολουθείται αντιστοιχεί στην διαδικασία ομαδοποίησης που περιγράψαμε στην προηγούμενη ενότητα, πριν όμως από την έναρξη της σημειώνουμε ότι πρέπει το σύνολο των αρχείων κειμένου να μετατραπεί σε είσοδο που να μπορεί να διαβαστεί από το weka. Έτσι, τοποθετώντας τη συλλογή των εγγράφων μέσα σε κάποιο κατάλογο, μπορούμε με μια απλή εφαρμογή να τοποθετήσουμε όλα τα κείμενα μέσα σε ένα αρχείο arff, δημιουργώντας έτσι ένα σύνολο από δεδομένα (data set), στο οποίο κάθε instance αντιστοιχεί σε ένα αρχείο και έχει δύο attributes (σε μορφή string): το όνομα του αρχείου, και το κείμενο που περιέχει το κείμενο και το οποίο αποτελεί ένα ολόκληρο και μοναδικό string. Στη συνέχεια, το αρχείο arff μπορεί να υποστεί την προ-επεξεργασία (κατά τη διάρκεια της οποίας γίνεται και η εξαγωγή γνωρισμάτων) και να παραχθεί ένα word vector (παράγουμε δηλαδή την αναπαράσταση κειμένου ως διάνυσμα από weighted όρους, όπως περιγράψαμε νωρίτερα ότι γίνεται στην εξόρυξη κειμένου). Στη συνέχεια μπορούν να εφαρμοστούν αλγόριθμοι ομαδοποίησης, να γίνει μετέπειτα επεξεργασία και τελικά να πάρουμε τα αποτελέσματα της ομαδοποίησης.

3.4 Μέθοδοι της Επεξεργασίας Φυσικής Γλώσσας στην ομαδοποίηση κειμένου

Όπως αναφέρθηκε και στην πρώτη ενότητα του κεφαλαίου, η εξόρυξη κειμένου χρησιμοποιεί μεθόδους από τον τομέα της Επεξεργασίας Φυσικής Γλώσσας στις τεχνικές που χρησιμοποιεί. Στην ενότητα αυτή θα μελετήσουμε μεθόδους της Επεξεργασίας Φυσικής Γλώσσας που εφαρμόζονται στην ομαδοποίηση αρχείων κειμένου, και πιο συγκεκριμένα στο στάδιο της προ-επεξεργασίας.

3.4.1 Εισαγωγή στην Επεξεργασία Φυσικής Γλώσσας

Ύστερα από την σχεδίαση τεχνητών γλωσσών επικοινωνίας ανθρώπου μηχανής, σήμερα διανύουμε μια περίοδο στην οποία υπάρχει ιδιαίτερο ενδιαφέρον και γίνεται σημαντική επιστημονική έρευνα για την χρήση της ανθρώπινης ή φυσικής γλώσσας ως «μέσο επικοινωνίας» του ανθρώπου με τη μηχανή. Έτσι, έχει γνωρίσει ιδιαίτερη ανάπτυξη ο τομέας της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP), ο οποίος αποτελεί πεδίο έρευνας από διαφορετικές επιστημονικές περιοχές: την Επιστήμη της Πληροφορίας (Information Science), την Τεχνητή Νοημοσύνη (Artificial Intelligence) καθώς και μία νέα ερευνητική κατεύθυνση, διεπιστημονικού χαρακτήρα, την Υπολογιστική Γλωσσολογία (Computational Linguistics) η οποία αξιοποιεί την γλωσσολογική θεωρία και γνώση στην Επεξεργασία Φυσικής Γλώσσας.

Η επεξεργασία της φυσικής γλώσσας καλύπτει τόσο απλές διεργασίες (πχ στατιστική γραμμάτων ενός κειμένου) οι οποίες καλύπτονται από ένα απλό πρόγραμμα, όσο και πιο σύνθετες απεικονίσεις γλωσσολογικής γνώσης (πχ η γραπτή απεικόνιση σε ένα σύστημα λογισμικού μιας ομιλίας που εισάγεται φωνητικά) Γενικά, η επεξεργασία της φυσικής γλώσσας διακρίνεται σε επεξεργασία κειμένου και επεξεργασία φωνής, ανάλογα με το αντικείμενο της επεξεργασίας.

Η διαδικασία της επεξεργασίας φυσικής γλώσσας μπορεί να χωριστεί στα ακόλουθα διακριτά μεταξύ τους επίπεδα:

- Φωνολογικό επίπεδο: είναι το επίπεδο στο οποίο η γλώσσα, η οποία πραγματώνεται με το λόγο, αποτελείται από φθόγγους και φωνήματα.
- Μορφολογική ανάλυση (Morphological Analysis): είναι το επίπεδο παραγωγής λέξεων, ανάλυσής τους στα συστατικά τους, διαχωρισμού τους από μη λεκτικά σύμβολα όπως είναι τα σημεία στίξης.
- Συντακτική ανάλυση (Syntactic Analysis): είναι η μετατροπή γραμμικών ακολουθιών λέξεων σε φράσεις ή προτάσεις, δηλαδή δομές που απεικονίζουν τον τρόπο με τον

οποίο συνδυάζονται οι λέξεις, ακολουθώντας κανόνες φραστικής δομής οι οποίοι δίνονται από γραμματικές.

- Σημασιολογική ανάλυση (Semantic Analysis): είναι η διαδικασία της προσθήκης νοήματος στις δομές που δημιουργούνται κατά την συντακτική ανάλυση. Έτσι γίνεται κατά κάποιο τρόπο μια απεικόνιση (αναφορά) ανάμεσα στις συντακτικές δομές και στα αντικείμενα του χώρου του προβλήματος.
- Ανάλυση διαλόγου (Discourse Integration): η ανάλυση διαλόγου έγκειται στο γεγονός ότι το νόημα κάποιας πρότασης μπορεί να εξαρτάται από την πρόταση που είχε προηγηθεί, για παράδειγμα σε ένα διάλογο.
- Πραγματολογική Ανάλυση (Pragmatic Analysis): είναι η απόδοση σημασίας σε γλωσσικές σημασίες τις οποίες δεν έχει καταφέρει να εκφράσει η σημασιολογία χρησιμοποιώντας λογικές διαδικασίες. Για παράδειγμα, είναι δύσκολο να ερμηνεύσουμε εάν κάποια πρόταση εκφράζει παράκληση, ευχή, διαταγή, κλπ.

3.4.2 Εισαγωγή μεθόδων NLP στην Ομαδοποίηση Κειμένου

Στην εξόρυξη γνώσης από κείμενα ασχολούμαστε με την ανεύρεση προτύπων (patterns) ανάμεσα στα σύνολα δεδομένων που περιλαμβάνονται στα κείμενα, τα οποία αποτελούν γραπτή μορφή της φυσικής γλώσσας. Είναι αυτονόητο ότι πρέπει να εισαχθούν μέθοδοι επεξεργασίας φυσικής γλώσσας στις διαδικασίες της εξόρυξης κειμένου, ώστε να μπορούν να μελετηθούν τα νοήματα των δεδομένων που περιέχονται στα κείμενα για να μπορέσουμε να εξάγουμε χρήσιμη γνώση.

Στη διαδικασία της ομαδοποίησης κειμένου, οι μέθοδοι Natural Language Processing ενδιαφέρουν κυρίως το στάδιο της προ-επεξεργασίας, κατά το οποίο γίνεται η εξαγωγή των γνωρισμάτων και η δημιουργία του πίνακα όρων-εγγράφων για τη διανυσματική αναπαράσταση των αρχείων κειμένου. Εδώ αντιλαμβανόμαστε ότι η επεξεργασία φυσικής γλώσσας είναι χρήσιμη σε όλα τα στάδια: στην επεξεργασία των ακατέργαστων κειμένων για την εύρεση των χαρακτηριστικών τους, στην επιλογή των σημαντικότερων χαρακτηριστικών κάθε αρχείου, στην εύρεση χαρακτηριστικών τα οποία καθορίζουν ομοιότητα μεταξύ των κειμένων, στην απόρριψη άσχετων γνωρισμάτων. Στην επόμενη ενότητα θα αναφερθούμε σε μεθόδους NLP στο στάδιο της προ-επεξεργασίας για την ομαδοποίηση αρχείων κειμένου.

3.5 Το στάδιο της προ-επεξεργασίας

Κατά τη διαδικασία της προ-επεξεργασίας των κειμένων προς ομαδοποίηση, στόχος είναι η εξαγωγή των γνωρισμάτων (features) προκειμένου να δημιουργηθούν στη συνέχεια οι

διανυσματικές αναπαραστάσεις των αρχείων κειμένου. Στην ενότητα αυτή θα δούμε πώς μπορούμε να κάνουμε την εξαγωγή χαρακτηριστικών χρησιμοποιώντας μεθόδους NLP.

Όπως αναφέραμε και προηγουμένως, εφόσον χρησιμοποιούμε το Μοντέλο Διανυσματικού Χώρου για την αναπαράσταση κειμένου, μας ενδιαφέρει στο στάδιο της προ-επεξεργασίας να κατασκευάσουμε το πίνακα όρων-εγγράφων. Το πρώτο βήμα για την κατασκευή του πίνακα εγγραφών είναι η γλωσσική επεξεργασία (tokenization) των ακατέργαστων μέχρι αυτή τη στιγμή κειμένων προκειμένου να βρεθούν οι λέξεις (τα γνωρίσματα) που απαρτίζουν κάθε κείμενο και να διαχωριστούν από μη λεκτικά σύμβολα όπως είναι τα σημεία στίξης. (Εδώ σημειώνουμε την παρατήρηση ότι συνήθως οι πίνακες που προκύπτουν από την παραπάνω αναπαράσταση είναι εξαιρετικά αραιοί, δεδομένου ότι υπολογίζονται τα βάρη κάθε όρου για όλα τα αρχεία κειμένου, και αυτά στα οποία δεν περιέχονται, που συνήθως είναι τα περισσότερα.)

Προκειμένου να μειωθεί η πολυπλοκότητα χρόνου και χώρου, η μείωση της διάστασης των διανυσμάτων των όρων στο αρχικό μοντέλο διανυσματικού χώρου είναι σίγουρα απαραίτητη. Παράλληλα, εφόσον είμαστε στη διαδικασία εξαγωγής γνωρισμάτων, προκειμένου να επιτύχουμε στη συνέχεια μια ικανοποιητική ομαδοποίηση των εγγράφων, είναι υψίστης σημασίας να κάνουμε εκείνη την επιλογή γνωρισμάτων που δε θα δημιουργήσει θόρυβο στη διαδικασία της ομαδοποίησης, αλλά θα βοηθήσει αποτελεσματικά στη διεξαγωγή της διαδικασίας και στην απόδοσή της.

Από τις πιο γνωστές τεχνικές που χρησιμοποιούνται στη διαδικασία εξαγωγής γνωρισμάτων κατά την ομαδοποίηση κειμένου, έχουμε τις ακόλουθες:

Καταρχάς, θέλουμε να απορρίψουμε τους όρους που εμφανίζονται πολύ συχνά στα κείμενα χωρίς να προσδίδουν κάποια θεματική ιδιότητα στο αρχείο (αφαίρεση stop-words), έτσι ώστε να αποφύγουμε όρους που θα είχαν μεγάλο βάρος χωρίς να είναι σημαντικοί.

Στη συνέχεια, δεδομένου ότι υπάρχουν λέξεις όμοιας ρίζας αλλά με διαφορετική μορφή (λόγω διαφορετικής κλίσης, καταλήξεων κλπ) και οι οποίες παρουσιάζουν προφανώς ομοιότητα, συγχωνεύουμε τους όρους αυτούς εξάγοντας τη ρίζα κάθε λέξης (stemming).

Επιπλέον, μας ενδιαφέρει να εντοπίσουμε όσο το δυνατόν αποτελεσματικότερα τους όρους που θεωρούνται σημαντικοί για την ομαδοποίηση. Αυτό γίνεται με την απόδοση βάρους σε κάθε όρο (term weighting), με βάση τη συχνότητα εμφάνισής τους σε κάθε έγγραφο ξεχωριστά αλλά και στη συλλογή των εγγράφων συνολικά (ορισμένες μέθοδοι απόδοσης βάρους αναλύθηκαν στην ενότητα 3.1.4 σχετικά με την διανυσματική αναπαράσταση κειμένου), και στη συνέχεια με την αφαίρεση όρων οι οποίοι λόγω της πολύ σπάνιας εμφάνισής τους συνολικά στα έγγραφα δεν έχουν ιδιαίτερο ενδιαφέρον για την εύρεση ομοιότητας μεταξύ κειμένων (pruning).

Ως εδώ μπορούμε να δούμε ότι η γλωσσική επεξεργασία (tokenization) των ακατέργαστων κειμένων, καθώς και η διαδικασία του stemming, αποτελούν μεθόδους επεξεργασίας φυσικής γλώσσας σε μορφολογικό επίπεδο. Ωστόσο, η ομοιότητα των κειμένων έγκειται κυρίως στην εύρεση κοινού νοηματικού περιεχομένου, γεγονός το οποίο μας προτρέπει να προχωρήσουμε σε μετέπειτα επεξεργασία των κειμένων, σε συντακτικό και σημασιολογικό επίπεδο, ώστε να αποκτήσουμε μέσω αυτής μια νοηματική προσέγγιση του περιεχομένου των αρχείων, προκειμένου να μελετήσουμε την ομοιότητά τους. Αναφέρουμε ακολούθως ορισμένες επιπλέον μεθόδους οι οποίες μελετούνται τα τελευταία χρόνια όσον αφορά στην προ-επεξεργασία των κειμένων, σχετικές με τη συντακτική και τη σημασιολογική ανάλυσή τους.

Όπως αναφέραμε πριν, στην προσπάθεια να μειώσουμε τη διάσταση των διανυσμάτων στο μοντέλο διανυσματικού χώρου με τη μείωση των όρων, εκτελούμε συγχώνευση όρων που παρουσιάζουν ομοιότητα λόγω του κοινού τους μορφήματος. Ωστόσο, η συγχώνευση αυτή μπορεί να αποκτήσει πέρα από μορφολογική, και σημασιολογική σημασία. Δηλαδή, να επιτελούμε επιπλέον συγχώνευση όρων οι οποίοι έχουν κοινό νόημα. Αυτή η διαδικασία μπορεί να γίνει με τη χρήση κάποιου λεξικού και την εύρεση συνώνυμων λέξεων. Έτσι, μπορούμε αναλύοντας συντακτικά τα κείμενα να βρούμε κάθε όρος τι μέρος του λόγου είναι (part of speech tagging) και στη συνέχεια με τη χρήση κάποιου λεξικού να προχωρήσουμε σε σημασιολογική ανάλυση κάθε πρότασης προκειμένου να αποσαφηνίσουμε το νόημα κάθε λέξης (word sense disambiguation), με αποτέλεσμα να είναι εύκολος ο εντοπισμός όρων με όμοιο νοηματικό περιεχόμενο και η μετέπειτα συγχώνευσή τους.

Ας δούμε τώρα λίγο εκτενέστερα τις μεθόδους που προαναφέραμε ως στάδια της προ-επεξεργασίας των αρχείων κειμένου που πρόκειται να ομαδοποιηθούν.

3.5.1 Γλωσσική προ-επεξεργασία (Λεξικολογική Ανάλυση)

Σε αυτή τη διαδικασία (η οποία ονομάζεται και Tokenization) εντοπίζουμε όλες τις λέξεις του κειμένου και απομακρύνουμε μη λεκτικά σύμβολα (όπως τα σημεία στίξης) ή άλλα (όπως αριθμούς) τα οποία ενδεχομένως να θεωρούμε ότι δεν έχουν ιδιαίτερη σημασία στα έγγραφα. Ακόμη, σε αυτή τη φάση μπορούμε να κάνουμε και case folding, δηλαδή να μετατρέψουμε όλους τους χαρακτήρες ενός εγγράφου στο ίδιο σχήμα, δηλαδή αναπαράσταση όλων των χαρακτήρων σε μια τυποποιημένη μορφή, με μικρά ή κεφαλαία γράμματα.

3.5.2 Αφαίρεση των stop-words

Τα stop-words είναι λέξεις που από γλωσσολογική άποψη δεν φέρουν χρήσιμη πληροφορία. Περιλαμβάνουν τις προθέσεις, τα άρθρα, τις αντωνυμίες κλπ, που είναι προφανές ότι θα παρουσιάζουν μεγάλη συχνότητα εμφάνισης σε πολλά έγγραφα και θα φέρουν πολύ μικρή πληροφορία για το περιεχόμενο του εγγράφου στο οποίο εμφανίζονται. Έτσι, η αφαίρεση των

stop-words αποτελεί την αφαίρεση μη περιγραφικών λέξεων από τα αρχεία κείμενου διατηρώντας το νόημα των προτάσεων. Κατ' αυτόν τον τρόπο μπορούμε να μειώνουμε το θόρυβο στην προσπάθεια εύρεσης ομοιότητας και να διατηρούμε τις ουσιώδεις λέξεις, κάνοντας τη μετέπειτα εργασία περισσότερο χρήσιμη και αποτελεσματική. Οι stop-words εξαρτώνται από τη φυσική γλώσσα κι έτσι είναι διαφορετικές για κάθε γλώσσα στην οποία είναι γραμμένο ένα κείμενο.

Η αφαίρεση των stop-words γίνεται συνήθως με τη χρήση κάποιας λίστας με λέξεις που θεωρούνται stop-words. Φυσικά υπάρχουν πολλές τέτοιες λίστες που πέρα από τη γλώσσα εξαρτώνται και από το πεδίο εφαρμογής της ομαδοποίησης σε κείμενα. Για παράδειγμα σε κάποιο συγκεκριμένο πεδίο όπως η ιατρική θα υπάρχουν επιπλέον λέξεις που θεωρούνται stop-words, συνεπώς εάν έχουμε να κάνουμε ομαδοποίηση μόνο σε ιατρικά κείμενα, η χρήση μιας εξειδικευμένης λίστας θα βελτιώσει την αποτελεσματικότητα της διαδικασίας.

3.5.3 Stemming

Το stemming (στα ελληνικά στελέχωση κειμένου) είναι η διαδικασία της μετατροπής μια λέξης στη ρίζα της. Στις περισσότερες γλώσσες οι λέξεις μπορεί να εμφανίζονται με πολλούς τρόπους: για παράδειγμα τα ουσιαστικά έχουν ενικό και πληθυντικό αριθμό (πχ dog, dogs) ή και πτώσεις, ενώ τα ρήματα κλίνονται σε διαφορετικά πρόσωπα και έχουν διάφορους χρόνους (ενεστώτας, παρελθοντικός χρόνος, μέλλοντας) και φωνές (ενεργητική, παθητική, μετοχές) πχ do, does, doing. Έτσι, μπορεί λέξεις που ουσιαστικά έχουν το ίδιο ή παρόμοιο νόημα να εμφανίζονται με διαφορετική ορθογραφία. Η θεώρηση των διαφορετικών μορφών μιας λέξης ως διαφορετικούς όρους, είναι αρκετά επιβλαβής για τη διαδικασία της ομαδοποίησης όχι μόνο λόγω του χώρου στη μνήμα που καταλαμβάνουν οι όροι, αλλά και γιατί κατ' αυτόν τον τρόπο χάνεται η νοηματική σύνδεση μεταξύ των λέξεων αυτών. Είναι εύκολα λοιπόν αντιληπτό ότι το stemming είναι απαραίτητο να διεξαχθεί σε όλα τα αρχεία κειμένου πριν το στάδιο της ομαδοποίησης. Με την εύρεση κοινής ρίζας μεταξύ των λέξεων, οι αντίστοιχοι όροι συγχωνεύονται. Φυσικά, όσο χρήσιμο μπορεί να είναι το stemming στη διαδικασία της ομαδοποίησης, μπορεί να έχει και αρνητική επιρροή εάν συμβεί σε υπερβολικό βαθμό (over-stemming), ώστε να συγχωνευθούν τελικά λέξεις οι οποίες δεν έχουν κοινό νόημα με αποτέλεσμα να εισαχθεί θόρυβος και να μειωθεί η αποδοτικότητα της διαδικασίας της ομαδοποίησης μετά.

Υπάρχουν διάφοροι τρόποι εφαρμογής του stemming:

- Table lookup: με την χρησιμοποίηση ενός πίνακα που περιέχει όλες τις ρίζες και όλες τις πιθανές μορφές τους, ώστε να αναζητείται εκεί η ρίζα κάθε όρου. Ωστόσο αυτή η μέθοδος δεν είναι πρακτική όταν έχουμε να κάνουμε με μεγάλο σύνολο αρχείων κειμένου.

- Successor variety: γίνεται stemming βάσει των συχνοτήτων των ακολουθιών γραμμάτων σε ένα σώμα κειμένου. Εξετάζοντας όλες τις λέξεις δημιουργείται ο πίνακας Ποικιλίας Διαδόχων (successor variety table) ο οποίος περιέχει όλες τις πιθανές ακολουθίες γραμμάτων που εμφανίστηκαν στο κείμενο, από τον οποίο παίρνουμε στατιστικές πληροφορίες για την ποικιλία των προθεμάτων λέξεων. Στη συνέχεια, εφαρμόζοντας κάποια μέθοδο τεμαχισμού (segmentation) των λέξεων εντοπίζουμε επιλέγουμε ένα τεμάχιο ως ρίζα. Η μέθοδος αυτή ενδείκνυται για μεγάλα σύνολα αρχείων και εμπεριέχει το ρίσκο ότι λανθασμένη μέθοδος τεμαχισμού μπορεί να μας οδηγήσει σε over-stemming των αρχείων.
- Affix removal: Η αφαίρεση προσφύματος αποτελεί μια αρκετά συχνή διαδικασία, η οποία περιλαμβάνει την αφαίρεση του προθέματος (prefix) και του αποθέματος-κατάληξης (suffix) κάθε λέξης ώστε να απομένει μόνο η ρίζα της. Υπάρχουν δύο τύποι αυτής της μεθόδου:
    - Porter Algorithm: ο Porter αλγόριθμος βασίζεται στην ιδέα ότι οι καταλήξεις στην αγγλική γλώσσα (περίπου 1200) δημιουργούνται από συνδυασμούς μικρότερων και απλούστερων καταλήξεων. Αποτελείται από σύνθετους κανόνες όπως κανόνες συνθήκης/ενέργειας, κανόνες για τη ρίζα, πρότυπο κατάληξης και αντικατάσταση για την αφαίρεση του προσφύματος. Ο Porter αλγόριθμος είναι η πιο διαδεδομένη μέθοδος που εφαρμόζεται για το stemming εγγράφων.
    - Lovins Method: η μέθοδος αυτή χρησιμοποιεί μια μεγαλύτερη λίστα καταλήξεων και αφαιρεί τις καταλήξεις επαναληπτικά ακολουθώντας την προσέγγιση του «μεγαλύτερου ταιριάσματος» (longest match). Οι καταλήξεις στη λίστα συνδέονται με έναν περιορισμό από μια διαθέσιμη λίστα περιορισμών, οι οποίοι αποτρέπουν την αφαίρεση της κατάληξης μιας λέξης εφόσον πληρούνται κάποιες προϋποθέσεις. Επίσης χρησιμοποιούνται αρκετοί κανόνες, όπως ότι η παραγόμενη ρίζα θα πρέπει να έχει τουλάχιστον δύο γράμματα, κα.

3.5.4 Η χρήση συνώνυμων όρων

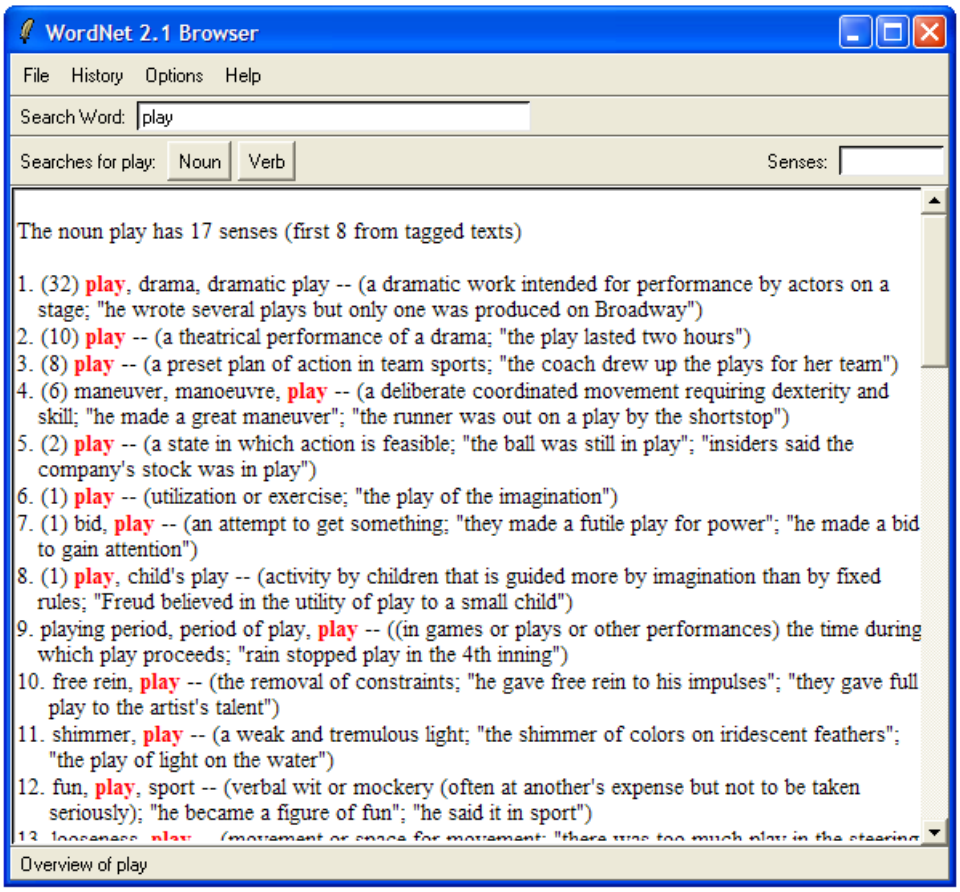
Κατά το στάδιο της εξαγωγής γνωρισμάτων πριν την ομαδοποίηση μπορούμε να συγχωνεύουμε όρους οι οποίοι έχουν όμοια νοήματα. Δύο φαινόμενα πρέπει να ληφθούν υπόψη σε αυτό το πλαίσιο: η πολυσημία και η συνωνυμία των λέξεων. Η συνωνυμία των λέξεων αναφέρεται στην ύπαρξη διαφορετικών μορφών λέξεων με όμοιο ή παρόμοιο νόημα. Η πολυσημία αναφέρεται στο συχνό φαινόμενο της ύπαρξης πολλών εννοιών για την ίδια λέξη, που παρατηρείται σε όλες τις γλώσσες. Μπορούμε συνεπώς, πέρα από την συγχώνευση όρων με κοινή ρίζα, να εντοπίζουμε με τη χρήση λεξικού όρους οι οποίοι θεωρούνται συνώνυμοι και να τους συγχωνεύουμε. Ωστόσο, η πολυσημία μπορεί να δημιουργήσει

θόρυβο στην ομαδοποίηση με τη συγχώνευση. Για να αποφευχθεί αυτό φροντίζουμε να αντιστοιχούμε τη σωστή έννοια σε κάθε λέξη, περνώντας από δύο στάδια: πρώτα με συντακτική ανάλυση, όπου βρίσκουμε τι μέρος του λόγου είναι κάθε λέξη (part-of-speech tagging), και στη συνέχεια με σημασιολογική ανάλυση, όπου κάνουμε αποσαφήνιση της έννοιας της λέξης (word sense disambiguation).

3.5.4.1 Εισαγωγή στο WordNet

Το WordNet είναι ένα ηλεκτρονικό λεξικό, αποθηκευμένο σε μια μεγάλη βάση δεδομένων, το οποίο βασίζεται στις έννοιες των λέξεων και στις σημασιολογικές σχέσεις που ισχύουν μεταξύ τους. Αναπτύχθηκε στο Princeton για την αγγλική γλώσσα, και αποτέλεσε βάση για την ανάπτυξη παρόμοιων ηλεκτρονικών σε άλλες γλώσσες καθώς και για την ανάπτυξη εφαρμογών σχετικών με την αποσαφήνιση λέξης σε συστήματα Επεξεργασίας Φυσικής Γλώσσας.

Τα ουσιαστικά, ρήματα, επίθετα και επιρρήματα ομαδοποιούνται σε σύνολα συνωνύμων(synsets), καθένα από τα οποία εκφράζει μια διακριτή έννοια. Τα synsets διασυνδέονται με σημασιολογικές (semantic) και λεξικές (lexical) σχέσεις, έτσι ώστε να αποτελούν ένα δίκτυο εννοιών, δηλαδή το WordNet. Το δίκτυο αυτό που αποτελείται από νοηματικά σχετιζόμενες λέξεις και έννοιες μπορεί να προσπελαστεί από ένα browser. Το WordNet είναι πρόγραμμα ανοιχτού κώδικα και διανέμεται δωρεάν μαζί με τον κώδικά του καθώς και με ένα γραφικό περιβάλλον, έτσι ώστε να μπορεί να χρησιμοποιηθεί ως ανεξάρτητο ηλεκτρονικό λεξικό από τους χρήστες, να συνδεθεί με άλλες προγραμματιστικές εφαρμογές από προγραμματιστές, αλλά επίσης και να μπορεί να δεχθεί επέκταση των ήδη υπαρχουσών λειτουργιών του.



Εικόνα 3-16 : Το WordNet διαθέτει γραφικό περιβάλλον για λειτουργία ως ανεξάρτητο ηλεκτρονικό λεξικό

Παρακάτω ορίζουμε μερικούς όρους οι οποίοι μας βοηθούν στην κατανόηση των σχέσεων που υπάρχουν μεταξύ των λέξεων στο WordNet καθώς και της δομής και της λειτουργίας του.

#Synset: Είναι ένα σύνολο συνωνύμων, δηλαδή ένα σύνολο λέξεων οι οποίες σε ένα συγκεκριμένο περιβάλλον μπορούν να χρησιμοποιηθούν η μία στη θέση της άλλης. Κάθε synset συνοδεύεται από ένα gloss.

#Gloss: Είναι ο επεξηγηματικός ορισμός και/ή παράδειγμα προτάσεων για ένα synset. Για παράδειγμα το σύνολο {car, auto, automobile, machine, motorcar} αποτελεί ένα synset, με gloss: “4-wheeled motor vehicle; usually propelled by an internal combustion engine”.

#Sense: Είναι μια έννοια μιας λέξης στο WordNet. Κάθε sense μιας λέξης εντάσσεται σε διαφορετικό synset.

#SenseKey: Αποτελεί το κλειδί μιας έννοιας, είναι η απαραίτητη πληροφορία για να βρεθεί μια έννοια στην βάση δεδομένων του WordNet.

#Lemma: είναι η λέξη γραμμένη σε ένα ASCII κείμενο με μικρά γράμματα που έτσι όπως είναι καταγεγραμμένη στα αρχεία ευρετηρίου της βάσης δεδομένων του WordNet. Έτσι κάθε Lemma αντιστοιχεί σε μια ακριβώς λέξη που την περιγράφει μονοσήμαντα.

#Lexical pointer: ένας λεξικός δείκτης δηλώνει τη σχέση μεταξύ λέξεων στα synsets (μορφές λέξεων)

#PartOfSpeech: Το WordNet ορίζει το μέρος του λόγου μιας λέξης ως ουσιαστικό (noun), ρήμα (verb), επίθετο (adjective) ή επίρρημα (adverb). Μία λέξη μπορεί να αντιστοιχεί σε περισσότερα από ένα μέρη του λόγου.

#Collocation: είναι ένα String που αποτελείται από δύο ή περισσότερες λέξεις, τις οποίες όμως νοηματικά τις αντιμετωπίζουμε σαν μία έννοια (πχ take in, fountain pen, κλπ).

Στο WordNet, ορίζονται διάφορες σημασιολογικές σχέσεις μεταξύ των λέξεων, και άρα των synsets στα οποία ανήκουν. Οι πιο συχνά χρησιμοποιούμενες είναι οι ακόλουθες::

Υπωνυμία (hyponymy) / Υπερωνυμία (hypernymy): Όταν λέμε ότι το Α είναι υπερώνυμο (**hypernym**) του Β ή ότι το Β είναι ένα υπώνυμο (**hyponym**) του Α, εννοούμε ότι το Β είναι ένα είδος του Α. Για παράδειγμα, το synset {ασθενοφόρο, νοσοκομειακό} είναι hyponym του {αυτοκίνητο}, δηλαδή το ασθενοφόρο είναι ένα είδος αυτοκινήτου.
Μερωνυμία (meronymy) / Ολωνυμία (holonymy): Όταν λέμε ότι το Α είναι ολώνυμο (**holonym**) του Β ή ότι το Β είναι μερώνυμο (**meronym**) του Α, εννοούμε ότι το Β είναι μέρος του Α. Για παράδειγμα, το {προφυλακτήρας} είναι meronym του {αυτοκίνητο}, δηλαδή ο προφυλακτήρας είναι μέρος του αυτοκινήτου.
Το WordNet αποτελεί ένα πολύ εύχρηστο εργαλείο για την αντιμετώπιση της πολυσημίας και της συνωνυμίας μεταξύ των λέξεων, γι' αυτό και τελευταία έχει αρχίσει να χρησιμοποιείται και στον τομέα της ομαδοποίησης κειμένου.

3.5.4.2 Part-of-Speech Tagging

Το part-of-speech tagging (POS tagging, επισημείωση μερών του λόγου) είναι η διαδικασία της αντιστοίχισης λέξεων σε ένα κείμενο με ένα συγκεκριμένο μέρος του λόγου, με βάση τον ορισμό της λέξης καθώς και το πλαίσιο στο οποίο είναι τοποθετημένη, δηλαδή τη σχέση της λέξης με γειτονικές και σχετιζόμενες λέξης σε μια φράση, πρόταση ή παράγραφο. Το part-of-speech tagging αποτελεί σήμερα μέρος του τομέα της Υπολογιστικής Γλωσσολογίας και της Επεξεργασίας Φυσικής Γλώσσας.

Ο Part-Of-Speech Tagger (POS Tagger) είναι ένα κομμάτι λογισμικού το οποίο διαβάζει κείμενο σε κάποια γλώσσα, το χωρίζει σε τμήματα (tokens) καθένα από το οποίο αντιστοιχεί σε κάθε λέξη, και αναθέτει μια ετικέτα με το μέρος του λόγου (ρήμα, ουσιαστικό, κλπ) σε κάθε token. Υπάρχουν διάφορα σύνολα ετικετών part-of speech τα οποία χρησιμοποιούνται από τους POS taggers για ανάθεση POS στις λέξεις, με πιο συνηθισμένο το Penn Treebank POS tag set, το οποίο παρουσιάζουμε ακολούθως:

PENN TREEBANK TAGSET

CC	Coordinating conjunction e.g. and,but,or...
CD	Cardinal Number
DT	Determiner
EX	Existential *there*
FW	Foreign Word
IN	Preposision or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List Item Marker
MD	Modal e.g. can, could, might, may...
NN	Noun, singular or mass
NNP	Proper Noun, singular
NNPS	Proper Noun, plural
NNS	Noun, plural
PDT	Predeterminer e.g. all, both ... when they precede an article
POS	Possessive Ending e.g. Nouns ending in 's
PRP	Personal Pronoun e.g. I, me, you, he...
PRP$	Possessive Pronoun e.g. my, your, mine, yours...
RB	Adverb -Most words that end in -ly as well as degree words like quite, too and very
RBR	Adverb, comparative -Adverbs with the comparative ending -er, with a strictly comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol -Should be used for mathematical, scientific or technical symbols
TO	*To*
UH	Interjection e.g. uh, well, yes, my...
VB	Verb, base form - subsumes imperatives, infinitives and subjunctives
VBD	Verb, past tense - includes the conditional form of the verb to be
VBG	Verb, gerund or persent participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner, eg which, and *that* when it is used as a relative pronoun
WP	Wh-pronoun e.g. what, who, whom...
WP$	Possessive wh-pronoun
WRB	Wh-adverb e.g. how, where why

Punctuation Tags
#
$
"
(
)
,
.
:
``

Το part-of speech tagging γίνεται είτε με βάσει προϋπάρχοντα σύνολα κειμένων (trained data) με βάση τα οποία γίνεται η ανάθεση ετικετών part-of-speech χρησιμοποιώντας πιθανότητες, είτε και με «ανεπίβλεπτο» (unsupervised) tagging, με το οποίο οι taggers χρησιμοποιούν σύνολο κειμένων χωρίς ετικέτες και, παρατηρώντας πρότυπα στη χρήση των λέξεων, παράγουν κατηγορίες μερών του λόγου μόνοι τους.

Μερικοί πρόσφατοι γνωστοί αλγόριθμοι part-of-speech tagging περιλαμβάνουν τον Virtebi algorithm, τον Brill Tagger, και τον Baum-Welch algorithm (επίσης γνωστός ως μπρος-πίσω αλγόριθμος).

3.5.4.3 Word Sense Disambiguation

Ο όρος αποσαφήνιση της έννοιας μιας λέξης (Word Sense Disambiguation - WSD) αναφέρεται στη διαδικασία με την οποία προσδιορίζουμε για κάθε αμφίσημη λέξη ενός κειμένου τη σωστή έννοιά της μέσα από μια πληθώρα νοημάτων, ώστε να ταιριάζει στο πλαίσιο των λέξεων που την περιβάλλουν (σε μια φράση, πρόταση, παράγραφο). Η επιλογή της έννοιας της λέξης γίνεται συνήθως μεταξύ των δυνατών εννοιών που έχουν αντιστοιχηθεί στη συγκεκριμένη λέξη σε ένα ηλεκτρονικό λεξικό, όπως το WordNet.

Για παράδειγμα, στις προτάσεις:

(1) she needed a fix of chocolate

(2) she fixed her TV set

(3) you have to fix the variables now

συναντάμε τη λέξη fix ως ουσιαστικό (1) και ως ρήμα (2), να έχει διαφορετικές έννοιες σε κάθε πρόταση:

(1) fix: something craved, especially an intravenous injection of a narcotic drug

(2) fix: repair, mend, fix, bushel, doctor, furbish up, restore, touch on

(3) fix: specify, set, determine, fix, limit

Για την ακρίβεια, στο WordNet η λέξη fix έχει πέντε έννοιες ως ουσιαστικό και εννιά έννοιες ως ρήμα, οπότε αντιλαμβανόμαστε την πολυσημία της, η οποία απαντάται στις περισσότερες λέξεις μιας γλώσσας.

Οι αλγόριθμοι που ασχολούνται με τη διαδικασία του Word Sense Disambiguation μπορούν να χωριστούν σε δύο κατηγορίες: στους αλγορίθμους που απαιτούν κάποιο σύνολο προ-χαρακτηρισμένων εγγράφων για προ-εκπαίδευση (trained data) και στους αλγορίθμους που βασίζονται σε λεξικογραφικές πηγές (συνήθως ηλεκτρονικά λεξικά) και επεξεργάζονται εξ' αρχής κάποιο σύνολο εγγράφων. Γνωστοί αλγόριθμοι WSD είναι η μέθοδος των Brown κ.ά., ο αλγόριθμος των Gale κ.ά., η μέθοδος του Yarowsky και ο αλγόριθμος του Lesk.

Τελικά, έπειτα από τον προσδιορισμό της έννοιας κάθε λέξης στα αρχεία κειμένων, είναι εύκολο να προσδιοριστούν ποιοι όροι είναι συνώνυμοι μεταξύ τους με τη χρήση κάποιου ηλεκτρονικού λεξικού (για παράδειγμα στο WordNet λέξεις που ανήκουν στο ίδιο synset θα θεωρούνται συνώνυμες) και να συγχωνευθούν κατά τη δημιουργία του πίνακα-εγγράφων.

3.5.5 Στάθμιση (term weighting)

Κατά τη διάρκεια της προ-επεξεργασίας των εγγράφων και της δημιουργίας των πίνακα εγγράφων, αποδίδονται βάρη στους όρους ανάλογα με την συχνότητα εμφάνισής τους στο εκάστοτε κείμενο αλλά και στο σύνολο των αρχείων κειμένου. Οι διάφοροι μέθοδοι απόδοσης βάρους μελετήθηκαν νωρίτερα στην ενότητα 3.1.4 γι' αυτό και δε θα επαναληφθούν εδώ.

3.5.6 Pruning

Τέλος, κατά την προ-επεξεργασία των αρχείων κειμένου, υπάρχει η δυνατότητα της διαδικασίας pruning (στα ελληνικά μεταφράζεται ως κλάδεμα), που αποτελεί την αφαίρεση κάποιων όρων σε σχέση με την συχνότητα εμφάνισής τους στα έγγραφα. Με το σκεπτικό ότι όροι που εμφανίζονται σπάνια στο σύνολο των εγγράφων δεν βοηθούν για τον προσδιορισμό των κατάλληλων ομάδων, αλλά μπορούν ακόμη και να προσθέσουν θόρυβο στον υπολογισμό της απόστασης μεταξύ των term vectors μειώνοντας έτσι τη γενική απόδοση της ομαδοποίησης, συνίσταται ότι η αφαίρεση των σπάνιων όρων μπορεί να επηρεάσει θετικά τα αποτελέσματα.

4 *Περιγραφή Συστήματος*

Στο προηγούμενο τοποθετήσαμε την έρευνά μας στο πλαίσιο της Εξόρυξης Κειμένου και της Επεξεργασίας Φυσικής Γλώσσας. Τώρα είμαστε σε θέση να περιγράψουμε τις μεθόδους και τεχνικές που αποφασίσαμε να χρησιμοποιήσουμε για να επιτύχουμε ομαδοποίηση αρχείων κειμένου, και στη συνέχεια να μελετήσουμε κατά πόσον αντιστοιχεί το σύστημα που αποφασίσαμε να φτιάξουμε στις λειτουργικές απαιτήσεις του συστήματος διαχείρισης καινοτομίας IDeM και στην αρχιτεκτονική του.

4.1 Το σύστημα ομαδοποίησης αρχείων κειμένου

Στην ενότητα αυτή θα περιγράψουμε ένα σύστημα για ομαδοποίηση αρχείων κειμένου που αποφασίσαμε να κατασκευάσουμε λόγω της ανάγκης για ημιαυτόματη ομαδοποίηση ιδεών. Το σύστημα αναπτύχθηκε έτσι ώστε να είναι σε θέση να λειτουργεί ανεξάρτητα από οποιαδήποτε άλλη εφαρμογή, μέσω της εκτέλεσης ορισμένων εκτελέσιμων αρχείων, αλλά και να υπάρχει η δυνατότητα άμεσης κλήσης των κλάσεων του συστήματος από κάποια άλλη εφαρμογή ή σύστημα και η διασύνδεση των δεδομένων και των αποτελεσμάτων.

Βάση για την ανάπτυξη του συστήματος ομαδοποίησης αρχείων αποτέλεσε το πρόγραμμα Weka, το οποίο περιγράφηκε στο προηγούμενο κεφάλαιο και εφαρμόζεται πολύ συχνά σε ερευνητικές εφαρμογές εξόρυξης δεδομένων. Το Weka, ως πρόγραμμα εξόρυξης γνώσης από δεδομένα, περιέχει τη λειτουργία της ομαδοποίησης αρχείων κειμένου. Έτσι αποφασίσαμε να ξεκινήσουμε τον ανοιχτό κώδικα του Weka, και συγκεκριμένα της έκδοσης 3.4, και να τον επεκτείνουμε, προσθέτοντας ή αναπτύσσοντας επιπλέον τις λειτουργίες επεξεργασίας φυσικής γλώσσας που έχουν να κάνουν με την εξαγωγή χαρακτηριστικών στο στάδιο της προ-επεξεργασίας, προκειμένου να βελτιώσουμε την απόδοση της ομαδοποίησης. Ως ηλεκτρονικό λεξικό για τις λειτουργίες επεξεργασίας φυσικής γλώσσας χρησιμοποιήσαμε το

WordNet 3.0, στο οποίο αναφερθήκαμε στο προηγούμενο κεφάλαιο, λόγω της ευρείας εφαρμογής του και της δωρεάν διανομής τόσο του προγράμματος όσο και του κώδικα. Επειδή το Weka είναι γραμμένο σε Java, χρησιμοποιήσαμε ένα Java API του WordNet, και συγκεκριμένα το JWI (MIT Java Interface to WordNet), το οποίο αναπτύχθηκε από τον Mark Finlayson.

Οι επιπλέον λειτουργίες που προσθέσαμε στο Weka έχουν να κάνουν με:

- Part-of-speech tagging
- Αφαίρεση stopwords
- Word Sense Disambiguation
- Εύρεση συνωνύμων και συγχώνευση όρων με όμοιο νόημα
- Stemming
- Μελέτη διαφορετικών weights και διαφορετικών μέτρων ομοιότητας

Θα παρουσιάσουμε λοιπόν ακολούθως τη διαδικασία που προτείνουμε για την ομαδοποίηση αρχείων κειμένου με χρήση του Weka:

4.1.1 Μετατροπή σε αρχείο arff

Τα αρχεία κειμένου που πρόκειται να ομαδοποιηθούν είναι αποθηκευμένα σε μορφή κειμένου *.txt σε ένα κατάλογο. Επειδή όπως προαναφέρθηκε το Weka δέχεται ως είσοδο μόνο flat files, χρειάζεται τα αρχεία κειμένου να μετατραπούν σε ένα σύνολο δεδομένων (data set) αποθηκευμένο σε αρχείο της μορφής arff (η μορφή αρχείων του Weka) το οποίο θα αποτελείται από instances, ένα για κάθε αρχείο κειμένου, που περιλαμβάνουν δύο attributes: το όνομα του αρχείου και το κείμενο που περιέχεται στο αρχείο σε μορφή μιας συμβολοσειράς (string). Η διαδικασία της μετατροπής είναι ιδιαίτερα απλή, και γίνεται μέσω της java κλάσης TextDirectoryToArff.

4.1.2 Part-of-speech tagging

Ωστόσο, παράλληλα με τη διαδικασία TextDirectoryToArff, η οποία θα μπορούσαμε να πούμε ότι αποτελεί ένα στάδιο προ-επεξεργασίας των αρχείων κειμένου πριν την προ-επεξεργασία, εκτελείται και η διαδικασία Part-Of-Speech Tagging, έτσι ώστε κάθε κείμενο, που αποθηκεύεται στο αρχείο arff σε μια συμβολοσειρά, να αποτελείται από λέξεις που συνοδεύονται ήδη από το part-of-speech τους στη μορφή "λέξη/POS". Η επιλογή να γίνει το Part-Of-Speech Tagging σε αυτή τη φάση έγινε κυρίως γιατί κρίθηκε ότι η επεξεργασία κάθε αρχείου κειμένου αμέσως πριν εισαχθεί ως instance στο αρχείο arff, θα καταλάμβανε λιγότερο χώρο μνήμης και θα εκτελούταν ταχύτερα από αν γινόταν κατά τη διάρκεια του pre-processing.

Ως part-of-speech tagger χρησιμοποιήσαμε τον Stanford Log-linear Part-Of-Speech Tagger, ο οποίος έχει κατασκευαστεί από το Stanford Natural Language Processing Group. Αυτός ο POS Tagger χρησιμοποιεί ένα σώμα εκπαίδευσης (training set), δηλαδή ένα σύνολο από κείμενα στα οποία έχει ήδη γίνει επισημείωση των μερών του λόγου και το οποίο αποτελεί βάση για την επιλογή των ετικετών στο σώμα ελέγχου (δηλαδή τα κείμενα που μας ενδιαφέρουν να κάνουμε επισημείωση των μερών του λόγου), και είναι βασισμένος στα συστήματα Μέγιστης Εντροπίας (Maximum Entropy).

Στα συστήματα Μέγιστης Εντροπίας, για να εκτιμήσουμε την πιθανότητα μία ακολουθία ετικετών $t_1,\ldots t_n$ να αντιστοιχεί σε μία ακολουθία λέξεων $w_1,\ldots,w_n$:

$$p(t_1 \ldots t_n \mid w_1 \ldots w_n) = \prod_{i=1}^{n} p(t_i \mid t_1 \ldots t_{i-1}, w_1 \ldots w_n) \approx \prod_{i=1}^{n} p(t_i \mid h_i)$$

(με h_i τα συμφραζόμενα της λέξης w_i)

επιλέγεται η κατανομή πιθανότητας η οποία έχει μέγιστη εντροπία από όλες τις κατανομές που ικανοποιούν συγκεκριμένους περιορισμούς. Οι περιορισμοί αυτοί προκύπτουν από στατιστικά στοιχεία που εξάγονται από τα δεδομένα εκπαίδευσης, που συνήθως εκφράζονται ως οι αναμενόμενες τιμές κατάλληλων συναρτήσεων που αναπαριστούν ιδιότητες λέξεων οι οποίες εξαρτώνται από τα συμφραζόμενα και τις ετικέτες. Αν για παράδειγμα, θέλουμε να θέσουμε στο μοντέλο τον περιορισμό να προσδίδει στη λέξη "care" την ετικέτα verb ή noun με την ίδια συχνότητα που η λέξη έχει επισημειωθεί ως ρήμα ή ουσιαστικό στο σώμα εκπαίδευσης, πρέπει να ορίσουμε τις ιδιότητες:

1. $f_1(h,t) = 1$ αν w_i=care και t=verb και

2. $f_2(h,t) = 1$ αν w_i=care και t=noun.

Έτσι, αν στο σώμα εκπαίδευσης η λέξη "care" παρατηρείται επισημειωμένη ως ρήμα ή ως ουσιαστικό με μία αναλογία 7/3 οι παραπάνω ιδιότητες θα ωθήσουν το μοντέλο να επισημειώνει τη λέξη "care" στα νέα κείμενα (του σώματος ελέγχου) ως ουσιαστικό ή ρήμα μα την ίδια αναλογία.

Ο αλγόριθμος του POS Tagger χρησιμοποιεί τα σύνολα ετικετών και άλλες χρήσιμες πληροφορίες (μορφολογικά χαρακτηριστικά, μήκος λέξης, λεξικό καταλήξεων, κα) από τα συμφραζόμενα που βρίσκονται τόσο πριν όσο και μετά την εκάστοτε λέξη για την οποία ψάχνει να βρει ετικέτα, χρησιμοποιώντας μια αναπαράσταση δικτύου εξάρτησης διπλής κατεύθυνσης (bidirectional dependency network) Επίσης, με την ευρεία χρήση λεξικών γνωρισμάτων (συμπεριλαμβανομένης της από κοινού υπόθεσης για πολλαπλές συνεχείς λέξεις), την αποτελεσματική χρήση των προηγουμένων στοιχείων όπως γίνεται σε conditional loglinear models, καθώς και μοντελοποίηση των άγνωστων χαρακτηριστικών λέξεων, η

ακρίβεια του POS Tagger είναι ιδιαίτερα υψηλή (μετρήθηκε ακρίβεια 97.24% στη δοκιμή με το σύνολο κειμένων Penn Treebank WSJ).

Στο ακόλουθο παράδειγμα μπορούμε να δούμε πώς δείχνει ένα αρχείο κειμένου πριν και μετά την επισημείωση των μερών του λόγου των λέξεών του:

Το αρχείο πριν την εκτέλεση της κλάσης TextDirectoryToArff:

```
In this essay I am going to evaluate my ability to use the English language. I am going to assess my strengths and weaknesses in the four skills of listening, reading, speaking and writing. Eight years ago I moved to the US and I stayed there for two years. The evaluation of my English is based on how competent I feel today, at this point. I must honestly say that I have lost a lot of my confidence in the English language since my days in the US and that includes all four skills more or less.
```

Η αναπαράσταση του αρχείου ως instance μέσα στο αρχείο arff:

```
@relation 'text_files_in_.\\topic1'

@attribute filename string
@attribute contents string

@data
aa10100.a1.txt,'In/IN this/DT essay/NN I/PRP am/VBP going/VBG to/TO evaluate/VB my/PRP$ ability/NN to/TO use/VB the/DT English/NNP language/NN ./. I/PRP am/VBP going/VBG to/TO assess/VB my/PRP$ strengths/NNS and/CC weaknesses/NNS in/IN the/DT four/CD skills/NNS of/IN listening/NN ,/, reading/VBG ,/, speaking/VBG and/CC writing/VBG ./. Eight/CD years/NNS ago/IN I/PRP moved/VBD to/TO the/DT US/NNP and/CC I/PRP stayed/VBD there/EX for/IN two/CD years/NNS ./. The/DT evaluation/NN of/IN my/PRP$ English/JJ is/VBZ based/VBN on/IN how/WRB competent/JJ I/PRP feel/VBP today/NN ,/, at/IN this/DT point/NN ./. I/PRP must/MD honestly/RB say/VB that/IN I/PRP have/VBP lost/VBN a/DT lot/NN of/IN my/PRP$ confidence/NN in/IN the/DT English/NNP language/NN since/IN my/PRP$ days/NNS in/IN the/DT US/NNP and/CC that/DT includes/VBZ all/DT four/CD skills/NNS more/RBR or/CC less/RBR ./. '
```

Έτσι, τελικά καταλήγουμε σε ένα αρχείο της μορφής arff το οποίο περιέχει ως instances όλα τα αρχεία κειμένου που θέλουμε να ομαδοποιήσουμε, με τις λέξεις τους αντιστοιχισμένες με το POS τους στη μορφή "λέξη/POS".

4.1.3 Η κλάση StringToWordVector

Για την αναπαράσταση των αρχείων κειμένου στο σύστημα της ομαδοποίησης, επιλέξαμε να χρησιμοποιήσουμε το μοντέλο διανυσματικού χώρου, το οποίο είχαμε περιγράψει στο προηγούμενο κεφάλαιο. Έτσι, πριν την ομαδοποίηση χρειάζεται να γίνει η κατάλληλη προ-επεξεργασία των εγγράφων κατά την οποία θα γίνει η επιλογή των γνωρισμάτων (attributes) και θα δημιουργηθεί ο πίνακας όρων-εγγράφων που χρειάζεται για τη διανυσματική αναπαράσταση των αρχείων κειμένου.

Η κλάση StringToWordVector του Weka αποτελεί την κύρια κλάση με την οποία κάνουμε pre-processing των αρχείων κειμένου (που δίνονται ως είσοδο μέσω του αρχείου arff) και επιλογή των χαρακτηριστικών. Ακολουθεί τη μέθοδο διήθησης (filter) που υπάρχει για την επιλογή του υποσυνόλου χαρακτηριστικών και στην οποία γίνεται ανεξάρτητη αποτίμηση, βασισμένη στα γενικά χαρακτηριστικά των δεδομένων, χωρίς επίβλεψη (unsupervised), δηλαδή χωρίς να χρειάζεται προηγούμενη εκπαίδευση πάνω σε άλλα δεδομένα για τα οποία έχει γίνει επιλογή χαρακτηριστικών. Η μέθοδος διήθησης επιλέγει το μικρότερο δυνατό σύνολο χαρακτηριστικών που επαρκεί για το διαχωρισμό όλων των υποδειγμάτων (instances).

Έτσι, αφού δημιουργήσουμε το αρχείο arff καλούμε τη StringToWordVector του Weka η οποία το παίρνει ως είσοδο προκειμένου να επεξεργαστεί τα instances προς ομαδοποίηση. Η StringToWordVector επεξεργάζεται κάθε instance ξεχωριστά, και εκτελεί τις λειτουργίες που θα περιγράψουμε ακολούθως, συμπληρώνοντας τον πίνακα όρων-εγγράφων.

Στην κλήση της κλάσης, ο χρήστης δίνει ως παραμέτρους το αρχείο arff που αποτελεί το αρχείο εισόδου προς επεξεργασία, καθώς και τις παραμέτρους -R 2 με τις οποίες δηλώνει ότι το εύρος των string attributes κάθε instance που επιθυμεί να επεξεργαστεί η κλάση να είναι 2, όσα δηλαδή και τα attributes που έχουν δημιουργηθεί στο αρχείο arff για κάθε instance (ένα για το όνομα του αρχείου και ένα για ολόκληρο το κείμενο). Ακόμη, υπάρχει η δυνατότητα να δώσει τις παραμέτρους -W number όπου αντί για number βάζει των αριθμό των λέξεων από κάθε έγγραφο που επιθυμεί να υποστούν επεξεργασία και να μπουν στον πίνακα όρων-εγγράφων, καθώς μπορεί να έχουμε κάποια αρχεία πολύ μεγάλα και να θεωρούμε ότι πχ οι πρώτες 500 λέξεις τους είναι επαρκείς για να βγάλουμε συμπεράσματα για το νοηματικό τους περιεχόμενο. Τέλος, υπάρχει η δυνατότητα να προσθέσει ο χρήστης και παραμέτρους σχετικές με τι είδους βάρη επιθυμεί να ανατίθενται στους όρους.

4.1.4 Γλωσσική προ-επεξεργασία (Tokenization)

Εξετάζοντας ένα-ένα κάθε instance, εξετάζουμε το αντίστοιχο κείμενο που είναι αποθηκευμένο σε ένα string attribute. Αρχικά χρειαζόμαστε να αναλύσουμε το κείμενο σε

λέξεις, αφαιρώντας τα κενά και τα σημεία στίξης. Η διαδικασία αυτή ονομάζεται tokenization. Εφόσον οι λέξεις είναι στη μορφή "λέξη/POS_tag", παράλληλα παίρνουμε το λήμμα κάθε λέξης και το χωρίζουμε από την ετικέτα του part of speech της, την οποία κρατάμε προκειμένου να χρησιμοποιήσουμε μετά. Η διαδικασία του tokenization γίνεται ανά πρόταση. Μόλις συμπληρωθεί μία πρόταση εκτελούνται οι υπόλοιπες λειτουργίες της προ-επεξεργασίας, έτσι ώστε να εκτελείται άμεσα και το word sense disambiguation που θα αναφέρουμε μετά.

4.1.5 Αποσαφήνιση της έννοιας των λέξεων (Word Sense Disambiguation)

Από τη στιγμή που έχουμε μια ολόκληρη πρόταση, καλούμε την κλάση Word Sense Disambiguator προκειμένου να αποσαφηνίσουμε την έννοια κάθε λέξης της πρότασης. Η διαδικασία αυτή γίνεται με τη χρήση του WordNet, από το οποίο βρίσκουμε όλες τις πιθανές έννοιες που υπάρχουν για την εκάστοτε λέξη ως προς το μέρος του λόγου (ουσιαστικό, ρήμα, επίθετο, επίρρημα) που βρέθηκε ότι είναι από τη διαδικασία του part-of-speech tagging. Για την αποσαφήνιση χρησιμοποιήσαμε μία παραλλαγή του αλγόριθμου του Michael Lesk, την οποία και περιγράφουμε ακολούθως:

Word Sense Disambiguation με τον αλγόριθμο του Michael Lesk

Η αποσαφήνιση είναι η διαδικασία τη εξεύρεσης της πιο κατάλληλης έννοιας μιας λέξης που χρησιμοποιείται σε μια δοσμένη πρόταση. Ο αλγόριθμος του Michal Lesk χρησιμοποιεί τους ορισμούς του λεξικού (όπως αναφέραμε στο προηγούμενο κεφάλαιο, στο WordNet για κάθε sense υπάρχει και ένα gloss, που είναι κοινό για όλα τα senses που ανήκουν στο ίδιο synset) για να αποσαφηνίσει μια αμφίσημη λέξη στο πλαίσιο μιας πρότασης. Ο κύριος στόχος αυτής της ιδέας είναι να μετρηθούν οι λέξεις που είναι κοινές μεταξύ δύο glosses. Όσο περισσότερες είναι οι λέξεις που επικαλύπτονται, τόσο πιο πολύ σχετίζονται οι δύο αντίστοιχες έννοιες.

Για να αποσαφηνιστεί μία λέξη, το gloss κάθε έννοιάς της συγκρίνεται με τα glosses όλων των υπόλοιπων λέξεων της πρότασης. Μια λέξη αντιστοιχίζεται με το sense που έχει το gloss με το μεγαλύτερο αριθμό κοινών λέξεων με τα glosses των άλλων λέξεων.

Για παράδειγμα: Για την αποσαφήνιση της φράσης "pine cone", η λέξη "pine" βλέπουμε ότι έχει δύο έννοιες:

Sense 1: kind of evergreen tree with needle-shaped leaves

Sense 2: waste away through sorrow or illness

Για τη λέξη "cone" υπάρχουν τρεις έννοιες:

Sense 1: solid body which narrows to a point

Sense 2: something of this shape whether solid or hollow

Sense 3: fruit of a certain evergreen tree

Συγκρίνοντας καθέναν από τους δύο ορισμούς εννοιών της λέξης "pine" με καθέναν από τους ορισμούς εννοιών της λέξης "cone", βρίσκεται ότι οι λέξεις "evergreen tree" εμφανίζονται σε μια έννοια σε καθεμία από τις δύο λέξεις. Συνεπώς οι δύο έννοιες θεωρούνται ότι είναι οι πιο κατάλληλες όταν οι λέξεις "pine" και "cone" χρησιμοποιούνται μαζί.

Ο προσαρμοσμένος αλγόριθμος του Michael Lesk

Ο αρχικός αλγόριθμος του Lesk χρησιμοποιεί τον ορισμό μιας λέξης και περιορίζεται στον υπολογισμό του score των επικαλύψεων. Ωστόσο, στο σύστημά μας, αποφασίσαμε να ακολουθήσουμε μια παραλλαγή αυτού του αλγορίθμου, όπως συστήνεται από τους Troy Simpson και Thanh Dao: Έτσι, χρησιμοποιούμε το WordNet, στο οποίο οι λέξεις είναι διατεταγμένες ιεραρχικά, ώστε να μη χρησιμοποιούμε μόνο τα gloss (ορισμοί) των synset αλλά επιπλέον να λαμβάνουμε υπόψη και την έννοια των σχετιζόμενων λέξεων. Επιπλέον, εφαρμόζουμε ένα νέο μηχανισμό για τον υπολογισμό των score επικαλύψεων ο οποίος δίνει ένα πιο ακριβές score από τον απλό μετρητή λέξεων του αλγορίθμου Lesk.

Για να αποσαφηνίσουμε κάθε λέξη σε μια πρόταση που έχει N λέξεις, καλούμε κάθε λέξη που πρόκειται να αποσαφηνιστεί ως λέξη-στόχο. Τα βήματα του αλγορίθμου λοιπόν είναι:

1. Επέλεξε ένα πλαίσιο: βελτιστοποιούμε τον υπολογιστικό χρόνο ώστε αν το N είναι μεγάλο, θα ορίσουμε το K πλαίσιο γύρω από τη λέξη-στόχο (ή τον k-κοντινότερο γείτονα) ως μια ακολουθία από λέξεις που ξεκινάει K λέξεις αριστερά από τη λέξη-στόχο και τελειώνει K λέξεις στα δεξιά. Αυτό θα μειώσει τον υπολογιστικό χώρο που μειώνει το χρόνο εκτέλεσης. Για παράδειγμα: αν το K είναι 4, θα έχουμε δύο λέξεις αριστερά από τη λέξη-στόχο και δύο λέξεις δεξιά.
2. Για κάθε λέξη στο επιλεγμένο πλαίσιο, αναζητούμε και κρατάμε όλες τις πιθανές έννοιες που αντιστοιχούν σε POS (part-of-speech) ρήμα και ουσιαστικό.
3. Για κάθε έννοια μιας λέξης, βρίσκουμε όλες τις πιθανές σχέσεις που μπορεί να έχει.
    - Βρίσκουμε το δικό της ορισμό (gloss) που περιλαμβάνει παραδείγματα με κείμενα που παρέχονται από το WordNet.
    - Βρίσκουμε τα gloss των synsets που συνδέονται με την έννοια αυτή μέσω σχέσεων hypernym. Εάν υπάρχουν περισσότερα από ένα hypernym για μια έννοια λέξης, τότε τα gloss όλων των hyeprnym και τα συνδέουμε αλυσιδωτά (concatenation) σε ένα string που θεωρούμε ότι αποτελεί ένα gloss.
    - Βρίσκουμε τα gloss των synset που συνδέονται με την έννοια αυτή μέσω σχέσεων hyponym, ενώ επίσης αν χρειαστεί κάνουμε συνένωση σε ένα gloss.
    - Βρίσκουμε τα gloss των synset που συνδέονται με την έννοια αυτή μέσω σχέσεων meronym, ενώ επίσης αν χρειαστεί κάνουμε συνένωση σε ένα gloss.

- Βρίσκουμε τα gloss των synset που συνδέονται με την έννοια αυτή μέσω σχέσεων troponym, ενώ επίσης αν χρειαστεί κάνουμε συνένωση σε ένα gloss.

4. Συνδυάζουμε όλα τα πιθανά ζεύξη gloss που βρέθηκαν στα προηγούμενα βήματα και υπολογίζουμε τη σχετικότητά τους (relatedness) ψάχνοντας για επικαλύψεις. Το συνολικό score είναι το άθροισμα όλων των score για κάθε ζεύγος σχέσης.

   Όταν υπολογίζουμε τη σχετικότητα μεταξύ δύο synsets s1 και s2, το ζεύγος hype-hype σημαίνει ότι το gloss του hypernym του s1 συνδυάζεται με το hypernym του s2. Το ζεύγος hype-hypo σημαίνει ότι το gloss του hypernym του s1 συνδυάζεται με το hyponym του s2.

   Στο παράδειγμα "pine cone" που προαναφέραμε, υπάρχουν 3 έννοιες του pine και 2 έννοιες του cone, οπότε μπορούμε να έχουμε συνολικά 18 πιθανούς συνδυασμούς. Ένας από αυτούς είναι ο σωστός.

   Για να υπολογίσουμε την επικάλυψη χρησιμοποιούμε ένα μηχανισμό υπολογισμού score που αντιμετωπίζει κάθε gloss σαν ένα σύνολο λέξεων, και βασίζεται στην ιδέα ότι το μήκος των λέξεων είναι αντιστρόφως ανάλογο με τη χρήση τους. Όσο μικρότερες είναι οι λέξεις τόσο πιο συχνά χρησιμοποιούνται, ενώ όσο μεγαλύτερες είναι, τόσο πιο σπάνια εμφανίζονται.

   Η μέτρηση των επικαλύψεων μεταξύ δύο string υποβιβάζεται στην επίλυση του προβλήματος εύρεσης του μεγαλύτερου κοινού sub-string με το μεγαλύτερο αριθμό συνεχόμενων λέξεων. Κάθε επικάλυψη που περιέχει N συνεχόμενες λέξεις, συνεισφέρει N2 στο score του συνδυασμού των gloss των εννοιών. Για παράδειγμα: μια επικάλυψη "ABC" έχει ένα score 3^2= 9 και δύο σκέτες επικαλύψεις "AB" και "C" έχει ένα score 2^2+1^2=5.

5. Όταν έχει υπολογιστεί το score για κάθε συνδυασμό, επιλέγουμε την έννοια που έχει το υψηλότερο score ως την πιο κατάλληλη έννοια για τη λέξη-στόχο στο επιλεγμένο πλαίσιο.

4.1.6 Αφαίρεση stop-words

Αφού παίρνουμε κάθε λέξη του κειμένου από το tokenization, τη συγκρίνουμε με μία λίστα από λέξεις (strings), οι οποίες θεωρούνται ως stopwords και είναι αποθηκευμένες σε κάποιο αρχείο. Εάν η λέξη είναι stopword, η επεξεργασία της σταματάει σε αυτό το σημείο και δεν εισάγεται στον πίνακα όρων-εγγραφών. Εάν δεν είναι, η επεξεργασία της συνεχίζεται.

Η λίστα με τα stopwords που χρησιμοποιήσαμε στο σύστημά μας παρατίθεται στο παράρτημα Α.

4.1.7 Εύρεση συνωνύμων

Αφού έχει αποσαφηνιστεί η έννοια μιας λέξης, χρησιμοποιούμε το WordNet για να βρούμε τα συνώνυμά της, δηλαδή λέξεις που έχουν όμοιο ή παρόμοιο όνομα. Παίρνουμε συνεπώς ως συνώνυμες τις λέξεις που ανήκουν στο synset της έννοιας που βρέθηκε για τη λέξη.

Κρατάμε μία λίστα με όλους τους όρους που έχουν διαβαστεί από όλα τα κείμενα, μαζί με τα συνώνυμά τους. Κατ' αυτόν τον τρόπο, κάθε φορά που βρίσκουμε μία λέξη και τα συνώνυμά της, μπορούμε να ελέγχουμε κάνοντας μια αναζήτηση στη λίστα όρων με συνώνυμα εάν υπάρχει κάποια λέξη με την οποία να είναι συνώνυμες, συγκρίνοντας τα συνώνυμά τους. Εάν έχουν κοινά συνώνυμα, οι λέξεις θεωρούνται συνώνυμες μεταξύ τους. Σ' αυτήν την περίπτωση, κάνουμε αντικατάσταση της υπό εξέταση λέξης με την λέξη που είχαμε ήδη βρει. νωρίτερα, προσθέτοντας ωστόσο τις συνώνυμες λέξεις μαζί με τα υπόλοιπα συνώνυμα της λέξης στη λίστα που έχουμε.

Ουσιαστικά δηλαδή κάνουμε συγχώνευση των όρων με παρόμοιο νόημα.

4.1.8 Stemming

Στη συνέχεια προχωρούμε στην εύρεση της ρίζας της λέξης, έτσι ώστε να πετύχουμε συγχώνευση λέξεων που έχουν κοινή ρίζα αλλά διαφορετική μορφή. Για το stemming χρησιμοποιήσαμε τον αλγόριθμο του Porter Stemmer, που αποτελεί την πιο διαδεδομένη τεχνική που εφαρμόζεται στο stemming εγγράφων:

Ο αλγόριθμος του Porter Stemmer ακολουθεί τη μέθοδο affix removal, η οποία περιλαμβάνει την αφαίρεση του προθέματος (prefix) και του αποθέματος-κατάληξης (suffix) κάθε λέξης ώστε να απομένει μόνο η ρίζα της. Βασίζεται στην ιδέα ότι οι καταλήξεις στην αγγλική γλώσσα (περίπου 1200) δημιουργούνται από συνδυασμούς μικρότερων και απλούστερων καταλήξεων. Θα περιγράψουμε εδώ τον αρχικό αλγόριθμο που γράφτηκε από τον Martin Porter το 1976.

Ξεκινάμε με την απόδοση μερικών ορισμών οι οποίοι θα βοηθήσουν στην κατανόηση του αλγορίθμου:

Consonant (σύμφωνο) σε μια λέξη είναι ένα γράμμα διάφορο των A, E, I, O ή U, και διάφορο του Y ακολουθούμενο από κάποιο σύμφωνο. Οπότε στη λέξη TOY τα σύμφωνα είναι τα T και Y ενώ στη λέξη SYZYGY τα σύμφωνα είναι τα S, Z και G.

Ένα γράμμα που δεν είναι σύμφωνο είναι vowel (φωνήεν).

Ένα σύμφωνο θα συμβολίζεται με το γράμμα c ενώ ένα φωνήεν με το v. Μια λίστα ccc… μήκους μεγαλύτερου του 0 θα συμβολίζεται με C, και μια λίστα vvv… με μήκος μεγαλύτερο του 0 θα συμβολίζεται με V. Συνεπώς, κάθε λέξη, ή τμήμα μιας λέξης, θα έχει μια από τις ακόλουθες μορφές:

CVCV … C

CVCV … V

VCVC … C

VCVC … V

Όλες οι παραπάνω μορφές μπορούν να αναπαρασταθούν από μία μόνο μορφή:

[C] VCVC … [V]

Όπου οι αγκύλες συμβολίζουν αυθαίρετη παρουσία των περιεχομένων τους. Χρησιμοποιώντας το συμβολισμό (VC){m} για να συμβολίσουμε ότι το VC επαναλαμβάνεται m φορές, η παραπάνω μορφή μπορεί να γραφεί ξανά ως:

[C] (VC) {m} [V].

Το m θα ονομάζεται μέτρο (measure) οποιασδήποτε λέξης ή τμήματος λέξης που αναπαρίσταται με αυτή τη μορφή. Η περίπτωση που m = 0 καλύπτει την κενή (null) λέξη. Ας δούμε μερικά παραδείγματα:

m=0 TR, EE, TREE, Y, BY.

m=1 TROUBLE, OATS, TREES, IVY.

m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

Οι κανόνες για την αφαίρεση μιας κατάληξης θα δίνονται στη μορφή:

(condition) $S_1 \rightarrow S_2$

Αυτό σημαίνει ότι εάν μια λέξη καταλήγει με την κατάληξη S_1 , και η ρίζα (stem) πριν την S_1 ικανοποιεί τη δοσμένη συνθήκη, τότε η κατάληξη S_1 αντικαθίσταται από την S_2. Η συνθήκη συνήθως ορίζεται ως προς το m, για παράδειγμα

(m > 1) EMENT →

όπου η S1 είναι `EMENT' και η S2 είναι null. Οπότε εδώ θα είχαμε την αντιστοίχιση του REPLACEMENT με το REPLAC, εφόσον το REPLAC είναι τμήμα λέξης για το οποίο ισχύει m = 2.

Το μέρος της συνθήκης μπορεί ακόμη να περιέχει τα ακόλουθα:

*S - η ρίζα τελειώνει με S (και παρόμοια για άλλα γράμματα)

v - η ρίζα περιέχει ένα φωνήεν

*d - η ρίζα τελειώνει με διπλό σύμφωνο (πχ -TT, -SS)

*o - η ρίζα τελειώνει με cvc, όπου το c δεν είναι W, X ή Y (πχ -WIL, -HOP).

Επιπλέον, το μέρος της συνθήκης μπορεί να περιέχει λογικές εκφράσεις, όπως and, or, not, για παράδειγμα η συνθήκη

(m>1 and (*S or *T))

ελέγχει αν η ρίζα έχει μέτρο m>1 και τελειώνει με S ή T, ενώ η συνθήκη

(*d and not (*L or *S or *Z))

ελέγχει αν η ρίζα τελειώνει με ένα διπλό σύμφωνο διάφορο των L, S ή Z.

Ωστόσο περίπλοκες συνθήκες σαν αυτή απαιτούνται πολύ σπάνια.

Σε ένα σύνολο κανόνων γραμμένοι ο ένας κάτω από τον άλλο, μόνο ένας πρέπει να τηρείται, και θα είναι αυτός με τη μεγαλύτερη αντιστοιχία κατάληξης S_1 της δοσμένης λέξης.

Για παράδειγμα, αν έχουμε τους κανόνες

SSES → SS

IES → I

SS → SS

S →

(με κενές όλες τις συνθήκες), τότε η λέξη CARESSES θα αντιστοιχείται στη ρίζα CARESS εφόσον η SSES είναι η μεγαλύτερη κατάληξη που αντιστοιχείται. Ομοίως, η λέξη CARESS αντιστοιχείται στη ρίζα CARESS (S_1=`SS') και η λέξη CARES στη ρίζα CARE ($S_1$=`S').

Στους κανόνες που δίνονται παρακάτω δίνονται μαζί στα δεξιά και παραδείγματα της εφαρμογής τους, με μικρά γράμματα. Τα παραδείγματα που βρίσκονται κάτω από τους κανόνες αποτελούν εξαιρέσεις, στις οποίες δεν εφαρμόζεται ο εκάστοτε κανόνας. Ακολουθεί τώρα ο αλγόριθμος:

Βήμα 1a

SSES → SS	caresses → caress
IES → I	ponies → poni
	ties → ti
SS → SS	caress → caress
S →	cats → cat

Βήμα 1b

(m>0) EED → EE	feed → feed
	agreed → agree
(*v*) ED →	plastered → plaster
	bled → bled
(*v*) ING →	motoring → motor
	sing → sing

Αν ο δεύτερος ή ο τρίτος από τους κανόνες στο βήμα 1b τηρείται τότε έχουμε τα ακόλουθα:

AT → ATE	conflat(ed) → conflate
BL → BLE	troubl(ed) → trouble
IZ → IZE	siz(ed) → size
(*d and not (*L or *S or *Z)) → single letter	
	hopp(ing) → hop

tann(ed) → tan
fall(ing) → fall
hiss(ing) → hiss
fizz(ed) → fizz

(m=1 and *o) → E fail(ing) → fail
fil(ing) → file

Ο κανόνας αντιστοίχισης σε ένα μόνο γράμμα προκαλεί την αφαίρεση ενός από το ζευγάρι των διπλών γραμμάτων. Το -E is προστίθεται στα -AT, -BL and -IZ, έτσι ώστε να μπορούν να αναγνωριστούν αργότερα οι καταλήξεις -ATE, -BLE και -IZE. Αυτό το E θα αφαιρεθεί στο βήμα 4.

Βήμα 1c

(*v*) Y → I happy → happi
sky → sky

Το πρώτο βήμα ασχολείτο κυρίως με πληθυντικούς αριθμούς και μετοχές (past participles). Τα ακόλουθα βήματα είναι πιο ξεκάθαρα:

Βήμα 2

(m>0) ATIONAL → ATE relational → relate
(m>0) TIONAL → TION conditional → condition
rational → rational
(m>0) ENCI → ENCE valenci → valence
(m>0) ANCI → ANCE hesitanci → hesitance
(m>0) IZER → IZE digitizer → digitize
(m>0) ABLI → ABLE conformabli → conformable
(m>0) ALLI → AL radicalli → radical
(m>0) ENTLI → ENT differentli → different
(m>0) ELI → E vileli - > vile
(m>0) OUSLI → OUS analogousli → analogous
(m>0) IZATION → IZE vietnamization → vietnamize
(m>0) ATION → ATE predication → predicate
(m>0) ATOR → ATE operator → operate
(m>0) ALISM → AL feudalism → feudal
(m>0) IVENESS → IVE decisiveness → decisive
(m>0) FULNESS → FUL hopefulness → hopeful

(m>0) OUSNESS → OUS callousness → callous
(m>0) ALITI → AL formaliti → formal
(m>0) IVITI → IVE sensitiviti → sensitive
(m>0) BILITI → BLE sensibiliti → sensible

Ο έλεγχος για το string S_1 μπορεί να γίνει γρήγορα κάνοντας μια προγραμματισμένη αλλαγή στο προτελευταίο γράμμα της λέξης που ελέγχεται. Αυτό δίνει μια άρτια ταξινόμηση των πιθανών τιμών για το string S_1. Μάλιστα, θα δούμε ότι τα S_1-strings του βήματος 2 παρουσιάζονται εδώ σε αλφαβητική σειρά του προτελευταίου γράμματός τους. Παρόμοιες τεχνικές μπορούν να εφαρμοστούν και σε επόμενα βήματα.

Βήμα 3

(m>0) ICATE → IC triplicate → triplic
(m>0) ATIVE → formative → form
(m>0) ALIZE → AL formalize → formal
(m>0) ICITI → IC electriciti → electric
(m>0) ICAL → IC electrical → electric
(m>0) FUL → hopeful → hope
(m>0) NESS → goodness → good

Βήμα 4

(m>1) AL → revival → reviv
(m>1) ANCE → allowance → allow
(m>1) ENCE → inference → infer
(m>1) ER → airliner → airlin
(m>1) IC → gyroscopic → gyroscop
(m>1) ABLE → adjustable → adjust
(m>1) IBLE → defensible → defens
(m>1) ANT → irritant → irrit
(m>1) EMENT → replacement → replac
(m>1) MENT → adjustment → adjust
(m>1) ENT → dependent → depend
(m>1 and (*S or *T)) ION → adoption → adopt
(m>1) OU → homologou → homolog
(m>1) ISM → communism → commun
(m>1) ATE → activate → activ

(m>1) ITI → angulariti → angular
(m>1) OUS → homologous → homolog
(m>1) IVE → effective → effect
(m>1) IZE → bowdlerize → bowdler

Τώρα αφαιρούνται οι καταλήξεις.

Βήμα 5a

(m>1) E → probate → probat
rate → rate
(m=1 and not *o) E → cease → ceas

Βήμα 5b
(m > 1 and *d and *L) → single letter controll → control
roll → roll

Ο αλγόριθμος φροντίζει να μην αφαιρεί μία κατάληξη όταν η ρίζα είναι πολύ μικρή, ενώ το μήκος της ρίζας δίνεται από το μέτρο της m. Δεν υπάρχει κάποια γλωσσολογική βάση γι' αυτή την προσέγγιση, απλά παρατηρήθηκε ότι το m μπορούσε να χρησιμοποιηθεί αρκετά αποτελεσματικά για να αποφασιστεί εάν είναι κατάλληλη ή όχι η αφαίρεση της κατάληξης.
Για παράδειγμα, στις ακόλουθες λίστες:

Λίστα A	Λίστα B
RELATE	DERIVATE
PROBATE	ACTIVATE
CONFLATE	DEMONSTRATE
PIRATE	NECESSITATE
PRELATE	RENOVATE

Η κατάληξη -ATE αφαιρείται από τις λέξεις της λίστας B, όχι όμως από τις λέξεις της λίστας A. Αυτό σημαίνει ότι τα ζευγάρια DERIVATE/DERIVE, ACTIVATE/ACTIVE, DEMONSTRATE/DEMONSTRABLE, NECESSITATE/NECESSITOUS, θα συγχωνεύονται μαζί. Το γεγονός ότι δεν γίνεται καμία προσπάθεια να αναγνωριστούν προθέματα μπορεί να κάνει τα αποτελέσματα να μοιάζουν ασυνεπή. Έτσι η λέξη PRELATE δε χάνει το -ATE, αλλά η λέξη ARCHPRELATE γίνεται ARCHPREL. Στην πράξη αυτό δεν έχει πολλή σημασία, επειδή η παρουσία του προθέματος μειώνει την πιθανότητα κάποιας λανθασμένης συγχώνευσης.

Οι σύνθετες καταλήξεις αφαιρούνται κομμάτι-κομμάτι σε διαφορετικά βήματα. Έτσι η λέξη GENERALIZATIONS στο βήμα 1 γίνεται GENERALIZATION, στο βήμα 2 GENERALIZE, στο βήμα 3 GENERAL, και στο βήμα 4 GENER. Η λέξη OSCILLATORS στο βήμα 1 γίνεται OSCILLATOR, στο βήμα 2 OSCILLATE, στο βήμα 4 OSCILL, και τέλος στο βήμα 5 OSCIL.

Στο σύστημα, ο αλγόριθμος βρίσκεται στην κλάση Stemmer η οποία καλείται για κάθε λέξη που είναι υπό επεξεργασία στην κλάση StringToWordVector.

4.1.9 Ο πίνακας όρων-εγγραφών

Για κάθε λέξη που έχει υποστεί επεξεργασία και έχει βρεθεί το stem της (ή το stem κάποιας συνώνυμης λέξης της εάν είχε συγχωνευθεί ήδη με κάποια λέξη), ανανεώνεται ο πίνακας όρων-εγγράφων (dictionaryArr στην κλάση StringToWordVector) ο οποίος μετράει για κάθε όρο σε κάθε διάνυσμα πόσες φορές εμφανίζεται. Σημειώνουμε ότι εφόσον έχουμε περάσει τη διαδικασία του stemming, οι όροι θα είναι τα stems των λέξεων. Με αυτόν τον τρόπο επιτυγχάνουμε επιπλέον συγχώνευση των λέξεων που είχαν ίδια ρίζα αλλά διαφορετική μορφή, και άρα τείνουν να έχουν όμοιο ή παρόμοιο νόημα.

4.1.10 Pruning

Αφού διαβαστούν τα κείμενα όλων των instances και υποστούν την επεξεργασία όλες οι λέξεις τους, ο πίνακας όρων-εγγράφων έχει συμπληρωθεί στο ακέραιο. Τότε είμαστε σε θέση να προχωρήσουμε σε αφαίρεση όρων που είναι πολύ σπάνιοι σε κάθε αρχείο κειμένου. Η διαδικασία αυτή ονομάζεται pruning.
Συγκεκριμένα, για κάθε αρχείο, ανάλογα με τον αριθμό των attributes που περιέχει, αρχικά επιλέγεται το κατώτατο όριο συχνότητας εμφάνισης που μπορεί να έχει ένας όρος στο έγγραφο. Η επιλογή αυτή γίνεται υπολογίζοντας τον αρχικό αριθμό των λέξεων από τις οποίες αποτελείται το κείμενο. Εάν ο αριθμός είναι μικρότερος από τον αριθμό που επιλέγει ο χρήστης να επεξεργάζεται το σύστημα (mWordsToKeep) τότε επιλέγεται ως κατώτατο όριο μια προεπιλεγμένη κατώτατη συχνότητα, ειδάλλως το κατώτατο όριο επιλέγεται να είναι τουλάχιστον όσο η προεπιλεγμένη κατώτατη συχνότητα. Στη συνέχεια, για κάθε instance ελέγχεται αν η συχνότητα κάθε όρου είναι μεγαλύτερη από το κατώτατο όριο που είχε τεθεί, ώστε να προστεθεί στον τελικό πίνακα όρων-εγγράφων, ειδάλλως αγνοείται.

4.1.11 Στάθμιση (term weighting)

Αφού υπολογιστεί ο τελικός πίνακας όρων εγγράφων, για κάθε attribute (δηλαδή όρο) κάθε instance (δηλαδή έγγραφο) υπολογίζεται το βάρος που αντιστοιχεί.

Όπως προαναφέρθηκε, κατά την κλήση της κλάσης StringToWordVector υπάρχει η δυνατότητα να προσθέσει ο χρήστης και παραμέτρους σχετικές με τι είδους βάρη επιθυμεί να ανατίθενται στους όρους:

Εάν δεν τοποθετήσει καμία παράμετρο, χρησιμοποιείται το Boolean model με βάσει το οποίο για κάθε όρο κάθε εγγράφου θα έχουμε την τιμή 1 ή 0 ανάλογα με την εμφάνιση ή όχι του όρου από το συγκεκριμένο αρχείο.

Με την παράμετρο -C, το βάρος που αποδίδεται σε κάθε όρο είναι τα word counts, πόσες φορές δηλαδή εμφανίζεται ο όρος στο κείμενο.

Με την παράμετρο -T, θεωρείται ως βάρος το TF (term frequency) με τον υπολογισμό $TF(d,t) = 1 + \log n(d,t)$, όπου $n(d,t)$ είναι το word count του όρου t στο αρχείο d.

Με την παράμετρο -I, θεωρείται ως βάρος το TF-IDF (term frequency – inverse document frequency), με τον υπολογισμό $w(d,t) = TF(d,t) * IDF(t)$,

όπου

$IDF(t) = \log\left(\frac{D}{DF(t)}\right)$ με

DF(t) – Document frequency: εκφράζει πόσα κείμενα από τη συλλογή που έχουμε περιέχουν τον όρο t και

D: ο αριθμός των αρχείων που συγκροτούν τη συλλογή κειμένων που έχουμε (άρα και ο αριθμός των διανυσμάτων)

Τέλος, με την παράμετρο –N και "0=not normalize/1=normalize all data/2=normalize test data only" μπορούμε να επιλέξουμε εάν θέλουμε να κανονικοποιήσουμε τα βάρη που υπολογίζουμε με ως προς το μέσο μήκος των εγγράφων. Η κανονικοποίηση γίνεται με πολλαπλασιασμό κάθε τιμής του βάρους επί το συντελεστή m_AvgDocLength / docLength, δηλαδή το μέσο όρο μήκους των αρχείων κειμένου διά το μήκος του εκάστοτε αρχείου κειμένου για το οποίο υπολογίζονται τα βάρη των όρων του.

Όπως θα αναλύσουμε και στο επόμενο κεφάλαιο, αποφασίσαμε να κάνουμε διάφορες δοκιμές για να δούμε πώς επιτυγχάνονταν μετέπειτα αποτελεσματικότερη ομαδοποίηση. Έτσι, δοκιμάσαμε όλα τα βάρη που μόλις προαναφέραμε δίνοντας τις αντίστοιχες παραμέτρους τους, ενώ επιπλέον υπολογίσαμε και το TF-IDF χρησιμοποιώντας ως τύπους:

$$w(d,t) = TF(d,t) * IDF(t) \qquad TF(d,t) = n(d,t) \qquad IDF(t) = \log\left(\frac{D - DF(t)}{DF(t)}\right)$$

δηλαδή έχοντας ως term frequency τα word counts.

Ύστερα από διάφορε δοκιμές, καταλήξαμε να προτείνουμε τον υπολογισμό βάρους TF-IDF σε συνδυασμό με κανονικοποίηση ως προς το μέσο μήκος των αρχείων κειμένου, όπως

υπολογίζονται αν δώσουμε τις παραμέτρους –I για το TF-IDF και –N 1 για την κανονικοποίηση ως προς το μήκος στην κλήση της κλάσης StringToWordVector.

4.1.12 Ο πίνακας όρων-εγγράφων

Η έξοδος της κλάσης StringToWordVector είναι ο πίνακας όρων-εγγράφων, αποθηκευμένος σε ένα αρχείο της μορφής arff. Συγκεκριμένα, στο αρχείο αυτό περιέχονται instances, καθένα από τα οποία αντιστοιχεί και σε ένα αρχείο κειμένου, και το οποίο περιέχει ένα string attribute που αντιστοιχεί στο όνομα του αρχείου και από numeric (δηλαδή αριθμητικά, ποσοτικά) attributes καθένα από τα οποία αντιστοιχεί και σε ένα όρο και έχει ως τιμή το βάρος του όρου για το συγκεκριμένο έγγραφο. Έτσι, μπορούμε να φανταστούμε την αναπαράσταση του πίνακα όρων εγγράφων ως:

Έγγραφα/Instances	Όροι/Attributes					
	(instance)	attribute 0 (string attribute)	attribute 1 ($1^{ος}$ όρος)	attribute 2 ($2^{ος}$ όρος)	……	attribute m ($m^{ος}$ όρος)
	(instance 0)	“name_0.txt”	weight_0_1	weight_0_2		weight_0_m
	(instance 1)	“name_2.txt”	weight_1_1	weight_1_2		weight_1_m
	….	….				
	(instance n)	“name_n.txt”	weight_n_1	weight_n_2		weight_n_m

4.1.13 Ομαδοποίηση με τον αλγόριθμο k-means

Το αρχείο arff που παράγεται από την κλάση StringToWordVector και περιέχει τον πίνακα όρων-εγγράφων, μπορεί πλέον να χρησιμοποιηθεί για τη διαδικασία της ομαδοποίησης. Συγκεκριμένα, χρησιμοποιούμε το αρχείο αυτό ως αρχείο εισόδου της κλάσης AddCluster η οποία, όπως προαναφέραμε και για τη StringToWordVector, επεξεργάζεται τα χαρακτηριστικά με τη μέθοδο unsupervised filter. Η λειτουργία της AddCluster είναι να καλεί κάποια κλάση ομαδοποίησης που ομαδοποιεί τα instances που περιέχονται στο αρχείο εισόδου (δηλαδή τα έγγραφα στην περίπτωσή μας), και να παράγει ως έξοδο ένα νέο arff αρχείο στο οποίο κάθε instance έχει ένα επιπλέον attribute, μορφής nominal, το οποίο αντιπροσωπεύει την ομάδα (cluster) στην οποία κατατάσσεται το αντίστοιχο instance κατά την ομαδοποίηση. Η κλήση της AddCluster συνοδεύεται από τις παραμέτρους –W <cluster specification> – όπου cluster specification είναι το όνομα της κλάσης που θα κληθεί για να τελέσει την ομαδοποίηση από την AddCluster μαζί με τυχόν παραμέτρους που παίρνει η

κλάση ομαδοποίησης, και –I <att1,att2-att4,…> το εύρος των attributes που θα πρέπει να αγνοήσει ο clusterer (δηλαδή η μέθοδος ομαδοποίησης).

Ως κλάση ομαδοποίησης επιλέξαμε να χρησιμοποιούμε την SimpleKMeans, η οποία ομαδοποιεί τα δεδομένα εκτελώντας τον αλγόριθμο k-means. Στην κλήση της κύριας μεθόδου της κλάσης SimpleKMeans δίνονται οι παράμετροι –N <num> για τον αριθμό των ομάδων που θέλουμε να έχουμε μετά την ομαδοποίηση και -S <num> ένας τυχαίος αριθμός seed που χρησιμοποιείται για την εκτέλεση του αλγόριθμου k-means. Η προεπιλογή εάν δεν δοθούν παράμετροι είναι N=2 ομάδες και S=10 ο τυχαίος αριθμός.

Το μέτρο ομοιότητας που χρησιμοποιείται για την ομαδοποίηση με τη μέθοδο SimpleKMeans στο Weka είναι η Ευκλείδεια απόσταση των instances (αρχείων κειμένου) εφόσον χρησιμοποιούμε όπως έχουμε ήδη πει το μοντέλο του διανυσματικού χώρου και κάθε instance αποτελεί ένα διάνυσμα, όπως περιγράφηκε στο προηγούμενο κεφάλαιο:

$$SIM(X_j, X_k) = \sqrt{\sum_{i=1}^{n} \left(x_{ij} - x_{ik}\right)^2}$$

Όσο μικρότερη είναι η απόσταση μεταξύ των διανυσμάτων (όσο πιο κοντά βρίσκονται δηλαδή μέσα στο διανυσματικό χώρο) τόσο πιο όμοια θεωρούνται τα αντίστοιχα έγγραφα μεταξύ τους.

Ωστόσο, καθώς δοκιμάζαμε διάφορα βάρη για τους όρους, επιλέξαμε να δοκιμάσουμε (όπως θα δούμε και στις δοκιμές στο επόμενο κεφάλαιο) ως μέτρο ομοιότητας και το cosine similarity:

$$SIM(X_j, X_k) = \frac{\sum_{i=1}^{n} \left(x_{ij} \times x_{ik}\right)}{\sqrt{\sum_{i=1}^{n} \left(x_{ij}\right)^2} \times \sqrt{\sum_{i=1}^{n} \left(x_{jk}\right)^2}}$$

Όσο πιο κοντά στην τιμή 1 είναι το cosine distance τόσο πιο κοντά είναι μεταξύ τους τα term vectors και τόσο ομοιότερα μεταξύ τους τα αντίστοιχα έγγραφα.

Το cosine similarity προτείνεται συχνά ως μέτρο ομοιότητας γιατί επιτυγχάνει κανονικοποίηση ως προς το μήκος των αρχείων κειμένου, επομένως είναι προτιμότερο όταν χρησιμοποιούμε ως βάρος το TF ή το TF-IDF.

Ωστόσο, επειδή καταλήξαμε να χρησιμοποιούμε ως βάρος το TF-IDF μαζί με το συντελεστή κανονικοποίησης ως προς το μέσο μήκος των αρχείων, τελικά χρησιμοποιήσαμε ως μέτρο ομοιότητας την Ευκλείδεια απόσταση.

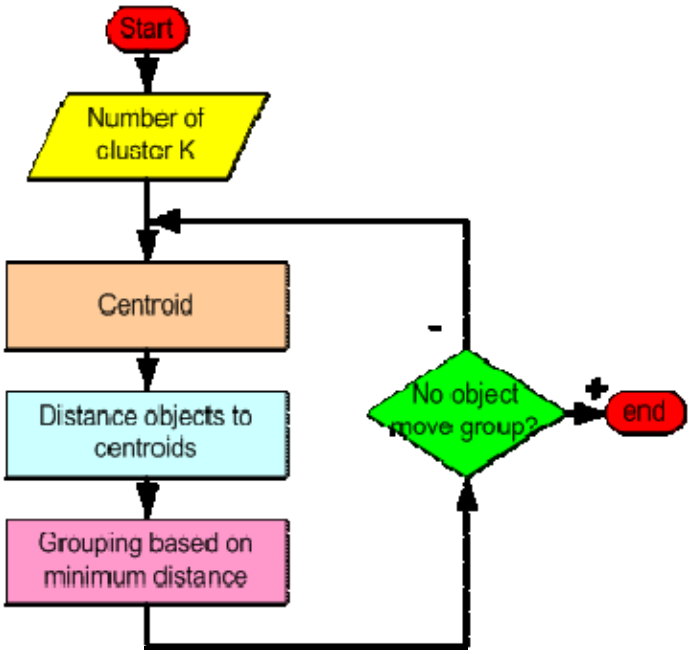
Ας δούμε τώρα πιο αναλυτικά τον αλγόριθμο K-means.

Επιλέξαμε ως μέθοδο ομαδοποίησης τον αλγόριθμο K-means λόγω της ευρείας χρήσης του, καθώς θεωρείται από τους πιο αποδοτικούς αλγόριθμους ομαδοποίησης. Ο αλγόριθμος K-

means ανήκει στους διαχωριστικούς αλγόριθμους, τους οποίους συναντήσαμε στο προηγούμενο κεφάλαιο. Θα μελετήσουμε τώρα τον τρόπο λειτουργίας αυτού του αλγορίθμου.

Ο αλγόριθμος K-means

Κατά την έναρξη του K-means θέτουμε τον αριθμό K των clusters που θέλουμε να έχουμε και υποθέτουμε ότι έχουμε ένα τυχαίο διαχωρισμό των στοιχείων (διανύσματα στο μοντέλο διανυσματικού χώρου) προς ομαδοποίηση σε K ομάδες, κάνοντας μια τυχαία επιλογή –που εξαρτάται από τον τυχαίο αριθμό S (seed) που δίνουμε, K στοιχείων ως τα κέντρα (centroids) των clusters. Τότε, ο K-means αλγόριθμος θα επαναλαμβάνει τον υπολογισμό των κέντρων και των αποστάσεων κάθε στοιχείου από κάθε κέντρο και το διαχωρισμό σε clusters με κριτήριο την ελάχιστη απόσταση στοιχείου από κάποιο κέντρο, έως ότου συγκλίνει, δηλαδή σταματήσει να έχει μετακινήσεις των στοιχείων από ένα cluster σε άλλο.



Εικόνα 4-1 : Διάγραμμα ροής του αλγόριθμου K-means

Βήμα 1: Ξεκινάμε με προσδιορισμό της τιμής k = αριθμός των clusters

Βήμα 2: Κάνουμε έναν αρχικό διαχωρισμό που ομαδοποιεί τα δεδομένα σε k ομάδες. Μπορούμε να χωρίσουμε τα instances τυχαία, ή συστηματικά ως ακολούθως:

1. Παίρνουμε τα k διαδοχικά στοιχεία ως μοναδικά στοιχεία ενός cluster
2. Αντιστοιχίζουμε τα υπόλοιπα (N-k) στοιχεία στη ομάδα με το κοντινότερο (με βάση κάποιο μέτρο απόστασης) κέντρο. Μετά την αντιστοίχιση, επαναϋπολογίζουμε το κέντρο κάθε cluster.

Βήμα 3: Παίρνουμε κάθε στοιχείο διαδοχικά και υπολογίζουμε την απόστασή του (με κάποιο μέτρο απόστασης, πχ Ευκλείδεια απόσταση) από το κέντρο κάθε cluster. Αν

ένα στοιχείο δεν βρίσκεται στο cluster με το κοντινότερο κέντρο σε αυτό, μεταφέρουμε το στοιχείο σε αυτό το cluster και επαναϋπολογίζουμε το κέντρο του cluster από το οποίο το αφαιρέσαμε και του cluster στο οποίο το προσθέσαμε.

Βήμα 4: Επαναλαμβάνουμε το βήμα 3 μέχρι να επιτύχουμε σύγκλιση, δηλαδή μέχρι να βρούμε κάποιο πέρασμα όλων των στοιχείων χωρίς να έχουμε νέα αντιστοίχιση (αλλαγή) σε cluster.

Αν ο αριθμός των στοιχείων που έχουμε είναι μικρότερος του αριθμού των cluster τότε θεωρούμε κάθε στοιχείο ως κέντρο μιας ομάδας. Κάθε κέντρο θα έχει και έναν αριθμό ενός cluster. Αν ο αριθμός των στοιχείων είναι μεγαλύτερος από τον αριθμό των cluster, για κάθε στοιχείο υπολογίζουμε την απόσταση από όλα τα κέντρα και παίρνουμε την μικρότερη απόσταση. Τότε αυτό το στοιχείο θεωρείται ότι ανήκει στην ομάδα που έχει τη μικρότερη απόσταση από το στοιχείο αυτό.

Αφού δεν είμαστε σίγουρη για τη θέση κάθε κέντρου, χρειάζεται να προσαρμόζουμε τις συντεταγμένες κάθε κέντρου στα πρόσφατα ανανεωμένα δεδομένα. Έπειτα αντιστοιχίζουμε κάθε στοιχείο με αυτό το νέο κέντρο. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να μην έχουμε καμία κίνηση στοιχείου σε άλλο cluster πια. Μαθηματικά, αυτός ο βρόχος έχει αποδειχθεί ότι συγκλίνει. Η σύγκλιση θα συμβαίνει πάντα όταν ικανοποιείται η ακόλουθη συνθήκη:

1. Για κάθε αλλαγή στο βήμα 2, το άθροισμα των αποστάσεων κάθε στοιχείου από το κέντρο της ομάδας στην οποία ανήκει το στοιχείο μειώνεται.
2. Ο αριθμός των διαχωρίσεων που υπάρχουν για να ομαδοποιηθούν τα στοιχεία σε k clusters είναι πεπερασμένος.

Όπως και άλλοι αλγόριθμοι, ωστόσο, η ομαδοποίηση με K-means μπορεί να εμπεριέχει ορισμένες αδυναμίες:

- Όταν ο αριθμός των στοιχείων δεν είναι μεγάλος, η αρχική τυχαία ομαδοποίηση θα επηρεάσει σημαντικά τη διαμόρφωση των ομάδων.
- Ο αριθμός των cluster, K πρέπει να έχει οριστεί πριν την εκκίνηση του αλγορίθμου.
- Αν ο αλγόριθμος συγκλίνει σε κάποιο τοπικό ελάχιστο, με αποτέλεσμα κάποιες φορές να μην προτείνει την πιο ικανοποιητική λύση.
- Ο k-means μπορεί να επεξεργαστεί μόνο αριθμητικά δεδομένα.
- Εξαιτίας του χρόνου που χρειάζεται για να ολοκληρωθεί μια επανάληψη δεν μπορεί να χειριστεί μεγάλες βάσεις δεδομένων γρήγορα.

4.1.14 Εξαγωγή αποτελεσμάτων

Κατά την εκτέλεση των μεθόδων της κλάσης SimpleKMeans παράγεται ένα txt αρχείο, το οποίο περιέχει αναλυτικά σε πιο cluster ομαδοποιείται κάθε instance (κάθε αρχείο κειμένου) καθώς και στατιστικά στοιχεία για την ομαδοποίηση (πχ τι ποσοστό αρχείων περιέχεται σε κάθε cluster). Το αρχείο αυτό μπορεί να διαβαστεί από το χρήστη καθώς επίσης και να προσπελαστεί από οποιαδήποτε εφαρμογή που συνδέεται με το σύστημα της ομαδοποίησης.

Επιπλέον, τα αποτελέσματα εξάγονται και μέσω του arff αρχείου που είπαμε ότι παράγεται στην κλάση AddCluster, η οποία προσθέτει στα instances του αρχείου που δέχεται ως είσοδο ένα επιπλέον attribute που αναφέρει σε ποιο cluster έχει ομαδοποιηθεί κάθε instance. Το αρχείο αυτό μπορεί φυσικά να προσπελαστεί από το Weka, έτσι ώστε εάν θέλουμε να επιτύχουμε και απεικόνιση - οπτικοποίηση (visualization) των αποτελεσμάτων μέσω της λειτουργίας Visualize του Weka.
Έτσι, με την εξαγωγή των αποτελεσμάτων, ολοκληρώνεται η διαδικασία της ομαδοποίησης του συστήματός μας.

4.2 Οι λειτουργικές προδιαγραφές του IDeM ως προς την ομαδοποίηση ιδεών

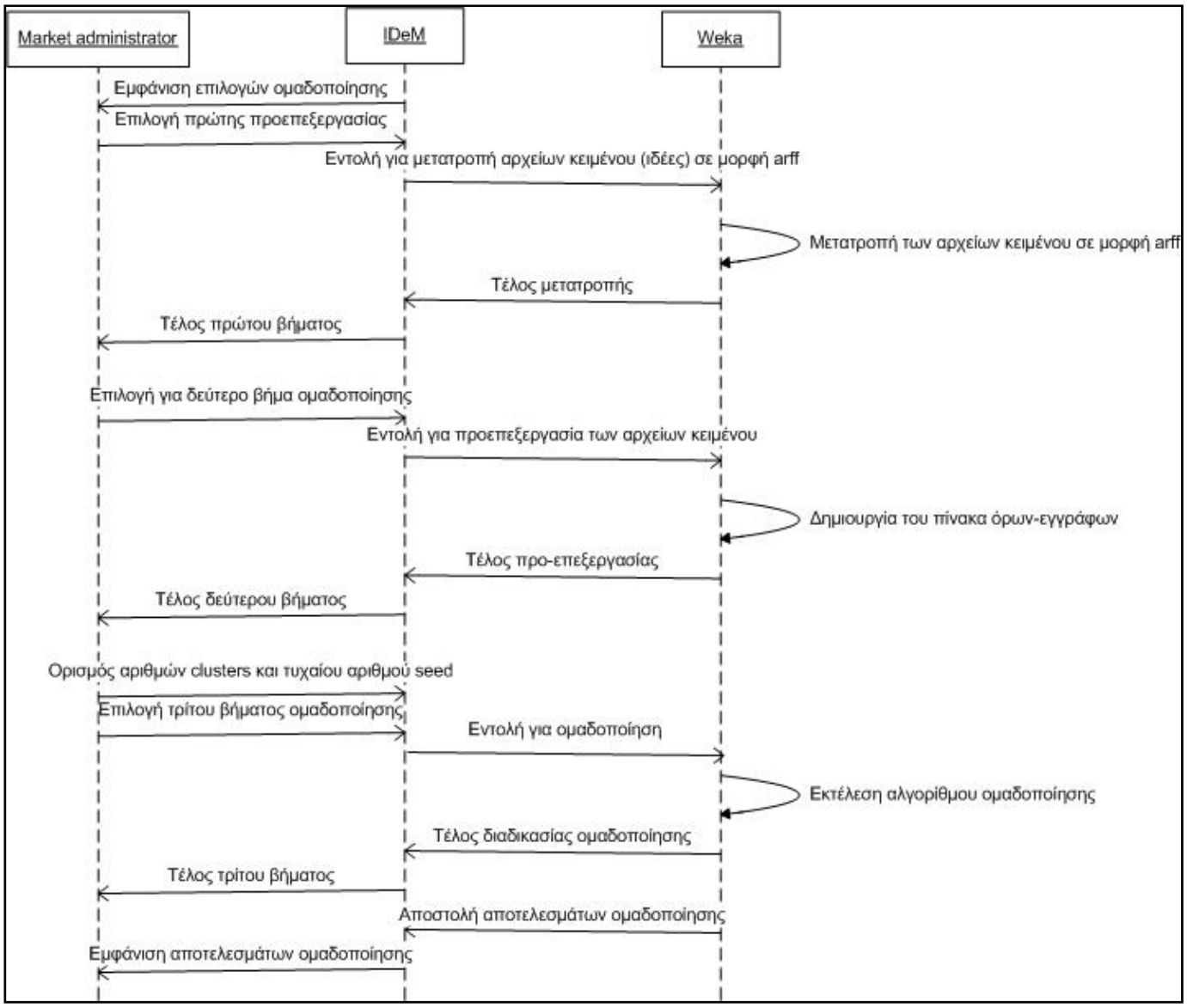
Επιστρέφουμε τώρα στο σκοπό για τον οποίο αποφασίσαμε να αναπτύξουμε ένα σύστημα ομαδοποίησης κειμένων: να ομαδοποιήσουμε τις ιδέες στο IDeM για την εξυπηρέτηση του administrator, ο οποίος έχει να κάνει με μεγάλο αριθμό ιδεών από τις οποίες πρέπει να επιλέξει ποιες θα βάλει στην αγορά. Όπως αναφέραμε στην ενότητα 2.3, οι ιδέες υποβάλλονται από τους χρήστες σε μορφή κειμένου. Συνεπώς, μια ομαδοποίηση των κειμένων που περιέχουν τις ιδέες ως προς το θεματικό τους περιεχόμενο θα εξυπηρετούσε τις ανάγκες του IDeM. Στην ενότητα αυτή θα ασχοληθούμε με τις λειτουργικές προδιαγραφές του IDeM για την ομαδοποίηση ιδεών.

Παραθέτουμε εδώ το γενικό διάγραμμα χρήσης του συστήματος, στο οποίο συμπεριλαμβάνεται και η λειτουργική απαίτηση για ομαδοποίηση των ιδεών:

Εικόνα 4-2 : Διάγραμμα Περίπτωσης Χρήσης του IDeM

Ας δούμε τώρα την περίπτωση χρήσης της ομαδοποίησης ιδεών:

Ο Market Administrator θα έχει τη δυνατότητα να ομαδοποιεί τις ιδέες που υπάρχουν στο σύστημα. Έτσι, χρειάζεται να έχει πρόσβαση σε κάποια σχετική σελίδα στην οποία θα μπορεί να ακολουθεί τα βήματα της (ημιαυτόματης) ομαδοποίησης των ιδεών. Η διαδικασία αυτή φαίνεται στο παρακάτω ακολουθιακό διάγραμμα.



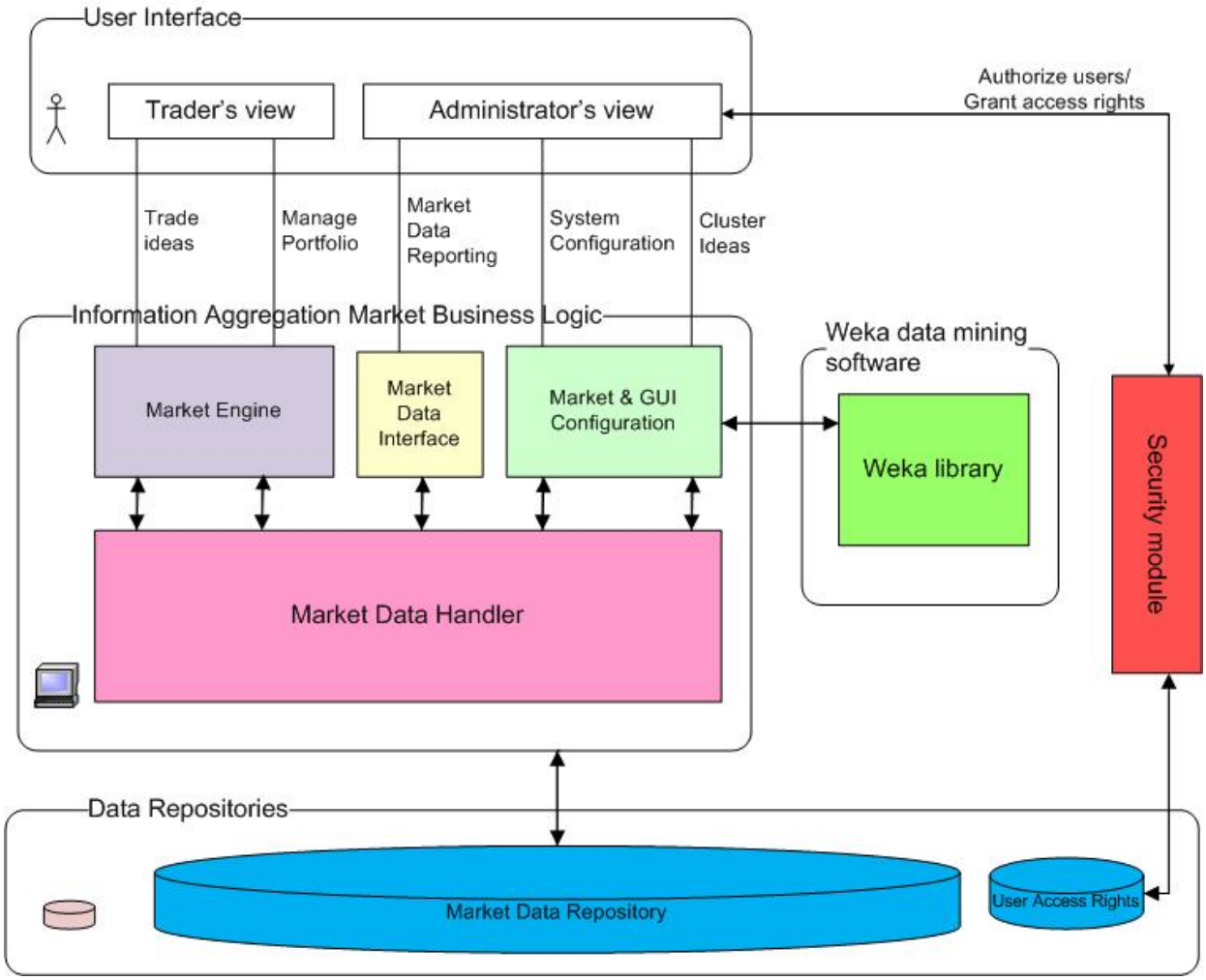
Εικόνα 4-3 : Ακολουθιακό διάγραμμα της διαδικασίας ομαδοποίησης ιδεών

Παρατηρούμε λοιπόν ότι το σύστημα IDeM θα πρέπει να συνδέεται με το σύστημα του Weka. Ο χρήστης (Market Administrator), θα πρέπει μέσα από το γραφικό περιβάλλον του IDeM να έχει πρόσβαση σε κάποια σελίδα στην οποία να μπορεί να ομαδοποιεί τις ιδέες σε όσες ομάδες επιθυμεί και να βλέπει το αποτέλεσμα της ομαδοποίησης. Ο χρήστης φυσικά θα έχει επαφή με το γραφικό περιβάλλον του IDeM χωρίς να γνωρίζει τη διαδικασία της ομαδοποίησης που γίνεται στο παρασκήνιο. Στην επόμενη ενότητα θα δούμε την αρχιτεκτονική του IDeM και τη διασύνδεσή του με το Weka για την ενσωμάτωση της διαδικασίας της ομαδοποίησης.

4.3 Αρχιτεκτονική συστήματος

Στο 1^{ο} κεφάλαιο αναφερθήκαμε συνοπτικά στην αρχιτεκτονική του IDeM. Όπως είδαμε, διακρίνουμε μια αρχιτεκτονική τριών στρωμάτων, η οποία επιτρέπει να χρησιμοποιούνται διαφορετικά εργαλεία για την ανάπτυξη του συστήματος, καθώς επίσης καθιστά δυνατή κάποια πιθανή επέκταση με νέα χαρακτηριστικά: Στο ανώτερο στρώμα διακρίνουμε τις μονάδες που σχετίζονται με τη διεπαφή χρήστη (User's interface). Βρίσκουμε λοιπόν την όψη του συστήματος που βλέπει ο Administrator (Administrator's view), την όψη που βλέπει ο παίκτης μιας αγοράς (Trader's view) καθώς και την όψη που μπορεί να βλέπει κάποιος παρατηρητής της αγοράς (Observer's view). Στο δεύτερο επίπεδο (PM Business Logic) διακρίνουμε τις μονάδες που σχετίζονται με εφαρμογές και λειτουργίες της εικονικής αγοράς, όπως οι διαδικασίες συναλλαγών, ενώ επιπλέον βρίσκουμε και βοηθητικές μονάδες που επεκτείνουν τις δυνατότητες του συστήματος, και προσθέτουν χαρακτηριστικά σχεδιασμένα για τον administrator προκειμένου να συντονίζει τους παίκτες και την αγορά, καθώς και να αναλύει και να αξιολογεί τις λειτουργίες και τα αποτελέσματα της προγνωστικής αγοράς. Το τρίτο στρώμα έχει να κάνει με την αποθήκευση των δεδομένων που απαιτούνται για τη λειτουργία της προγνωστικής αγοράς.

Η σύνδεση του IDeM με το σύστημα ομαδοποίησης κειμένου θα γίνει στο δεύτερο επίπεδο, αφού η διαδικασία της ομαδοποίησης όπως περιγράφηκε στις λειτουργικές προδιαγραφές του IDeM αποτελεί λειτουργία με την οποία θα ασχολείται ο administrator για να διευκολύνει τη διαχείριση των ιδεών για την εισαγωγή τους στις αγορές. Το σύστημα της ομαδοποίησης, όπως αναφέραμε, αποτελεί επέκταση της λειτουργίας ομαδοποίησης του Weka. Έτσι, θα πρέπει να συνδέσουμε τις λειτουργίες διαχείρισης του administrator (Market and GUI Configuration) με τις λειτουργίες του Weka. Η νέα αρχιτεκτονική του συστήματός μας φαίνεται στο ακόλουθο σχήμα:



Εικόνα 4-4 : Η νέα αρχιτεκτονική του συστήματος

Τεχνικά, το IDeM είναι μια web εφαρμογή που ακολουθεί το πρότυπο αρχιτεκτονικής "thin-client", στο οποίο οι περισσότερες δραστηριότητες επεξεργασίας βρίσκονται στο server. Έτσι ο κώδικας στη μεριά του client περιορίζεται σε απλή JavaScript και flash animations, ενώ η αποθήκευση γίνεται με τη χρήση cookies. Η αρχιτεκτονική του IDeM είναι διαμορφωμένη σε τέσσερα επίπεδα: αυτό του πελάτη (Client), της παρουσίασης των δεδομένων (Presentation), της λογικής της αγοράς (Business logic) και τα δεδομένα (Data):

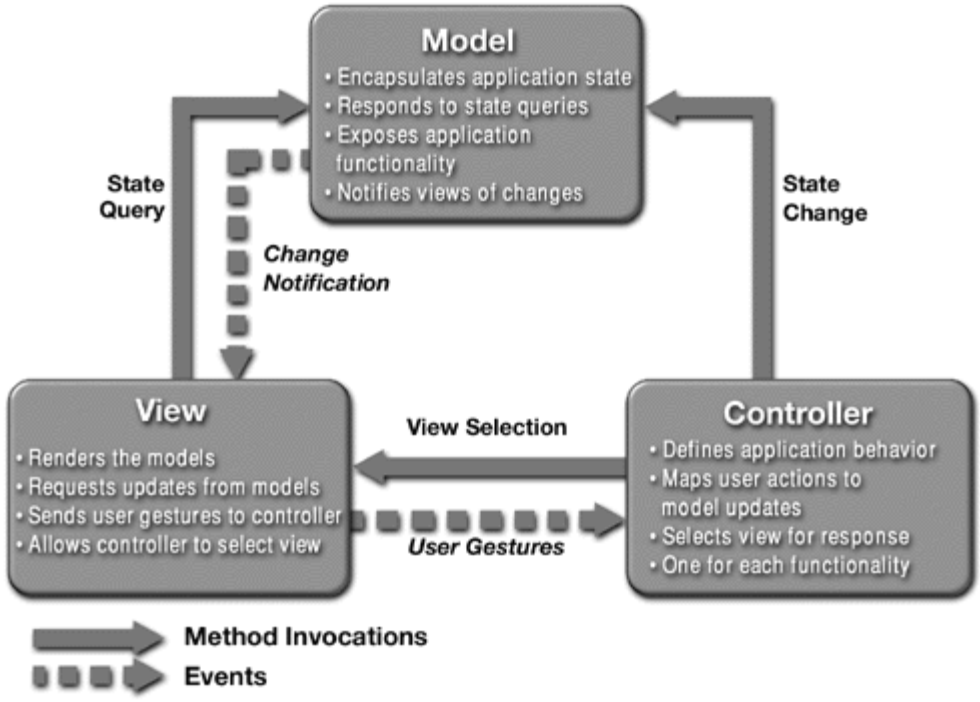
Client tier: Περιλαμβάνει τις εφαρμογές στο στρώμα πελάτη, οι περισσότερες εκ των οποίων μπορούν να εκτελεσθούν με τη χρήση ενός απλού Web Browser

Presentation: το επίπεδο αυτό περιλαμβάνει την παρουσίαση των δεδομένων μέσω διεπαφών χρήστη.

Business Logic: αυτό το κομμάτι είναι υπεύθυνο για τις αιτήσεις των clients και την αποστολή κατάλληλων απαντήσεων μέσω HTML.

Data Layer: Το στρώμα δεδομένων αποτελείται από τις αποθήκες των δεδομένων (σε κάποια βάση δεδομένων) και σχετικές κλάσεις που ορίζουν την πρόσβαση στα δεδομένα.

Ο τεχνικός σχεδιασμός του IDeM έχει κατασκευαστεί στην πλατφόρμα Ruby on Rails (RoR) και ακολουθεί το σχεδιαστικό πρότυπο στο οποίο είναι βασισμένη, Model-View-Controller (MVC). Το πρότυπο MVC βασίζεται σε ένα ξεκάθαρο διαχωρισμό των αντικειμένων σε τρεις κατηγορίες: μοντέλα (models) για τη διαχείριση δεδομένων, όψεις (views) για την απεικόνιση όλων ή κάποιου μέρους των δεδομένων, και ελεγκτές (controllers) για τη διαχείριση των γεγονότων (events) που επηρεάζουν τα models ή τα views.



Εικόνα 4-5 : Απεικόνιση του προτύπου Model-View-Controller

Τα γεγονότα γενικά προκαλούν τον controller να κάνει κάποια αλλαγή στα δεδομένα (στο model) ή σε μια απεικόνισή τους (στο view), ή και στα δύο. Όποτε ένας controller αλλάζει τα δεδομένα ή τις ιδιότητες ενός model, όλες οι σχετικές ενημερώνονται, και αντίστοιχα όποτε ο controller αλλάζει μια όψη, αυτή για να ανανεωθεί χρησιμοποιεί τα δεδομένα του αντίστοιχου μοντέλου.

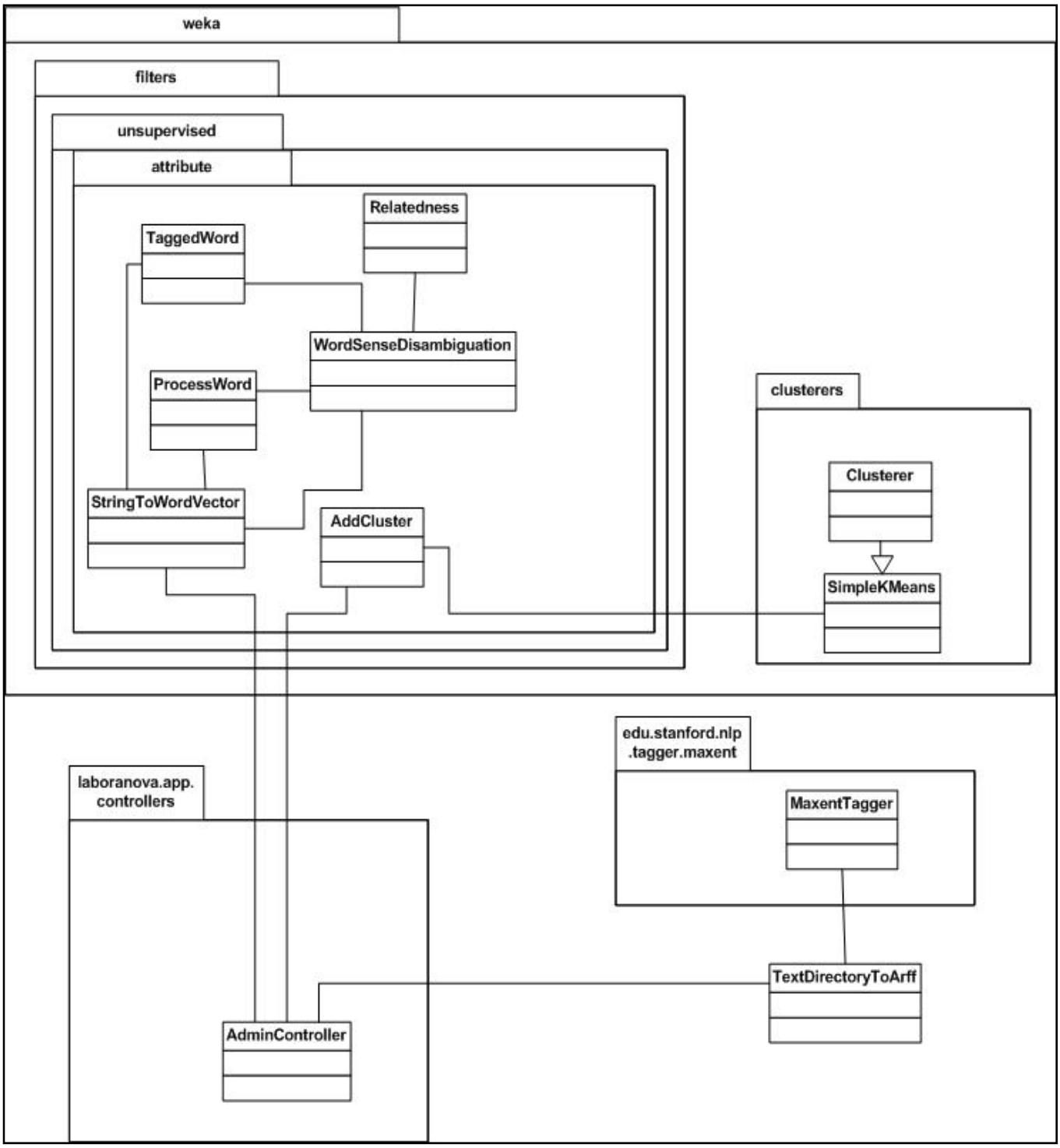
Στο IDeM, τα αντικείμενα controller είναι υπεύθυνα για την εφαρμογή του κομματιού business logic. Οι κλάσεις model περιλαμβάνουν τη διαχείριση των δεδομένων της αγοράς από το λογισμικό.

Όπως προαναφέρθηκε, η ανάπτυξη του IDeM έγινε στην πλατφόρμα Ruby on Rails, το οποίο αποτελεί περιβάλλον ανάπτυξης εφαρμογών web με τη χρήση της γλώσσας προγραμματισμού Ruby. Η εξέλιξη της JRuby (εφαρμογή της γλώσσας Ruby για χρήση της

Java) δίνει τη δυνατότητα ενσωμάτωσης εφαρμογών Java στο σύστημα. Με αυτή τη βάση ξεκινήσαμε την ενσωμάτωση του συστήματος ομαδοποίησης στο IDeM.

Οι ιδέες βρίσκονται στη βάση δεδομένων των ιδεών στο στρώμα τις αποθήκης δεδομένων. Έτσι η λειτουργία της ομαδοποίησης των ιδεών συνίσταται στην ανάκτηση των ιδεών από τη βάση, τη μετατροπή τους σε αρχεία κειμένου και την επεξεργασία τους από τις κλάσεις που διαμορφώσαμε στο Weka. Η διασύνδεση αυτή γίνεται στο επίπεδο των controllers, που είπαμε ότι αντιστοιχεί στο επίπεδο business logic,όπου μεταξύ άλλων συμπεριλαμβάνονται οι λειτουργίες του administrator, στις οποίες κατατάσσεται και η ομαδοποίηση των ιδεών.

Συνεπώς, με την μετατροπή των ιδεών που βρίσκονται στη βάση σε αρχεία κειμένου (txt) και τη μετέπειτα κλήση των κλάσεων σχετικά με την ομαδοποίηση, που βρίσκονται μέσα στη βιβλιοθήκη του Weka, με τη χρήση της JRuby, μπορούμε να επιτύχουμε την ενσωμάτωση της λειτουργίας της ομαδοποίησης ιδεών στις λειτουργίες του administrator. Η ενσωμάτωση, όπως είναι προφανές, θα γίνεται μέσα στον κώδικα του admin_controller, που είναι ο controller που διαχειρίζεται τις λειτουργίες του administrator, επομένως και ελέγχει τη διασύνδεση των σχετικών δεδομένων (ιδέες, δεδομένα της αγοράς, χρήστες, κλπ) με τις όψεις που βλέπει ο administrator. Τα παραπάνω μπορούμε να τα δούμε πώς εφαρμόζονται με το ακόλουθο διάγραμμα κλάσεων, το οποίο μας δείχνει πώς έχει γίνει η ενσωμάτωση της ομαδοποίησης στο IDeM σε επίπεδο κλάσεων.



Εικόνα 4-6 : Διάγραμμα κλάσεων που δείχνει τη διασύνδεση του IDeM μέσω του AdminController με τις κλάσεις για την ομαδοποίηση κειμένου που αναπτύχθηκαν στο Weka

5 Χρήση και Αξιολόγηση του Συστήματος

Στο κεφάλαιο αυτό θα ακολουθήσει παρουσίαση της χρήσης του συστήματος ομαδοποίησης κειμένων που δημιουργήσαμε, καθώς και η αξιολόγηση του συστήματος ως προς την αποτελεσματικότητα και την αποδοτικότητα του. Επειδή όπως αναφέραμε και πριν, το σύστημα της ομαδοποίησης αρχείων κειμένου υλοποιήθηκε με τρόπο ώστε να λειτουργεί και αυτόνομα αλλά και να μπορεί να ενσωματωθεί με κάποιο άλλο σύστημα, θα παρουσιάσουμε δύο τρόπους χρήσης: τη χρήση του συστήματος ομαδοποίησης αρχείων κειμένου ως ένα ανεξάρτητο σύστημα, καθώς και τη χρήση της λειτουργίας του IDeM για την ομαδοποίηση ιδεών.

5.1 Η πορεία χρήσης του συστήματος ομαδοποίησης αρχείων κειμένου

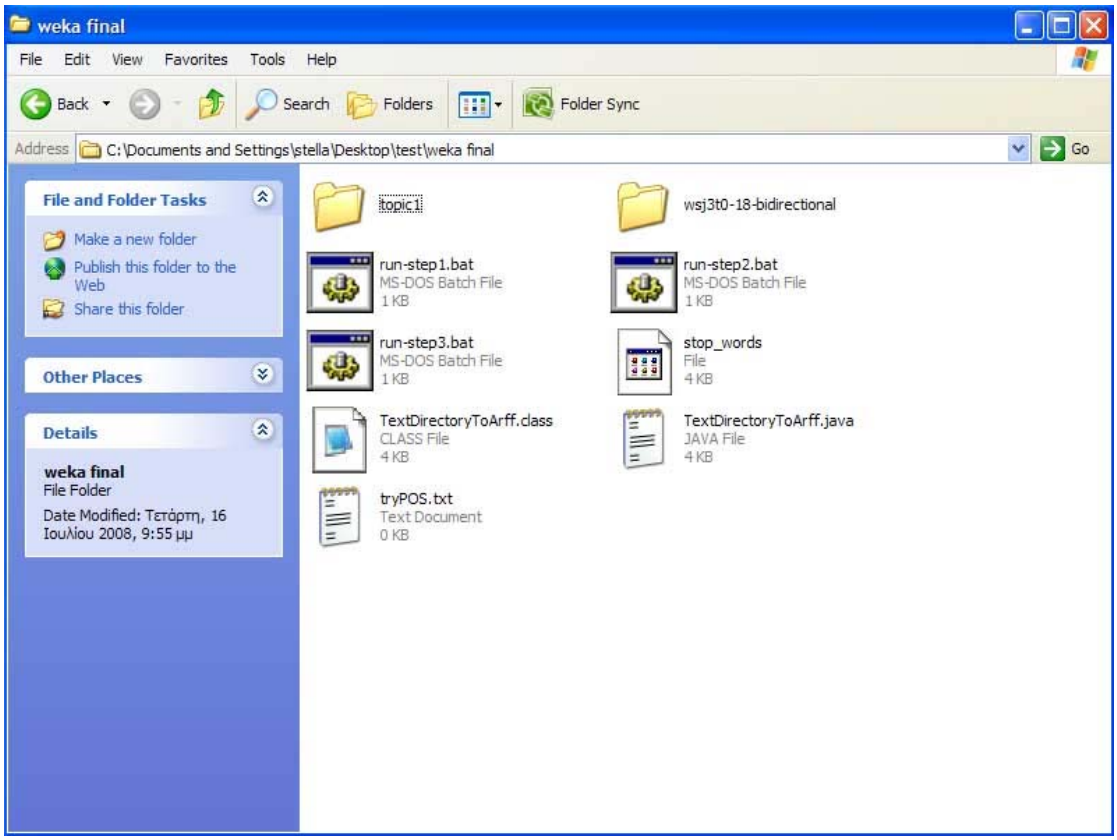
Για τη διαδικασία της ομαδοποίησης αρχείων κειμένου, ο χρήστης χρειάζεται να έχει στον υπολογιστή του εγκατεστημένη κάποια έκδοση της Java, τον κώδικα του Weka με τις κλάσεις στις οποίες εργαστήκαμε για την ανάπτυξη της ομαδοποίησης αρχείων κειμένου, καθώς και τα jar (συμπιεσμένα αρχεία κώδικα Java) της Java version του WordNet και του Stanford Log-Linear Part-Of-Speech Tagger, όπως αναφερθήκαμε σε αυτά στα κεφάλαια 3 και 4.

Στη συνέχεια, για την εκτέλεση της ομαδοποίησης, πρέπει να έχει τα αρχεία κειμένου τα οποία θέλει να ομαδοποιήσει αποθηκευμένα σε μορφή txt σε ένα κοινό φάκελο. Η

διαδικασία της ομαδοποίησης τότε αποτελείται από την εκτέλεση τριών εκτελέσιμων αρχείων, που αντιστοιχούν στη μετατροπή των αρχείων κειμένου σε μορφή αρχείου arff, στην προ-επεξεργασία των αρχείων κειμένου για την εξαγωγή χαρακτηριστικών, και στην εκτέλεση του αλγορίθμου ομαδοποίησης.

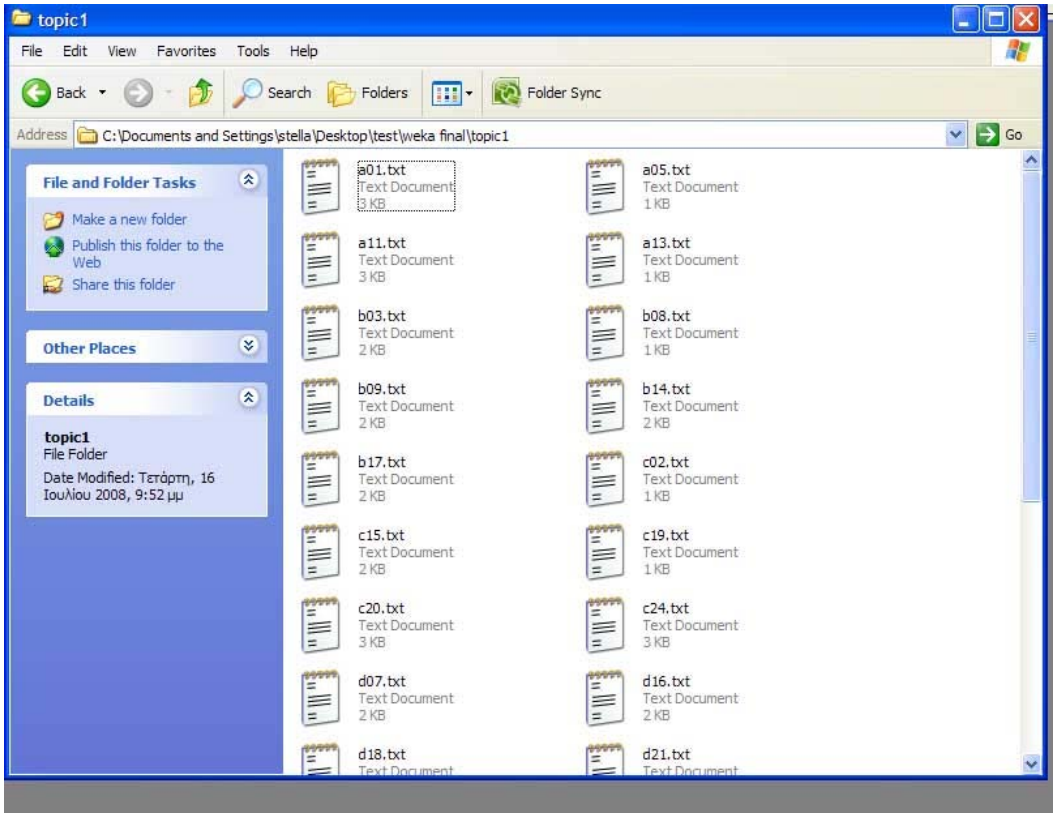
Ας δούμε αυτή τη διαδικασία σε ένα παράδειγμα:

Έχουμε τον ακόλουθο φάκελο με τα εκτελέσιμα αρχεία για τη διαδικασία της ομαδοποίησης:



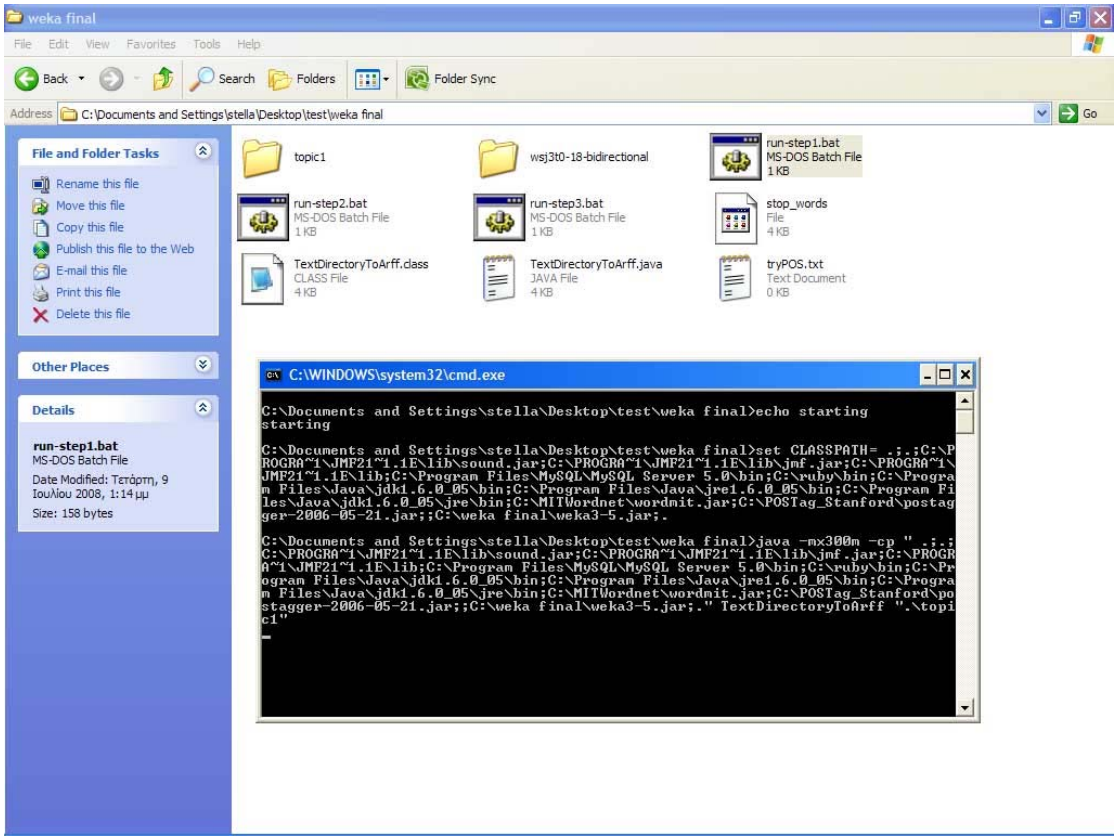
Εικόνα 5-1 : Ο φάκελος με τα αρχεία για την ομαδοποίηση

Στο φάκελο περιέχεται ο φάκελος topic1, ο οποίος περιέχει τα αρχεία κειμένου σε μορφή txt, τα οποία θέλουμε να ομαδοποιήσουμε.



Εικόνα 5-2 : Ο φάκελος με τα αρχεία που θέλουμε να ομαδοποιήσουμε

Για να ξεκινήσουμε τη διαδικασία, εκτελούμε το αρχείο runstep1.bat, το οποίο τρέχει μαζί με την κονσόλα του command prompt.



Εικόνα 5-3 : Εκτέλεση του runstep1.bat

Το αρχείο αυτό ουσιαστικά εκτελεί τις ακόλουθες εντολές:

```
echo starting
set CLASSPATH= %CLASSPATH%;C:\weka final\weka3-5.jar;.
java -mx300m -cp "%CLASSPATH%" TextDirectoryToArff ".\topic1"
echo execution ended
```

Καλείται δηλαδή η εκτέλεση της Java κλάσης TextDirectoryToArff.class, η οποία όπως είπαμε στο 4^{ο} κεφάλαιο τοποθετεί τα περιεχόμενα των αρχείων txt σε ένα αρχείο της μορφής arff, ώστε να μπορεί να διαβαστεί από τις κλάσεις του Weka, ενώ παράλληλα κατά τη μετατροπή αυτή γίνεται και το part-of-speech tagging με την κλήση των κατάλληλων μεθόδων από τον Stanford POS Tagger.

Μετά το τέλος της εκτέλεσης αυτής, έχει δημιουργηθεί το αρχείο new.arff, το οποίο μπορούμε να δούμε ότι έχει την ακόλουθη μορφή:

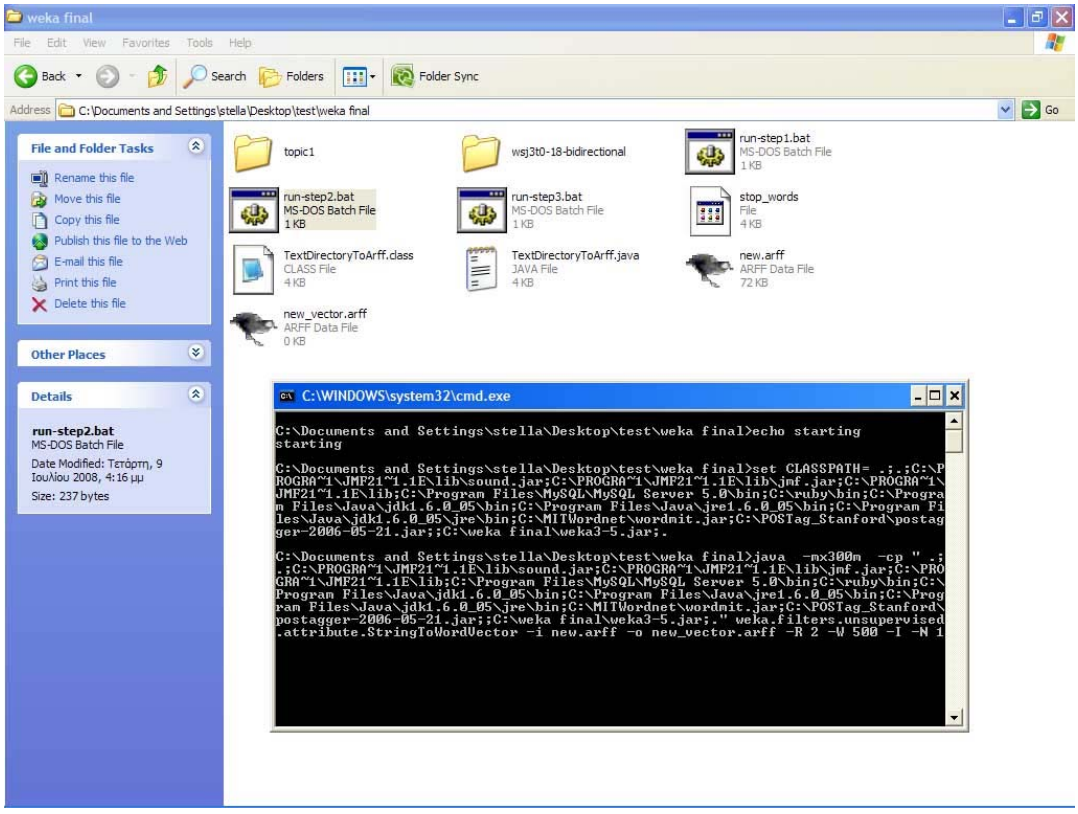
```
@relation 'text_files_in_.\\topic1'

@attribute filename string
@attribute contents string

@data
a01.txt,'Datamash/NNP sends/VBZ instant/JJ data/NNS messages/NNS between/IN different/JJ documents/NNS and/CC applica
a05.txt,'Ticketish/JJ is/VBZ a/DT simple/JJ web-based/JJ system/NN for/IN managing/VBG help/NN tickets/NNS ./. With/II
a11.txt,'Chapter/NNP 1/CD -/: Creating/VBG a/DT Better/NNP Way/NN to/TO Invoice/NNP We/PRP know/VBP billing/VBG is/VB
a13.txt,'What/WP is/VBZ skemma/NN ?/. 1/CD ./. A/DT tool/NN for/IN defining/VBG ,/, using/VBG and/CC improving/VBG bu
b03.txt,'Dehydration/NNP is/VBZ a/DT primary/JJ reason/NN for/IN daytime/JJ fatigue/NN and/CC can/MD slow/VB down/RP
b08.txt,'Gimme20/NNP ./. com/NN is/VBZ a/DT Free/NNP Fitness/NNP Tracking/NNP Tool/NNP and/CC Social/NNP Fitness/NNP
b09.txt,'CarePilot/NNP is/VBZ an/DT e-commerce/NN business/NN focused/VBD on/IN people/NNS considering/VBG elder/JJR
b14.txt,'Nuvora/NNP -/: Oral/NNP Health/NNP Products/NNPS Nuvora/NNP \'s/POS main/JJ focus/NN is/VBZ to/TO develop/VB
b17.txt,'Empowering/VBG everyone/NN in/IN their/PRP$ health/NN care/NN process/NN through/IN personalized/VBN search/I
c02.txt,'buzzoop/NN is/VBZ a/DT social/JJ cataloging/VBG service/NN that/WDT helps/VBZ you/PRP catalog/NN and/CC retr
c15.txt,'atlaspost/NN is/VBZ a/DT new/JJ social/JJ network/NN service/NN that/WDT integrates/VBZ Google/NNP maps/NNS
c19.txt,'Jumpsocial/NNP The/DT idea/NN is/VBZ to/TO create/VB a/DT community/NN calendar/NN that/WDT is/VBZ locale/NN
c20.txt,'Borrow/VB ,/, organize/VB ,/, recycle/VB ,/, share/NN ,/, meet/VB ,/, discuss/VB ,/, and/CC save/VB ./. Save,
c24.txt,'Do/NNP you/PRP remember/VBP being/VBG frustrated/VBN with/IN the/DT inability/NN to/TO find/VB relevant/JJ r
d07.txt,'SportsTwo/NNP is/VBZ a/DT sports/NNS news/NN driven/VBN community/NN www/NN site/NN platform/NN ./. Very/RB I
d16.txt,'ROCKETON/NNP is/VBZ a/DT venture-funded/JJ startup/NN that/IN is/VBZ developing/VBG a/DT new/JJ type/NN of/II
d18.txt,'TheSportsTV.com/NNP is/VBZ launching/VBG a/DT revolutionary/JJ online/NN sports/NNS community/NN that/IN has,
d21.txt,'Do/VBP n\'t/RB let/VB our/PRP$ low/JJ keenness/NN fool/NN you/PRP ./. MP3/CD ./. net/JJ is/VBZ launching/VBG
d27.txt,'VIDSZOO.com/NNP is/VBZ the/DT next/JJ generation/NN of/IN online/JJ video/NN sharing/NN and/CC after/IN only,
e04.txt,'This/DT is/VBZ Our/PRP$ Planet/NN ,/, Re-Volt/NNP -LRB-/-LRB- tm/NN -RRB-/-RRB- with/IN a/DT Flock/NN of/IN
e06.txt,'Solar/NNP Energy/NNP -/: What/WP is/VBZ it/PRP ?/. -LRB-/-LRB- 1Spig/NNP \\//CD 1Scrap/NNP -RRB-/-RRB- poste
e10.txt,'GluNetworks/NNPS is/VBZ a/DT global/JJ SaaS/NNP company/NN with/IN the/DT mission/NN to/TO use/VB the/DT con
e22.txt,'The/DT IC/NNP piston/NN engine/NN and/CC the/DT Environment/NNP ./. -LRB-/-LRB- 2Spig/JJ \\//NN 0Scrap/NNP -
f12.txt,'ON/IN THE/DT FLY/SYM -/: Solutions/NNPS for/IN the/DT modern/JJ gentleman/NN On/IN The/DT Fly/VB is/VBZ comm
f23.txt,'JibberJobber/NNP is/VBZ your/PRP$ private/JJ ,/, personal/JJ tool/NN to/TO manage/VB all/DT of/IN the/DT inf
f25.txt,'AdSymetrix/NNP Helps/VBZ Your/PRP$ Business/NNP Make/NNP Smarter/NNP Marketing/NNP Decisions/NNS AdSymetrix/I
g26.txt,'ki/NN work/NN is/VBZ a/DT marketplace/NN ,/, search/NN facility/NN and/CC operational/JJ platform/NN for/IN
g28.txt,'What/WP is/VBZ Iceberg/NNP ?/. Iceberg/NN is/VBZ a/DT 100/CD \%/NN web/NN based/VBN platform/NN for/IN build
g29.txt,'Unlimited/JJ Online/NNP Storage/NNP ,/, Sharing/VBG ,/, and/CC Access/NNP ElephantDrive/NNP provides/VBZ the,
g30.txt,'auditoriumA.com/NN is/VBZ the/DT premier/NN destination/NN for/IN human/JJ guided/VBN web/NN exploration/NN
```

Εικόνα 5-4 : Το αρχείο new.arff

Ακολούθως, είμαστε σε θέση να εκτελέσουμε το επόμενο βήμα, τρέχοντας το εκτελέσιμο runstep2.bat:



Εικόνα 5-5 : Εκτέλεση του runstep2.bat

Το αρχείο αυτό εκτελεί τις ακόλουθες εντολές:

```
echo starting
set CLASSPATH= %CLASSPATH%;C:\weka final\weka3-5.jar;.
java -mx300m -cp "%CLASSPATH%"
weka.filters.unsupervised.attribute.StringToWordVector -i new.arff -o
new_vector.arff -R 2 -W 500 -I -N 1
echo execution ended
```

Καλούμε δηλαδή την κλάση StringToWordVector του Weka, την οποία όπως αναφέραμε στο προηγούμενο κεφάλαιο διαμορφώσαμε προσθέτοντας επιπλέον λειτουργίες χρήσιμες για την ομαδοποίηση των αρχείων κειμένου. Η κλάση StringToWordVector παίρνει ως είσοδο το αρχείο new.arff που περιέχει τα ακατέργαστα (εκτός του part-of-speech tagging) κείμενα προς επεξεργασία για την εξαγωγή των χαρακτηριστικών και τελικά τη δημιουργία του πίνακα όρων εγγραφών, που θα αποθηκευθεί στο αρχείο εξόδου new_vector.arff. Επιπλέον, μαζί με την κλήση της κύριας μεθόδου της κλάσης StringToWordVector δίνουμε τις παραμέτρους:

–R 2 : σημαίνει ότι θα διαβάζονται 2 string attributes από κάθε instance. Όπως έχουμε προαναφέρει, κάθε instance αντιστοιχεί σε ένα αρχείο κειμένου, και έχει 2 string attributes, ένα που περιέχει το όνομα του αρχείου και ένα που περιέχει το κείμενο του αρχείου.

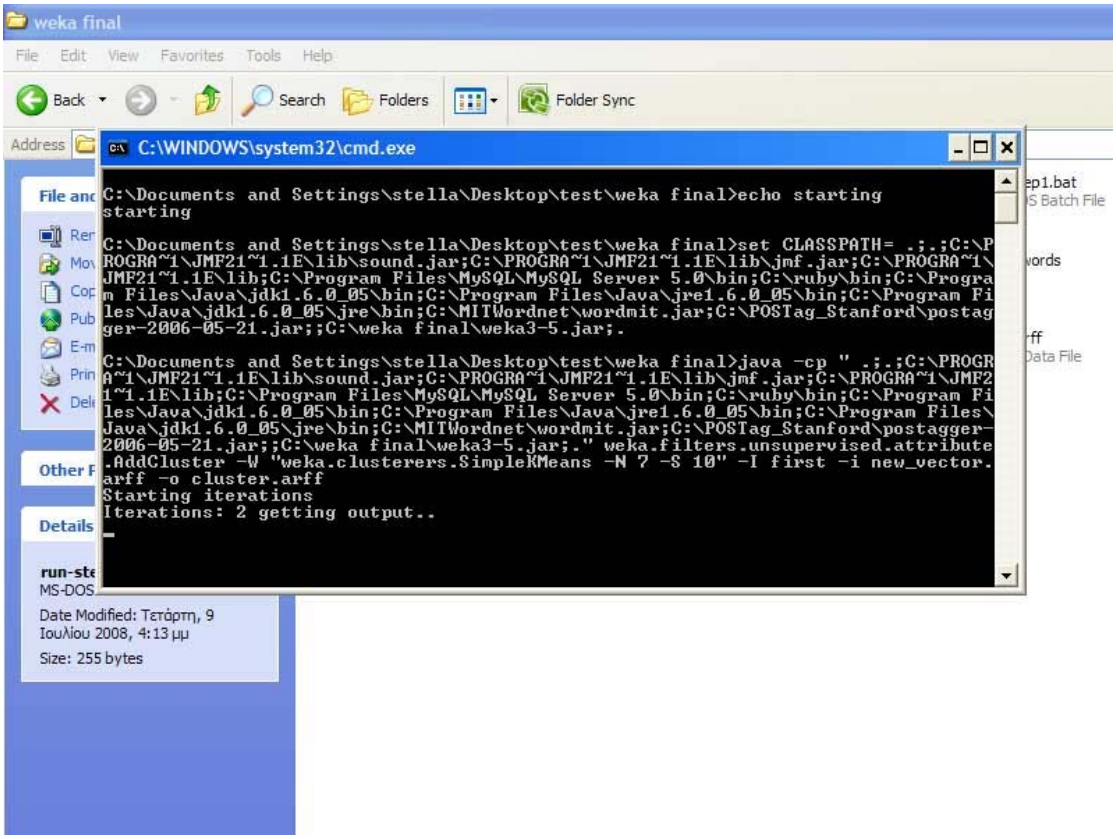
–W 500: σημαίνει ότι από κάθε κείμενο θα διαβάζονται και θα επεξεργάζονται για την εξαγωγή των όρων, οι πρώτες 500 λέξεις.

–I: σημαίνει ότι τα βάρη των όρων θα είναι σύμφωνα με τον τύπο TF-IDF, όπως αυτός περιγράφηκε στο προηγούμενο κεφάλαιο.

–N 1: σημαίνει ότι τα βάρη των όρων θα έχουν στον υπολογισμό τους επιπλέον την κανονικοποίηση ως προς το μέσο μήκος αρχείου κειμένου.

Μετά το τέλος της εκτέλεσης αυτής, έχει δημιουργηθεί το αρχείο new_vector.arff, το οποίο περιέχει τον πίνακα όρων-εγγράφων, δηλαδή τα στοιχεία της διανυσματικής αναπαράστασης των αρχείων κειμένου.

Στη συνέχεια, είμαστε σε θέση να εκτελέσουμε το τρίτο και τελευταίο βήμα της διαδικασίας, αυτό της ομαδοποίησης. Τρέχουμε λοιπόν το runstep3.bat:



Εικόνα 5-6 : Εκτέλεση του runstep3.bat

Το αρχείο αυτό εκτελεί τις ακόλουθες εντολές:

```
echo starting
set CLASSPATH= %CLASSPATH%;C:\weka final\weka3-5.jar;.
java -cp "%CLASSPATH%" weka.filters.unsupervised.attribute.AddCluster
-W "weka.clusterers.SimpleKMeans -N 7 -S 10" -I first -i
new_vector.arff -o cluster.arff
echo execution ended
```

Καλεί δηλαδή την κλάση AddCluster του Weka, η οποία παίρνει ως αρχείο εισόδου το new_vector..arff, (-i new_vector.arff), επεξεργάζεται και ομαδοποιεί τα instances που περιέχει (άρα τα διανύσματα-αρχεία κειμένου) με βάση τον αλγόριθμο που της δίνεται ως παράμετρος –άρα στην περίπτωσή μας λόγω της παραμέτρου -W "weka.clusterers.SimpleKMeans -N 7 -S 10" τον SimpleKMeans με τις παραμέτρους –N 7 ως επιθυμητό αριθμό clusters και –S 10 ως τυχαίο αριθμό (seed) για την εκτέλεση του αλγορίθμου – και δίνει έξοδο ένα νέο αρχείο arff (-o cluster.arff) το οποίο περιέχει τα instances του αρχείου εισόδου με ένα επιπλέον attribute το καθένα, το cluster στο οποίο ομαδοποιήθηκαν.

Επιπλέον, καθώς εκτελείται ο αλγόριθμος SimpleKMeans, εκτυπώνει τα αποτελέσματα της ομαδοποίησης στο αρχείο κειμένου KMEANS_CLUSTERS.txt, έτσι ώστε και να είναι ευανάγνωστα για τον απλό χρήστη, αλλά και να μπορούν να υποστούν μετέπειτα επεξεργασία από κάποια άλλη εφαρμογή που ενσωματώνει τη λειτουργία του συστήματος ομαδοποίησης αρχείων κειμένου που δημιουργήσαμε.

Τα αποτελέσματα για το συγκεκριμένο παράδειγμα τα βλέπουμε παρακάτω, έτσι ώστε να δούμε τη μορφή και δομή του αρχείου KMEANS_CLUSTERS.txt:

```
CLUSTER 1
 inst 18 ,  inst 20 ,  inst 21 ,  inst 29 ,
CLUSTER 2
 inst 28 ,
CLUSTER 3
 inst 3 ,  inst 5 ,  inst 6 ,  inst 7 ,  inst 8 ,  inst 9 ,  inst 11
,  inst 12 ,  inst 13 ,  inst 15 ,  inst 16 ,  inst 17 ,  inst 22 ,
inst 27 ,
CLUSTER 4
 inst 0 ,  inst 1 ,  inst 2 ,  inst 4 ,  inst 10 ,  inst 14 ,  inst
19 ,  inst 26 ,
CLUSTER 5
 inst 25 ,
CLUSTER 6
 inst 24 ,
CLUSTER 7
 inst 23 ,


Instance 0 a01.txt Cluster 4
Instance 1 a05.txt Cluster 4
Instance 2 a11.txt Cluster 4
Instance 3 a13.txt Cluster 3
Instance 4 b03.txt Cluster 4
Instance 5 b08.txt Cluster 3
Instance 6 b09.txt Cluster 3
Instance 7 b14.txt Cluster 3
Instance 8 b17.txt Cluster 3
Instance 9 c02.txt Cluster 3
Instance 10 c15.txt Cluster 4
Instance 11 c19.txt Cluster 3
Instance 12 c20.txt Cluster 3
Instance 13 c24.txt Cluster 3
Instance 14 d07.txt Cluster 4
Instance 15 d16.txt Cluster 3
Instance 16 d18.txt Cluster 3
Instance 17 d21.txt Cluster 3
Instance 18 d27.txt Cluster 1
```

```
Instance 19 e04.txt Cluster 4
Instance 20 e06.txt Cluster 1
Instance 21 e10.txt Cluster 1
Instance 22 e22.txt Cluster 3
Instance 23 f12.txt Cluster 7
Instance 24 f23.txt Cluster 6
Instance 25 f25.txt Cluster 5
Instance 26 g26.txt Cluster 4
Instance 27 g28.txt Cluster 3
Instance 28 g29.txt Cluster 2
Instance 29 g30.txt Cluster 1


STATISTICS:
Cluster 1: 4.0 instances (13.333333333333334%)
Cluster 2: 1.0 instances (3.3333333333333335%)
Cluster 3: 14.0 instances (46.666666666666664%)
Cluster 4: 8.0 instances (26.666666666666668%)
Cluster 5: 1.0 instances (3.3333333333333335%)
Cluster 6: 1.0 instances (3.3333333333333335%)
Cluster 7: 1.0 instances (3.3333333333333335%)
```

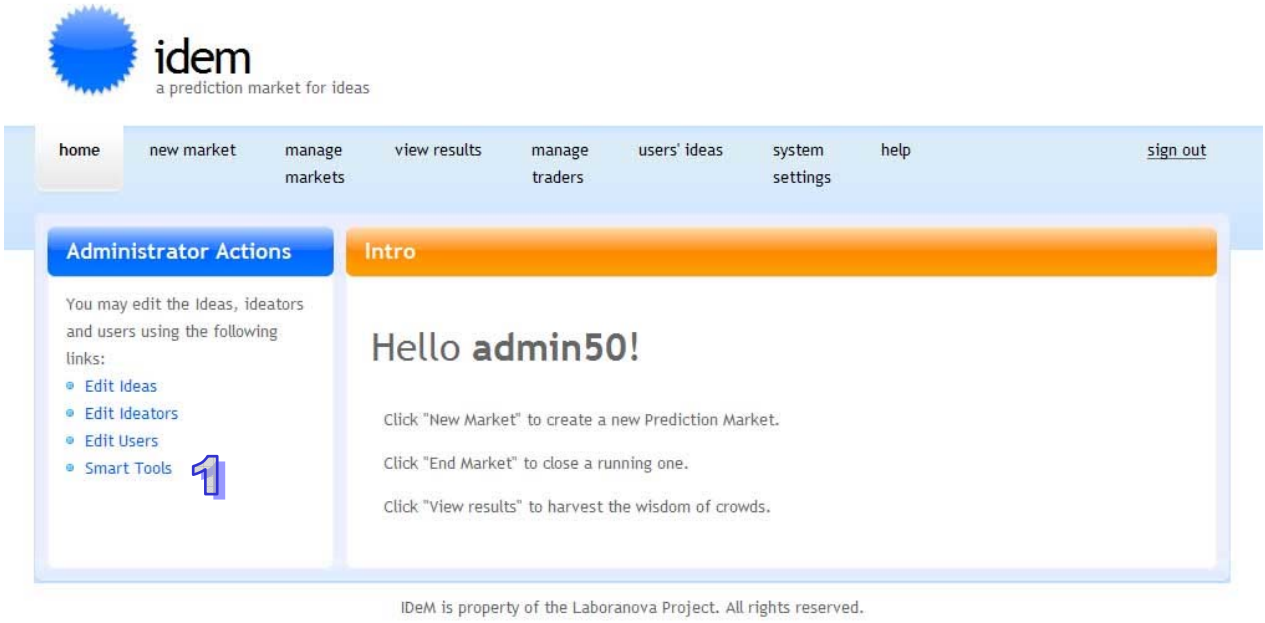
5.2 Η πορεία χρήσης του συστήματος ομαδοποίησης ιδεών

Ας δούμε τώρα τη λειτουργία της ομαδοποίησης ιδεών του IDeM, που απευθύνεται στον Market Administrator, όπως περιγράφηκε στις λειτουργικές προδιαγραφές στο προηγούμενο κεφάλαιο.

Αρχικά, ο χρήστης βλέπει τη σελίδα εισόδου στο IDeM, όπως βλέπουμε στην εικόνα 5.7. Εκεί, βάζει το όνομα χρήστη και τον κωδικό πρόσβασης, με τα οποία, δεδομένου ότι έχει τα αντίστοιχα δικαιώματα εξουσιοδότησης, θα εισαχθεί στο σύστημα ως Market Administrator.

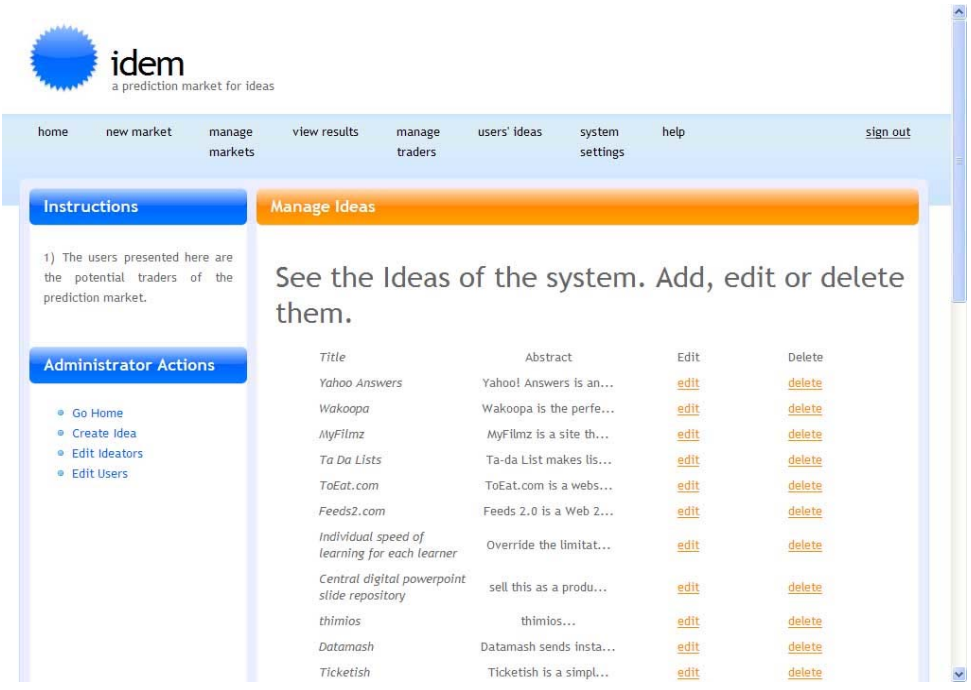
Εικόνα 5-7 : Σελίδα εισόδου στο IDeM

Αφού ο market administrator μπει στο σύστημα, βλέπει τη βασική home page, στο πάνω μέρος της οποίας μπορεί να πλοηγηθεί σε άλλες όψεις ώστε να δημιουργήσει μια νέα αγορά (new market), να διαχειριστεί τις υπάρχουσες αγορές (manage markets), να δει την έκβαση μιας αγοράς προβλέψεων (view results), να διαχειριστεί τους παίκτες των αγορών (manage traders), να διαχειριστεί τις ιδέες των χρηστών (users' ideas), να αλλάξει ρυθμίσεις του συστήματος (system settings), και τέλος να αναζητήσει βοήθεια για το σύστημα αλλά και για τα prediction markets (help).



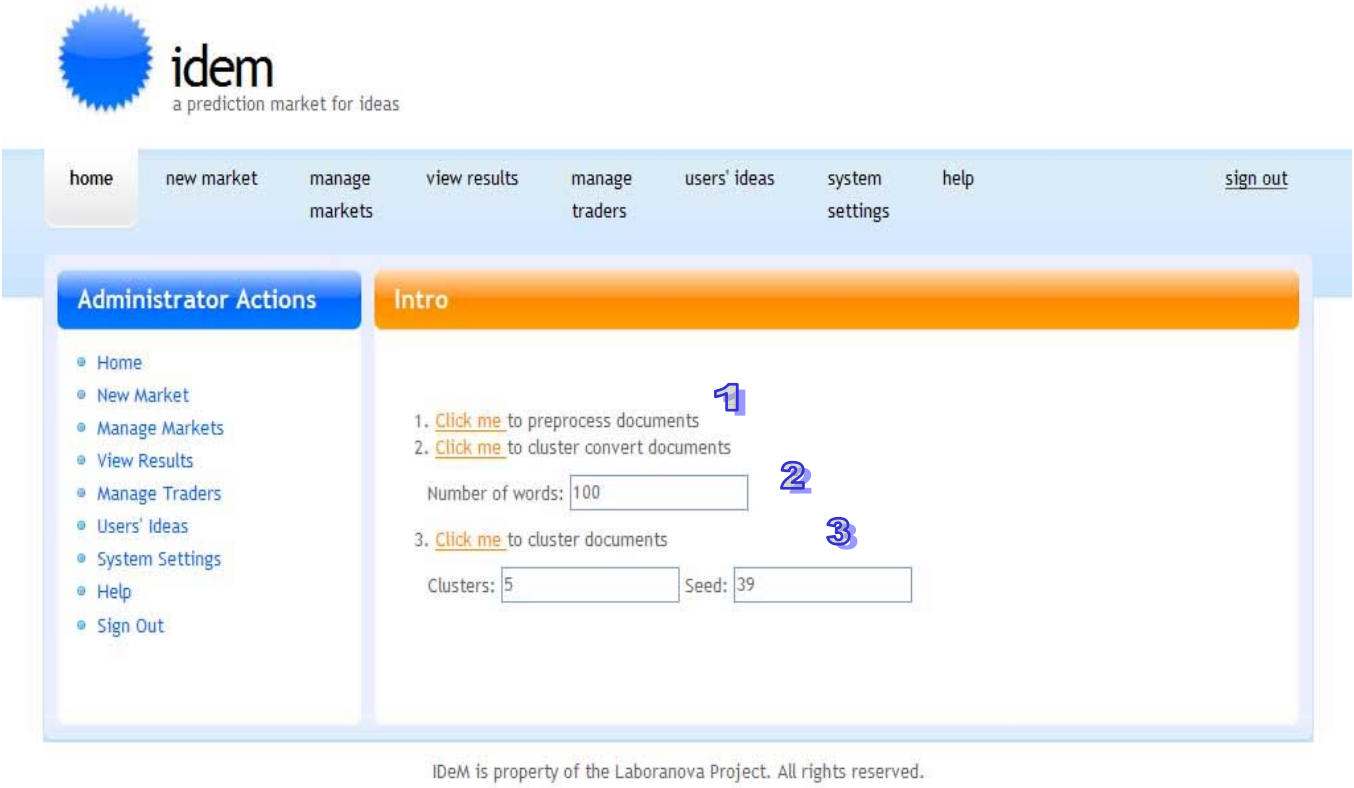
Εικόνα 5-8 : Η αρχική σελίδα του market administrator

Ο market administrator μπορεί να δει τις ιδέες που υπάρχουν στο σύστημα από την ακόλουθη σελίδα, ενώ επίσης τις βλέπει και όταν θέλει να κάνει επιλογή των ιδεών που θα μπουν στην έναρξη μιας αγοράς:



Εικόνα 5-9 : Απεικόνιση των ιδεών των χρηστών του συστήματος

Στην περίπτωση που ενδιαφέρεται να ομαδοποιήσει αυτές τις ιδέες, ο market administrator μπορεί να πάει στη σελίδα Smart Tools, κάνοντας κλικ στην αντίστοιχη επιλογή του αριστερού μενού της αρχικής σελίδας (σημείο 1 στην εικόνα 5.8).



Εικόνα 5-10 : Η σελίδα Smart Tools στην οποία ο administrator μπορεί να κάνει ομαδοποίηση των ιδεών

Εκεί, ο administrator μπορεί να εφαρμόσει την ομαδοποίηση των ιδεών, ακολουθώντας τα βήματα της ομαδοποίησης:

Στο 1^{ο} βήμα (το κομμάτι της εικόνας 5.10 με τον αριθμό 1), ο administrator κάνει κλικ προκειμένου να γίνει η πρώτη προ-επεξεργασία, δηλαδή η μετατροπή των ιδεών από αρχεία κειμένου σε μορφή arff.

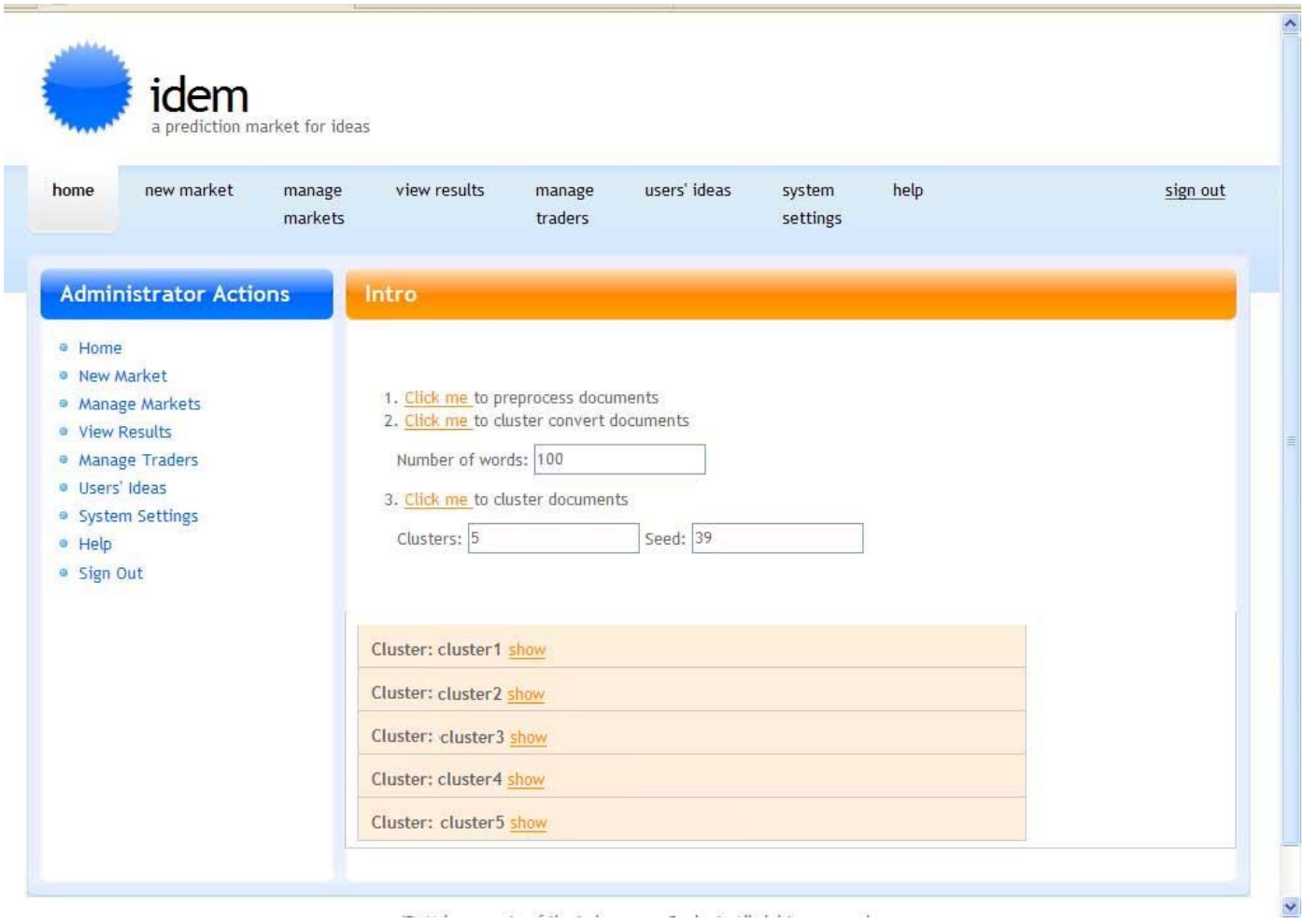
Μόλις ανανεωθεί η σελίδα, ο administrator μπορεί να επιλέξει το 2^{ο} βήμα (το κομμάτι της εικόνας 5.10 με τον αριθμό 2), ώστε να γίνει η δεύτερη προ-επεξεργασία, δηλαδή η εξαγωγή γνωρισμάτων και η δημιουργία του πίνακα όρων-εγγράφων, προκειμένου να

αναπαρασταθούν τα αρχεία. Στο βήμα αυτό, ο χρήστης επιλέγει τον αριθμό των λέξεων που επιθυμεί να διαβαστούν από κάθε κείμενο ώστε να υποστούν επεξεργασία για τη διαδικασία της ομαδοποίησης.

Όταν ολοκληρωθεί η διαδικασία της προ-επεξεργασίας, η σελίδα ανανεώνεται και πάλι. Τότε ο administrator μπορεί να επιλέξει το $3^{ο}$ βήμα (το κομμάτι της εικόνας 5.10 με τον αριθμό 3) όπου διαλέγει πόσα clusters επιθυμεί να δημιουργηθούν κατά την ομαδοποίηση καθώς και ένα τυχαίο αριθμό (seed) που θα χρησιμοποιηθεί από τον αλγόριθμο ομαδοποίησης, και κάνει κλικ για να ξεκινήσει η διαδικασία.

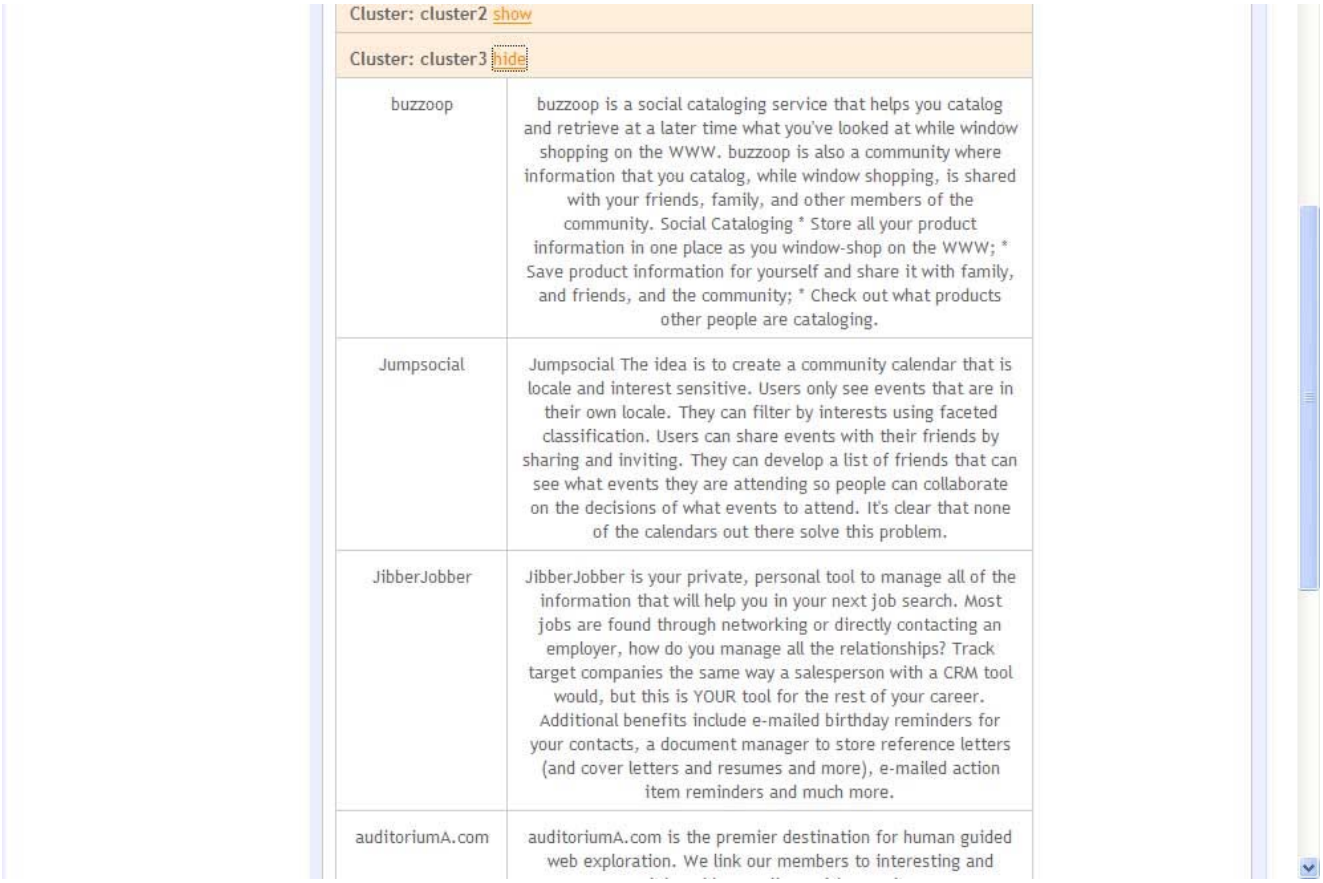
Όταν ολοκληρωθεί η ομαδοποίηση, η σελίδα ανανεώνεται και εμφανίζει τον πίνακα με τα clusters που έχουν δημιουργηθεί.

Για την περίπτωση της ομαδοποίησης με τις παραμέτρους της εικόνας 5.10 το αποτέλεσμα φαίνεται στην ακόλουθη εικόνα:



Εικόνα 5-11 : Το αποτέλεσμα της ομαδοποίησης

Πατώντας στην επιλογή show δίπλα από κάθε cluster, ο χρήστης μπορεί να δει ποιες ιδέες έχουν αντιστοιχηθεί στην αντίστοιχη ομάδα. Για παράδειγμα, για το cluster3 του παραδείγματός μας η εικόνα της σελίδας είναι όπως ακολούθως:



Εικόνα 5-12 : Οι ιδέες που έχουν αντιστοιχηθεί στο cluster3

5.3 Μεθοδολογία ελέγχου

Για να ελέγξουμε την αποτελεσματικότητα και την αποδοτικότητα του συστήματος ομαδοποίησης αρχείων κειμένου, κάναμε ελέγχους χρησιμοποιώντας κάποια σύνολα αρχείων κειμένου (corpora) ήδη ομαδοποιημένα, προκειμένου να συγκρίνουμε τα αποτελέσματα της δικής μας ομαδοποίησης με τις προϋπάρχουσες ομάδες.

Τα σύνολα αυτά ήταν:

Reuters-21578: Πρόκειται για μια συλλογή αρχείων από το Reuters news, η οποία αποτελείται από 21578 ομαδοποιημένα αρχεία κειμένου. Είναι από τα γνωστότερα corpus που χρησιμοποιούνται σε ελέγχους συστημάτων σχετικών με την επεξεργασία φυσικής γλώσσας, την εξόρυξη γνώσης από δεδομένα, κα. (διαθέσιμο στη σελίδα http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html)

20Newsgroups corpus: Περιέχει γύρω στα 20000 κείμενα (με τη δομή e-mail) ομαδοποιημένα σε 20 κατηγορίες, τα οποία έχουν ανακτηθεί από τη συλλογή των Usenet newsgroups. (διαθέσιμο στη σελίδα http://www.ai.mit.edu/people/jrennie/20Newsgroups/)

The Uppsala Student English Corpus: Διατίθεται από το αρχείο κειμένων του Oxford Text Archive. Αποτελείται από 1489 εκθέσεις γραμμένες στα αγγλικά από Σουηδούς φοιτητές, από τρία διαφορετικά επίπεδα γνώσης αγγλικής γλώσσας.(διαθέσιμο στη σελίδα http://ota.ahds.ac.uk/headers/2457.xml)

Όσον αφορά στη μέτρηση της ακρίβειας των αλγορίθμων, χρησιμοποιήσαμε τις τιμές των μέτρων purity και F-measure. Το F-measure αποτελεί συνδυασμό των τιμών precision και recall που χρησιμοποιούνται στην ανάκτηση γνώσης. Αναφερόμενοι στις έτοιμες ομάδες των corpora ως κλάσεις, ορίζουμε:

n_i: τον αριθμό των αρχείων που ανήκουν στην κλάση i

n_j:τον αριθμό των αρχείων που ανήκουν στην ομάδα j

n_{ij}: τον αριθμό των αρχείων της κλάσης i που ανήκουν στην ομάδα j

n: ο συνολικός αριθμός των αρχείων

Έτσι, έχουμε τα ακόλουθα μέτρα για κάθε ομάδα j για κάθε κλάση i:

Precision: $P(i,j) = \frac{n_{ij}}{n_j}$

Recall: $R(i,j) = \frac{n_{ij}}{n_i}$

F-measure: $F(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)}$

Έτσι, το F-measure για ολόκληρο το αποτέλεσμα της ομαδοποίησης ορίζεται ως:

$$F = \sum_i \frac{n_i}{n} \max_j (F(i,j))$$

Είναι προφανές ότι όσο μεγαλύτερο είναι το F-measure τόσο καλύτερο θεωρείται το αποτέλεσμα της ομαδοποίησης.

Όσον αφορά στο μέτρο Purity, αναπαριστά το κλάσμα της ομάδας που αντιστοιχεί στη μεγαλύτερη κλάση αρχείων που έχουν αντιστοιχηθεί στη συγκεκριμένη ομάδα. Έτσι, το Purity μιας ομάδας j ορίζεται ως:

$$Purity(j) = \frac{1}{n_j} \max_i (n_{ij})$$

Το συνολικό Purity της ομαδοποίησης ορίζεται ως:

$$Purity = \sum_j \frac{n_j}{n} Purity(j)$$

Και πάλι, όσο μεγαλύτερη είναι η τιμή του purity, τόσο καλύτερο θεωρείται το αποτέλεσμα της ομαδοποίησης.

5.4 Αναλυτική παρουσίαση ελέγχου των λειτουργιών της ομαδοποίησης

Αρχικά, διενεργήσαμε διάφορους ελέγχους για να επιβεβαιώσουμε ότι οι λειτουργίες που αναπτύξαμε για την ομαδοποίηση προσέθεταν επιπλέον ακρίβεια στην απλή ομαδοποίηση του Weka.

Έτσι, διενεργήσαμε tests στην προ-επεξεργασία των αρχείων κειμένου για τις ακόλουθες περιπτώσεις :

- Test 0: προ-επεξεργασία με τον κώδικα του Weka
- Test 1: προ-επεξεργασία με την υλοποίηση tokenization
- Test 2: προ-επεξεργασία με την υλοποίηση tokenization και αφαίρεση stopwords
- Test 3:προ-επεξεργασία με την υλοποίηση tokenization και αφαίρεση stopwords και εύρεση συνώνυμων όρων αλλά χωρίς τη χρήση του part-of-speech tagging και word sense disambiguation, δηλαδή να βρίσκονται όλα τα πιθανά συνώνυμα μιας λέξης
- Test 4: προ-επεξεργασία με την υλοποίηση tokenization,αφαίρεση stopwords και εύρεση συνώνυμων όρων με τη χρήση του part-of-speech tagging αλλά χωρίς τη χρήση word sense disambiguation
- Test 5: προ-επεξεργασία με την υλοποίηση tokenization,αφαίρεση stopwords και εύρεση συνώνυμων όρων με τη χρήση του part-of-speech tagging και word sense disambiguation

- Test 6: προ-επεξεργασία με την υλοποίηση tokenization, αφαίρεση stopwords και εύρεση συνώνυμων όρων με τη χρήση του part-of-speech tagging και word sense disambiguation, καθώς και stemming

Ακόμη, διενεργήσαμε ελέγχους της διαδικασίας της ομαδοποίησης έχοντας χρησιμοποιήσει διαφορετικά βάρη για τους όρους και διαφορετικά μέτρα ομοιότητας. Είχαμε συνεπώς τους ακόλουθους τύπους test για την εκτέλεση του αλγορίθμου SimpleKMeans:

- Test a: ως μέτρο ομοιότητας χρησιμοποιείται η Ευκλείδεια απόσταση.
- Test b: ως μέτρο ομοιότητας χρησιμοποιείται το cosine similarity.
- Test c: παίρνοντας ως βάρος κάθε όρου τη συχνότητα των λέξεων, μετατρέπει τα βάρη σύμφωνα με τον τύπο για το TF-IDF που αναφέραμε στο κεφάλαιο 4:

  $$w(d,t) = TF(d,t) * IDF(t) \qquad TF(d,t) = n(d,t) \qquad IDF(t) = \log\left(\frac{D - DF(t)}{DF(t)}\right)$$

  Ως μέτρο ομοιότητας χρησιμοποιείται η Ευκλείδεια απόσταση.
- Test d: υπολογίζονται τα βάρη με βάση τον τύπο TF-IDF που αναφέρθηκε στο προηγούμενο test, ενώ ως μέτρο ομοιότητας χρησιμοποιείται το cosine similarity.
- Test e: υπολογίζονται τα βάρη ως άνω, αλλά στη συνέχεια πολλαπλασιάζονται με τον συντελεστή:

  $\frac{U}{1 + 0.115 * U}$ (U -ο αριθμός των όρων που είναι μοναδικοί σε ένα αρχείο κειμένου)

  προκειμένου να επιτύχουμε κανονικοποίηση ως προς το μήκος των αρχείων. Ο τύπος αυτός προέρχεται από το pivoted length normalization κατά τους Buckley και Mitra. Ως μέτρο ομοιότητας χρησιμοποιείται η Ευκλείδεια απόσταση.
- Test f: το βάρος υπολογίζεται όπως και στο test e, αλλά ως μέτρο ομοιότητας χρησιμοποιείται το cosine similarity.
- Test g: ως βάρος χρησιμοποιείται το TF-IDF όπως υπολογίζεται στην κλάση StringToWordVector, μαζί με την κανονικοποίηση ως προς το μέσο μήκος των αρχείων κειμένου, όπως περιγράφηκε στην ενότητα 4.1.11. Ως μέτρο ομοιότητας χρησιμοποιήθηκε η Ευκλείδεια απόσταση.

Η διενέργεια των test αυτών έγινε χρησιμοποιώντας κείμενα από τα σύνολα κειμένων που αναφέραμε στην προηγούμενη ενότητα.

Παραθέτουμε ακολούθως τα αποτελέσματα από τα test που έγιναν σε ένα σύνολο 49 αρχείων από 5 κατηγορίες του reuters corpus (cor1).

Αρχικά, κάναμε το συνδυασμό Test 0 – Test ο, για να ελέγξουμε την ακρίβεια του αρχικού Weka χωρίς τις δικές μας επεκτάσεις. Τα αποτελέσματα ήταν:

Purity: 0.307 και F-measure: 0.307

Στον ακόλουθο πίνακα μπορούμε να δούμε τα αποτελέσματα της διενέργειας των υπόλοιπων συνδυασμό test. Σημειώνεται ότι χρησιμοποιήθηκε ως αρχικό seed ο αριθμός 10, ενώ μέσω μιας επαναληπτικής διαδικασίας ο αριθμός αυτός άλλαζε αυξητικά κατά 1 μέχρι να επιτευχθεί ο επιθυμητός αριθμός των κλάσεων, στην περίπτωσή μας 8.

Πίνακας μέτρου ακρίβειας F-measure

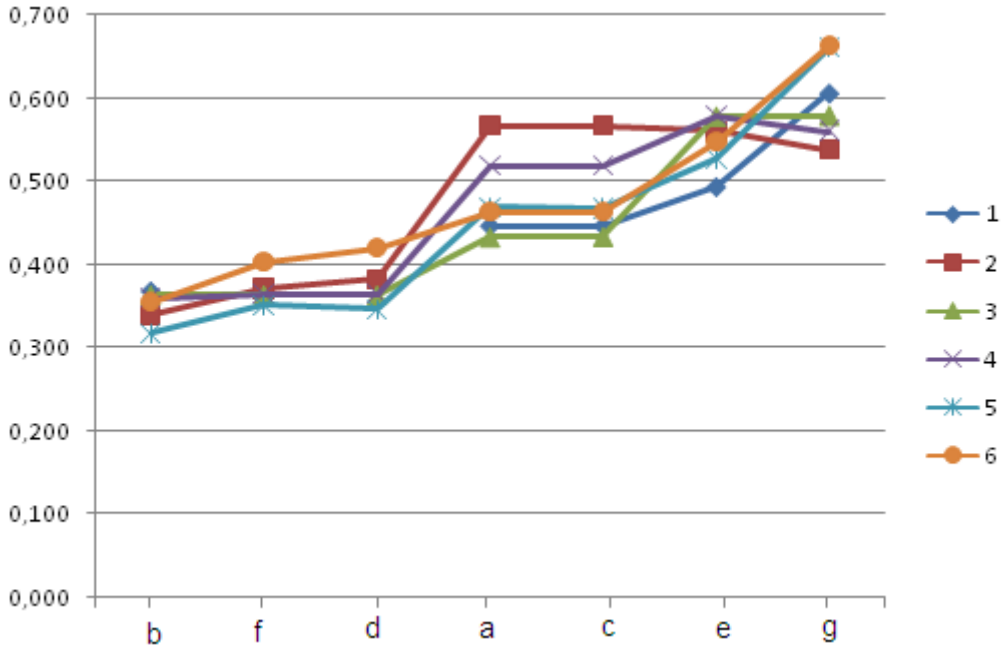
	Test a	Test b	Test c	Test d	Test e	Test f	Test g
Test 1	0.446	0.367	0.446	-	0.493	-	**0.606**
Test 2	**0.566**	0.340	0.566	0.383	0.560	0.372	0.538
Test 3	0.433	0.364	0.434	0.364	0.578	0.364	**0.579**
Test 4	0.519	0.360	0.519	0.365	**0.578**	0.365	0.560
Test 5	0.469	0.318	0.468	0.347	0.528	0.351	**0.661**
Test 6	0.463	0.355	0.463	0.420	0.547	0.403	**0.663**

Πίνακας μέτρου ακρίβειας Purity

	Test a	Test b	Test c	Test d	Test e	Test f	Test g
Test 1	0.469	0.367	0.469	-	0.489	-	**0.612**
Test 2	**0.571**	0.387	0.571	0.388	0.551	0.428	0.531
Test 3	0.408	0.347	0.408	0.347	**0.571**	0.347	0.571
Test 4	0.531	0.347	0.531	0.347	**0.571**	0.347	0.551
Test 5	0.429	0.367	0.429	0.388	0.551	0.388	**0.653**
Test 6	0.428	0.388	0.429	0.429	0.571	0.388	**0.673**

(Σε κάθε γραμμή έχει σημειωθεί με bold η υψηλότερη τιμή μέτρου που παρατηρήθηκε.)

Ας δούμε σύντομα την κατάταξη των τιμών του f-measure στο ακόλουθο γράφημα, στο οποίο ανακατατάξαμε τις στήλες των tests a-f με τη σειρά b, f, d, a, c, e, g :



Εικόνα 5-13 : Διάγραμμα σύγκρισης των τιμών F-measure για τα διάφορα test

Παρατηρούμε λοιπόν ότι οι τιμές του F-measure για τα test 1-6 ακολουθούν αυξητική συμπεριφορά στη διενέργεια των tests με την κατάταξη που φαίνεται στο διάγραμμα. Επιπλέον, μπορούμε να δούμε ότι στις περισσότερες περιπτώσεις η γραμμή που αντιστοιχεί στο test 6 εμφανίζει τις περισσότερες μέγιστες τιμές. Καταλήγουμε λοιπόν στο συμπέρασμα ότι η μεθοδολογία που αντιστοιχεί στο test 6 για την προ-επεξεργασία και στο test g για τον αλγόριθμο ομαδοποίησης εμφανίζει το βέλτιστο αποτέλεσμα.

Η διαδικασία αυτή των tests έγινε πάνω σε διαφορετικά σύνολα αρχείων επιλεγμένα από τα corpora που προαναφέραμε. Στη διάρκεια των test, όπως το προηγούμενο, οι τιμές των μέτρων που παρουσιάσαμε διέφεραν αρκετά. Παρατηρήσαμε αρχικά κάνοντας τα test 1-6 για την προ-επεξεργασία σε συνδυασμό με τα test a-f για τον αλγόριθμο ομαδοποίησης, ότι τις περισσότερες φορές (όχι σε όλες, κάτι το οποίο φαίνεται στο προηγούμενο παράδειγμα) τα μεγαλύτερα μέτρα παρατηρούνταν στο συνδυασμό του test 6 με το test e, δηλαδή στο σύνολο των λειτουργιών που προσθέσαμε στην προ-επεξεργασία αρχείων, με χρήση του υπολογισμού TF-IDF επί το συντελεστή κανονικοποίησης ως προς το μήκος, και την Ευκλείδεια απόσταση ως μέτρο ομοιότητας.

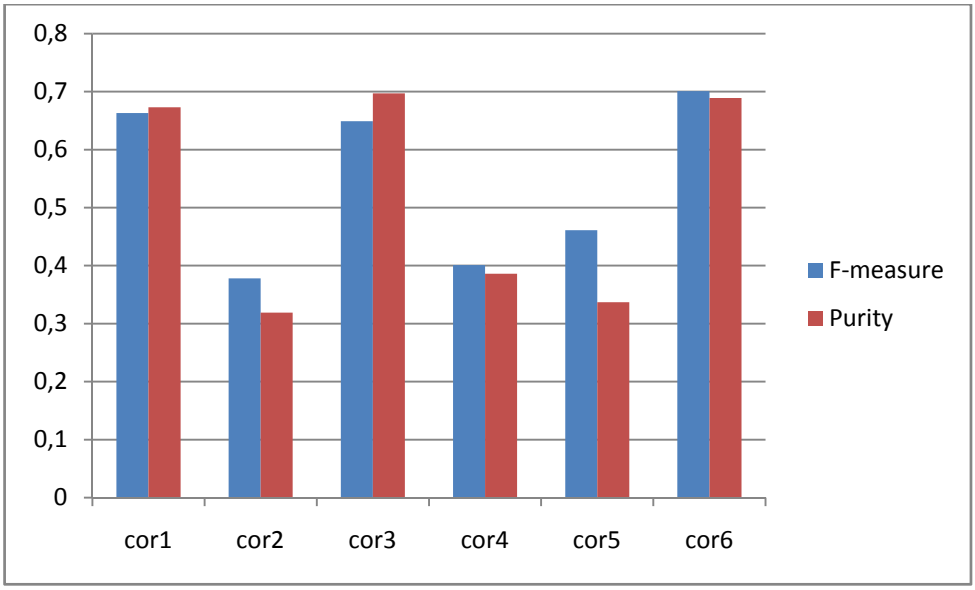
Ωστόσο, υπολογίζοντας τα βάρη με τον υπολογισμό TF-IDF που γίνεται στην κλάση του StringToWordVector, επί το συντελεστή κανονικοποίησης ως προς το μέσο μήκος των αρχείων κειμένου, και την Ευκλείδεια απόσταση ως διαίρεση, τελικά οι υψηλότερες τιμές των μέτρων στις περισσότερες περιπτώσεις ελέγχων παρατηρούνταν στο συνδυασμό test 6-test g.

Επισημαίνουμε εδώ πως το γεγονός ότι τα μέγιστα των μέτρων δεν ήταν πάντοτε υψηλότερα στην ίδια διαδικασία, μπορεί να οφείλεται στη διαφοροποίηση του λεξιλογίου, της θεματολογίας, του μεγέθους κειμένου που των αρχείων που χρησιμοποιήθηκαν κάθε φορά.

Ωστόσο, οι υψηλότερες τιμές παρουσιάζονταν πάντοτε μεταξύ των tests που περιείχαν τις λειτουργίες που προσθέσαμε στο Weka για την αποδοτικότερη εξαγωγή χαρακτηριστικών, κάτι το οποίο μας επιβεβαίωσε την αρχική σκέψη περί βελτίωσης του αποτελέσματος της ομαδοποίησης. Σημειώνουμε δε, ότι η χρήση του κατάλληλου βάρους βρίσκεται υπό αρκετή συζήτηση σε ερευνητικό επίπεδο, καθώς δεν υπάρχει κάποιος προτεινόμενος τύπος που να είναι βέλτιστος σε όλες τις περιπτώσεις.

Στην εργασία αυτή, σύμφωνα με τους ελέγχους που κάναμε, καταλήξαμε να προτείνουμε ως μέθοδο αυτήν που προτείναμε στην ενότητα 4.1, και ουσιαστικά ελέγχεται από τα test 6 με το test g. Παραθέτουμε εδώ τις τιμές των μέτρων που υπολογίσαμε σε ελέγχους με την προτεινόμενη μέθοδο:

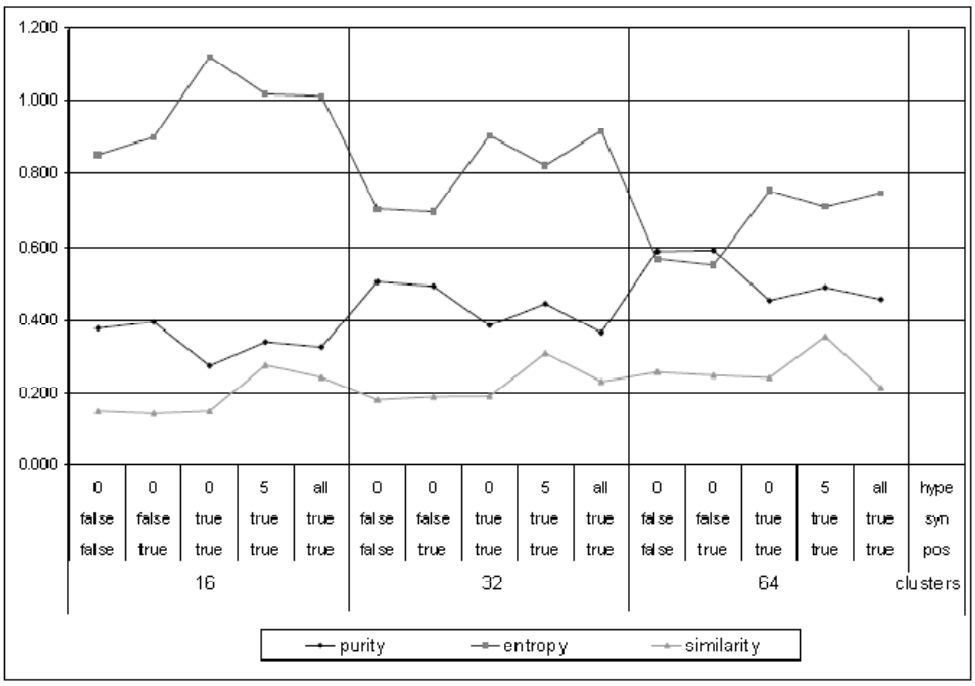
- Έλεγχος με τη χρήση 163 αρχείων από 8 κατηγορίες του reuters corpus (cor2):
  F-measure: 0.378 Purity: 0.319
- Έλεγχος με τη χρήση 132 αρχείων από 6 κατηγορίες του reuters corpus (cor3):
  F-measure: 0.649 Purity: 0.697
- Έλεγχος με τη χρήση 176 αρχείων από 6 κατηγορίες του corpus με τα newsgroups (cor4): F-measure: 0.401 Purity: 0.386
- Έλεγχος με τη χρήση 1855 αρχείων από 47 κατηγορίες του reuters corpus (cor5):
  F-measure: 0.461 Purity: 0.337
- Έλεγχος με τη χρήση 163 αρχείων από 8 κατηγορίες του USE corpus (cor6):
  F-measure: 0.786 Purity: 0.689



Εικόνα 5-14 : Οι τιμές των μέτρων Purity και F-measure για τους ελέγχους στα διάφορα corpora με τη μεθοδολογία που προτείνουμε

Ας δούμε τώρα τα αποτελέσματα από ελέγχους που περιγράφηκαν σε δύο δημοσιεύσεις τις οποίες μελετήσαμε για να συγκρίνουμε με τη μεθοδολογία μας.

Οι Sedding και Kazakov, στη δημοσίευσή τους "WordNet-based Text Document Clustering", χρησιμοποιώντας επίσης το WordNet για την προ-επεξεργασία των αρχείων και τον αλγόριθμο k-means για την ομαδοποίηση, κάνοντας έλεγχο με κείμενα από το Reuters Corpus, κατέληξαν στην ακόλουθη κατανομή αποτελεσμάτων ελέγχου:

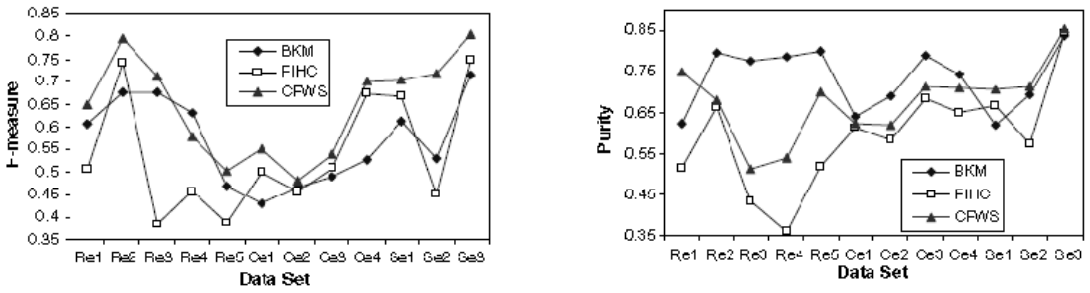


Εικόνα 5-15 : Οι τιμές του Purity στην υλοποίηση των Sedding και Kazakov

Από την κατανομή αυτή ξεχωρίζουμε το γράφημα που αντιστοιχεί στο Purity, και παρατηρούμε ότι οι τιμές του κυμαίνονται από περίπου 0.27 έως 0.6.

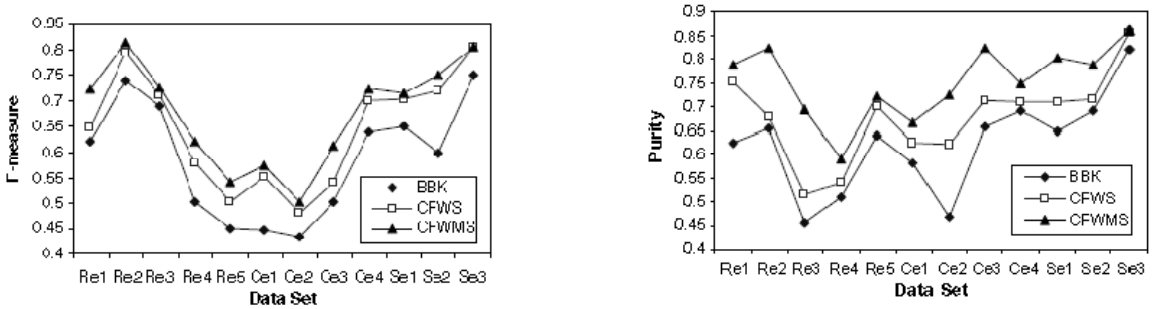
Οι Li, Chung και Holt στη δημοσίευσή τους "Text document clustering based on frequent word meaning sequences", κατασκευάζουν δύο αλγορίθμους, τον Clustering based on Frequent Word Sequences (CFWS) και τον Clustering based on Frequent Word Meaning Sequences, και χρησιμοποιώντας για τον έλεγχο σύνολα κειμένων συμπεριλαμβανομένου του Reuters Corpus, συγκρίνουν τα αποτελέσματά τους με τους αλγορίθμους bisecting k-means (BKM), FIHC, και bisecting k-means με χρήση του αλγορίθμου background knowledge, και καταλήγουν στα ακόλουθα συγκριτικά αποτελέσματα:

Για έλεγχο του CFWS συγκριτικά με τους αλγορίθμους BKM και FIHC, με μέτρα το F-measure και το Purity:



Εικόνα 5-16 : Σύγκριση των αλγορίθμων CFWS, BKM, FIHC

Για έλεγχο του CFWMS συγκριτικά με τους αλγορίθμους CFWS και BBK, με μέτρα το F-measure και το Purity:



Εικόνα 5-17 : Σύγκριση των αλγορίθμων CFWMS, CFWS, BBK

Παρατηρούμε ότι οι τιμές από τους ελέγχους που κάναμε με τη δική μας μεθοδολογία, κυμαίνονται μεταξύ των τιμών ελέγχου με τη χρήση άλλων αλγορίθμων:

Αλγόριθμοι που ελέγχθηκαν με κείμενα από το Reuters Corpus	**Ο αλγόριθμος (k-means) με τη μεθοδολογία των Sedding, Kazakov**	**Ο αλγόριθμος BKM όπως ελέγχθηκε από τους Li, Chung και Holt**	**Ο αλγόριθμος BKK όπως ελέγχθηκε από τους Li, Chung και Holt**	**Ο αλγόριθμος (k-means) με τη μεθοδολογία της διπλωματικής εργασίας**	**Ο αλγόριθμος CFWS των Li, Chung και Holt**	**Ο αλγόριθμος CFWMS των Li, Chung και Holt**
Υψηλότερη τιμή F-measure:	--	0.69	0.74	0.79	0.81	0.82
Χαμηλότερη τιμή F-measure:	--	0.41	0.41	0.38	0.42	0.44
Υψηλότερη τιμή Purity:	0.62	0.82	0.77	0.69	0.83	0.83
Χαμηλότερη τιμή Purity:	0.23	0.49	0.45	0.39	0.61	0.58

Αξίζει να παρατηρήσουμε ότι η σύγκριση της ακρίβειας μεταξύ των μεθόδων που χρησιμοποιούν παραλλαγές του αλγορίθμου ομαδοποίησης K-means, η μεθοδολογία που προτείναμε έφερε τα υψηλότερα αποτελέσματα.

Ωστόσο, παρατηρούμε υψηλότερη ακρίβεια με τη χρήση άλλων αλγορίθμων. Υπάρχει δηλαδή περιθώριο βελτιστοποίησης των μέτρων ακρίβειας, κάτι το οποίο μας ωθεί να στραφούμε στην περαιτέρω εξερεύνηση αλγορίθμων ομαδοποίησης, για τη βελτίωση των αποτελεσμάτων μας.

Βέβαια, οφείλουμε να εξηγήσουμε ότι ακόμη και αν χρησιμοποιήθηκαν αρχεία από το Reuters corpus σε όλες τις ανωτέρω περιπτώσεις, τα αποτελέσματα δεν είναι απόλυτα συγκρίσιμα για το λόγο ότι μιλάμε για διαφορετικό πλήθος και διαφορετική επιλογή κειμένων, κάτι το οποίο διαφοροποιεί αρκετά το αποτέλεσμα της ομαδοποίησης.

5.5 Αξιολόγηση

Σε γενικές γραμμές, τόσο σε ελέγχους που έγιναν σε προ-ομαδοποιημένα σύνολα αρχείων, όσο και σε ομαδοποιήσεις που κάναμε και προσπαθήσαμε να αντιστοιχήσουμε την ομαδοποίηση του συστήματος και πώς εμείς θα ομαδοποιούσαμε νοηματικά τα δοθέντα κείμενα, τα αποτελέσματα ήταν ικανοποιητικού βαθμού.

Ωστόσο, κάποιες παρατηρήσεις που υπάρχουν για την ομαδοποίηση κειμένων σχετίζονται με το χρόνο εκτέλεσης, ο οποίος σε μεγάλο πλήθος αρχείων είναι αρκετά μεγάλος. Θα ήταν λοιπόν χρήσιμο να γίνει κάποια μελέτη σχετικά με εναλλακτικούς αλγόριθμους που θα μπορούσαν να χρησιμοποιηθούν, όπως στη διαδικασία της αποσαφήνισης των εννοιών των λέξεων όπου τώρα χρησιμοποιούμε άπληστο αλγόριθμο. Επιπλέον, κάτι που αυξάνει το χρόνο εκτέλεσης είναι η θεώρηση των συνώνυμων λέξεων, η οποία θα μπορούσε να περιοριστεί στον κανόνα δύο λέξεις να θεωρούνται συνώνυμες εάν ανήκουν στο ίδιο synset μέσα στο WordNet,ενώ τώρα κρατάμε τις συνώνυμες λέξεις από κάθε λέξη και κάθε φορά που εξετάζουμε μια λέξη κάνουμε αναζήτηση μεταξύ των συνωνύμων όλων των λέξεων που έχουν διαβαστεί, για να δούμε εάν υπάρχουν συνώνυμοι όροι και εάν χρειάζεται κάποια συγχώνευση όρων.

Επιπλέον συστήνεται η δοκιμή και άλλων αλγορίθμων ομαδοποίησης πέρα από τον K-means, καθώς είδαμε στη σύγκριση με άλλα παρόμοια συστήματα ότι η χρήση διαφορετικών αλγορίθμων έφερε υψηλότερες τιμές στα μέτρα ακρίβειας.

6 Συμπεράσματα και Προοπτικές

Ας συνοψίσουμε τώρα τη μελέτη που κάναμε στη συγκεκριμένη διπλωματική εργασία.

6.1 Σύνοψη και συμπεράσματα

Μελετήσαμε την ομαδοποίηση των ιδεών σε ένα σύστημα διαχείρισης ιδεών με χρήση information aggregation markets, χρησιμοποιώντας μεθόδους από την εξόρυξη γνώσης από κείμενα καθώς και την επεξεργασία φυσικής γλώσσας.

Η ανάγκη για ομαδοποίηση προέκυψε από την ανάγκη που υπήρχε στο σύστημα διαχείρισης ιδεών (IDeM) για την ανάπτυξη ενός εργαλείου με το οποίο ο administrator μιας εικονικής αγοράς θα μπορούσε να μελετήσει με μεγαλύτερη άνεση την πληθώρα ιδεών που έχουν υποβληθεί από τους χρήστες στο IDeM.

Έτσι, αναπτύχθηκε ένα εργαλείο ομαδοποίησης αρχείων κειμένου το οποίο ενσωματώθηκε στο σύστημα IDeM για την ομαδοποίηση των ιδεών, εφόσον αυτές έχουν τη μορφή κειμένου.

Το σύστημα της ομαδοποίησης αρχείων κειμένου αναπτύχθηκε πάνω στον κώδικα του Weka, ενός εργαλείου εξόρυξης δεδομένων. Για την αναπαράσταση των αρχείων κειμένου χρησιμοποιήθηκε το μοντέλο του διανυσματικού χώρου. Η διαδικασία της ομαδοποίησης ξεκινάει από την προ-επεξεργασία των αρχείων κειμένου προκειμένου να γίνει εξαγωγή γνωρισμάτων, δηλαδή των σημαντικότερων όρων που παίζουν ρόλο στην ομαδοποίηση, συνεχίζει με την εκτέλεση του αλγορίθμου ομαδοποίησης (επιλέχθηκε ο αλγόριθμος k-means) και ολοκληρώνεται με την εξαγωγή των αποτελεσμάτων ομαδοποίησης.

Στη διαδικασία της προ-επεξεργασίας αρχείων προστέθηκαν μέθοδοι της επεξεργασίας φυσικής γλώσσας για τη βελτίωση της επιλογής των χαρακτηριστικών. Έτσι, προστέθηκαν οι

διαδικασίες αφαίρεσης stop-words, stemming, εύρεσης συνωνύμων με χρήση του ηλεκτρονικού λεξικού WordNet, part-of-speech tagging και αποσαφήνιση της έννοιας των λέξεων, στάθμισης των όρων και pruning των πολύ σπάνιων όρων.

Το τελικό σύστημα της ομαδοποίησης μπορεί είτε να λειτουργήσει αυτόνομα είτε να ενσωματωθεί σε κάποια εφαρμογή, όπως στην περίπτωσή μας που το ενσωματώσαμε στο IDeM.

Παρατηρώντας την εκτέλεση των ελέγχων του συστήματος, όπως περιγράφηκαν στο προηγούμενο κεφάλαιο, σε γενικές γραμμές τα αποτελέσματα της ομαδοποίησης ήταν ικανοποιητικά. Η έκβαση του αποτελέσματος βέβαια εξαρτάται πάντοτε από το είδος του περιεχομένου των κειμένων (αν για παράδειγμα έχουμε να κάνουμε με κάποια εξειδικευμένη κατηγορία κειμένων), το μήκος τους και τη νοηματική ομοιότητα που υπάρχει στην πραγματικότητα μεταξύ τους.

6.2 Μελλοντικές επεκτάσεις

Συμπερασματικά, μπορούμε να δούμε ότι η ανάπτυξη του εργαλείου της ομαδοποίησης αποτέλεσε μια επιπλέον λειτουργική αξία στο IdeM. Σε μια μεγάλη εταιρεία υπάρχουν πολλές ιδέες, η αξιολόγηση των οποίων αποτελεί ένα πολύ σημαντικό κομμάτι στον τομέα της διαχείρισης καινοτομίας. Ήταν λοιπόν αναγκαίο να βρεθεί ένα τρόπος να διαχωρίζονται οι ιδέες ως προς το περιεχόμενό τους, ώστε να μπορούν να μελετηθούν και τελικά να αξιολογηθούν ευκολότερα και πιο αποτελεσματικά.

Μια πρώτη επέκταση που προτείνουμε είναι το αμέσως επόμενο στάδιο, η αναγνώριση των κοινών όρων που καθορίζουν το νοηματικό περιεχόμενο κάθε ομάδας ώστε να βγουν κάποιες νοηματικές κατηγορίες.

Επιπλέον, η οπτικοποίηση (visualization) των αποτελεσμάτων της ομαδοποίησης θα βοηθούσε ακόμη περισσότερο το χρήστη να δει τις ομάδες και να μελετήσει τις ιδέες, καθώς και να δει ποιες ομάδες βρίσκονται μεταξύ τους πιο κοντά νοηματικά και να βγάλει επιπλέον συμπεράσματα.

Τέλος, το σύστημα της ομαδοποίησης κειμένου μπορεί να χρησιμοποιηθεί από το IDeM και για την ομαδοποίηση των προφίλ των χρηστών με βάση τα βιογραφικά τους, τα ενδιαφέροντά τους αλλά και τις ιδέες που έχουν καταθέσει στο παρελθόν, ώστε ο administrator να επιλέγει αποδοτικότερα ποιους χρήστες επιθυμεί να διαλέξει να παίξουν σε κάθε αγορά, ενώ ενδεχομένως μια τέτοια ομαδοποίηση να ενδιέφερε και τον υπεύθυνο ανθρωπίνων πόρων της εταιρείας που χρησιμοποιεί το IDeM για τη διαχείριση καινοτομίας.

7 *Βιβλιογραφία*

[AS07] Apostolou Dimitris, Bertrand Sereno, D5.2.1 Electronic Prediction Market V.1
Revision, Version: V1.0 Final, ICCS - Institute of Communications and Computer Systems, August 2007

[BAM07] Efthimios Bothos, Dimitris Apostolou, Gregoris Mentzas, Collective Intelligence for Idea Management with Internet-based Information Aggregation Markets, ICCS NTUA, 2007

[BAM08] Efthimios Bothos, Dimitris Apostolou and Gregoris Mentzas, A Collaborative Information Aggregation System for Idea Management,The Third International Conference on Internet and Web Applications and Services, 2008

[BAM08] Efthimios Bothos, Dimitris Apostolou, Gregoris Mentzas, Idea Selection and Information Aggregation Markets, IEMC 08

[BER03] Michael W. Berry, Survey of Text Mining: Clustering, Classification, and Retrieval, Springer 2003

[BU05] Pushpak Bhattacharyya, Narayan Unny, Word Sense Disambiguation and Text Similarity Measurement Using WordNet, Indian Institute of Technology, 2005

[CHA03] Soumen Chakrabarti, Mining the Web "Discovering Knowledge from Hypertext Data" , The Morgan Kaufmann Series in Data Management Systems, 2003

[DS05] Thanh Ngoc Dao, Troy Simpson, Measuring Similarity between sentences 2005

[FRA05] Κωνσταντίνος Φράγγος, Στατιστικοί Έλεγχοι στην Επεξεργασία Φυσικής Γλώσσας Μέσω Η/Υ, Διδακτορική Διατριβή ΕΜΠ, 2005

[GAR06] E. Garcia Cosine Similarity and Term Weight Tutorial, An Information Retrieval Tutorial on Cosine Similarity Measures, Dot Products and Term Weight Calculations, http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html, 2006

[IV98] Nancy Ide,Jean Véronis,Word Sense Disambiguation:The State of the Art, Computational Linguistics, 1998

[LCH07] Yanjun Li, Soon M.Chung, Jon D. Holt, Text document clustering based on frequent word meaning sequences, Science Direct, August 2007

[LSM95] Xiaobin Li, Stan Szpakowicz, Stan Matwin, A WordNet based Algorithm for Word Sense Disambiguation , 1995

[MAL05] Διπλωματική Εργασία Πρόδρομος Μαλακασιώτης, Αναγνώριση μερών του λόγου σε ελληνικά κείμενα με τεχνικές ενεργητικής μάθησης , Μεταπτυχιακού Διπλώματος Ειδίκευσης ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ, Ιούνιος 2005

[MM08] Γιάνης Μαΐστρος - Στέλλα Μαρκαντωνάτου, Παράσταση και Επεξεργασία Φυσικής Γλώσσας , Εργαστήριο Επεξεργασίας Φυσικής Γλώσσας ΕΜΠ, http://glotta.ntua.gr/nlp/courses/nlp-textbook.htm, 2008

[NAS06] Ιωάννης Νασίκας, Text Mining: Μια νέα προτεινόμενη μέθοδος με χρήση κανόνων συσχέτισης, Διπλωματική εργασία στο Διατμηματιό Μεταπτυχιακό Πρόγραμμα Ειδίκευσης "Μαθηματικά των Υπολογιστών και των Αποφάσεων, Πανεπιστήμιο Πατρών. Ιούνιος 2006

[NIN05] Wei Ning, Textmining and Organization in Large Corpus, Kongens Lyngby 2005

[PAP05] Θωμά Α Παπαστεργίου, Μέτρα ομοιότητας στην τεχνική ομαδοποίησης (Clustering): Εφαρμογή σε ανάλυση κειμένων (Text Mining),Διπλωματική εργασία στο Διατμηματιό Μεταπτυχιακό Πρόγραμμα Ειδίκευσης "Μαθηματικά των Υπολογιστών και των Αποφάσεων, Πανεπιστήμιο Πατρών, 2005

[PAV03] Αθανάσιος Ν. Παυλόπουλος, Text Mining: Ανασκόπηση και εφαρμογή στην περίληψη κειμένου , Τμήμα Πληροφορικής Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης , 2003

[PN08] Ευαγγελία Πιτουρά, Μυρτώ Ντέτσικα, Σημεώσεις μαθήματος Εξόρυξης Δεδομένων, http://www.cs.uoi.gr/~pitoura/courses/dm/, 2008

[POR80] M.F.Porter, An algorithm for suffix stripping Originally published in \Program\, \14\ no. 3, pp 130-137, July 1980

[SBM96] Amit Singhal , Chris Buckley, Mandar Mitra, Pivoted Document Length Normalization, Department of Computer Science Cornell University Ithaca, 1996

[SD05] Troy Simpson , Thanh Dao, WordNet-based semantic similarity measurement, http://www.codeproject.com/KB/string/semanticsimilaritywordnet.aspx ,October 2005

[SK04] Julian Sedding, Dimitar Kazakov, WordNet-based Text Document Clustering, 2004

[SWY75] G. Salton, A. Wong, C.S. Yang, A Vector Space Model for Automatic Indexing, Cornell University 1975

[TEK06] Teknomo, Kardi. K-Means Clustering Tutorials. http:\\people.revoledu.com\kardi\ tutorial\kMean\ , 2006

[TKM+03] Kristina Toutanova, Dan Klein, Christopher D. Manning, Yoram Singer, Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network, HLT-NAACL 2003

[TKM03] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[TM00] Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-

2000), pp. 63-70.

[TM00] Kristina Toutanova, Christopher D. Manning, Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger, 2000

[ΤΤ07] Γεώργιος Τζιραλής, Ηλίας Τατσιόπουλος, Οι Προγνωστικές Αγορές Ως Εργαλείο Υποστίρηξης Αποφάσεων στη Διοίκηση της Εφοδιαστικής Αλυσίδας, Β' Πανελλήνιο Συνέδριο Μηχανολόγων – Ηλεκτρολόγων, Μάιος 2007

[VAR03] Ηρακλής Γ. Βαρλάμης, Σημασιολογικός Χαρακτηρισμός, Οργάνωση και Διαχείριση Περιεχομένου του Παγκόσμιου Ιστού, με Χρήση Οντολογίων και Έμφαση στο Ρόλο των Υπερσυνδέσμων, Διδακτορική Διατριβή Οικονομικό Πανεπιστήμιο Αθηνών, Σεπτέμβριος 2003

[WEK08] Weka Documentation, http://www.cs.waikato.ac.nz/~ml/weka/ , 2008

Παράρτημα Α: Η λίστα Stopwords

Παραθέτουμε εδώ τη λίστα με τα stopwords που χρησιμοποιήσαμε για την αφαίρεση stopwords στο στάδιο της προ-επεξεργασίας των κειμένων.

Stopwords:

a	anybody	before	com
a's	anyhow	beforehand	come
able	anyone	behind	comes
about	anything	being	concerning
above	anyway	believe	consequently
according	anyways	below	consider
accordingly	anywhere	beside	considering
across	apart	besides	contain
actually	appear	best	containing
after	appreciate	better	contains
afterwards	appropriate	between	corresponding
again	are	beyond	could
against	aren't	both	couldn't
ain't	around	brief	course
all	as	but	currently
allow	aside	by	d
allows	ask	c	definitely
almost	asking	c'mon	described
alone	associated	c's	despite
along	at	came	did
already	available	can	didn't
also	away	can't	different
although	awfully	cannot	do
always	b	cant	does
am	be	cause	doesn't
among	became	causes	doing
amongst	because	certain	don't
an	become	certainly	done
and	becomes	changes	down
another	becoming	clearly	downwards
any	been	co	during

e
each
edu
eg
eight
either
else
elsewhere
enough
entirely
especially
et
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
f
far
few
fifth
first
five
followed
following
follows
for
former
formerly
forth
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had
hadn't
happens
hardly
has
hasn't
have
haven't
having
he
he's
hello
help
hence
her
here
here's
hereafter
hereby
herein
hereupon
hers
herself
hi
him
himself
his
hither
hopefully
how
howbeit
however
i
i'd
i'll
i'm
i've
ie
if
ignored
immediate
in
inasmuch
inc
indeed
indicate
indicated
indicates
inner
insofar
instead
into
inward
is
isn't
it
it'd
it'll
it's
its
itself
j
just
k
keep
keeps
kept
know
knows
known
l
last
lately
later
latter
latterly
least
less
lest
let
let's
like
liked
likely
little
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
n
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of

off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn't
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
t's
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that's
thats
the
their
theirs
them
themselves
then
thence
there
there's
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they'd
they'll
they're
they've
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was

wasn't
way
we
we'd
we'll
we're
we've
welcome
well
went
were
weren't
what
what's
whatever
when
whence
whenever
where
where's
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who's
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won't
wonder
would
would
wouldn't
x
y
yes
yet
you
you'd
you'll
you're
you've
your
yours
yourself
yourselves
z
zero