ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Σημασιολογική Συσταδοποίηση Αντικειμένων Με Χρήση Οντολογικών Περιγραφών.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΑΝΟΥ ΧΑΤΖΗΘΕΟΔΩΡΟΥ

Επιβλέπων : Γιώργος Στάμου
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2015

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Σημασιολογική Συσταδοποίηση Αντικειμένων Με Χρήση Οντολογικών Περιγραφών.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΑΝΟΥ ΧΑΤΖΗΘΕΟΔΩΡΟΥ

Επιβλέπων : Γιώργος Στάμου
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Ιουλίου 2015.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γιώργος Στάμου	Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης
Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2015

(Υπογραφή)

.....

ΜΑΝΟΣ ΧΑΤΖΗΘΕΟΔΩΡΟΥ

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η μελέτη των τρόπων με τους οποίους μια οντολογία μπορεί να χρησιμοποιηθεί ως πηγή χαρακτηριστικών για τη συσταδοποίηση ενός συνόλου δεδομένων. Τα δεδομένα αυτά συγκεκριμένα αφορούσαν το σενάριο μιας ταινίας. Η συσταδοποίηση έγινε για τα πλάνα της ταινίας με τα χρήση διαφορετικών τεχνικών, από τις οποίες επιλέχθηκε η αποτελεσματικότερη για την παραγωγή του τελικού αποτελέσματος.

Συγκεκριμένα, τα δεδομένα θεωρήθηκαν ότι είναι εν γένει ημιδομημένα, δηλαδή ότι είναι εν μέρει κατηγοριοποιημένα βάσει κάποιας οντολογίας, αλλά ότι μπορεί και να περιέχουν και κάποια επιπλέον, αδόμητη κειμενική πληροφορία που να αποτελεί δυνητική πηγή επιπλέον χαρακτηριστικών. Αρχικά υπολογίστηκε ο κατάλληλος χώρος αναπαράστασης των χαρακτηριστικών που προέρχονται από τις δύο διαφορετικές πηγές. Για την παραγωγή των χαρακτηριστικών που αφορούν την οντολογία έγινε χρήση τεχνικών συλλογιστικής, καθώς κάποια από αυτά ήταν ρητώς δηλωμένα, άλλα όμως αποτελούσαν υπονοούμενη γνώση και δεν ήταν άμεσα διαθέσιμα. Για την παραγωγή των χαρακτηριστικών που βρίσκονταν σε μορφή αδόμητης πληροφορίας, έγινε λημματοποίηση και αφαίρεση αδιάφορων λέξεων στα κειμενικά τμήματα του σεναρίου.

Με βάση τον χώρο των παραπάνω χαρακτηριστικών, πραγματοποιήθηκε η συσταδοποίηση με τον αλγόριθμο k-means. Τα αποτελέσματα οργανώθηκαν σε βάση δεδομένων και παρουσιάσθηκαν μέσω ιστοσελίδας.

Η μελέτη αυτή μπορεί να επεκταθεί εύκολα και σε άλλα δεδομένα, με μικρές μετατροπές, ανάλογα με τη μορφή αναπαράστασης των δεδομένων και της οντολογίας, καθώς η διαδικασία έχει καταγραφεί αναλυτικά και είναι πλήρως παραμετροποιημένη.

Λέξεις Κλειδιά: <<Οντολογία, Σημασιολογικός Ιστός, Συλλογιστική, RDF, SPARQL, Εξαγωγή χαρακτηριστικών, Λημματοποίηση, Συσταδοποίηση, K-means, Χάρτης Αυτό-Οργάνωσης, Ιεραρχική Συσταδοποίηση >>

Abstract

The scope of this thesis was the study of the ways in which an ontology can be used as the source of features for the clustering of a data set. This data set was, specifically, the script of a movie. The clustering was executed for the shots of the movie using different techniques. The final results were produced using the most efficient of those techniques.

Specifically, the data were considered to be, as a whole, semi-structured. That is, they are partially classified based on an ontology, but they can also contain some extra, unstructured textual information that constitutes a potential source of additional features. Initially, we calculated the appropriate representation space for the features that originate from both sources. For the extraction of the features that pertain to the ontology we used reasoning techniques, since some of them were explicitly stated, but others were implied knowledge and weren't directly available. For the extraction of the features contained in the unstructured information, we used lemmatization and stop words removal techniques on the textual parts of the script.

Based on the space of the aforementioned features, we did the clustering using the k-means algorithm. The results were organized in a database and were presented through a website.

This study can be easily expanded to include different data sets, with few modifications, depending on the representation format of the data and the ontology, since the procedure is thoroughly specified and fully parametrized.

Keywords: <<Ontology, Semantic Web, Reasoning, RDF, SPARQL, Feature extraction, Lemmatization, Clustering, K-means, Self-Organizing Map, Hierarchical Clustering>>

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή κ. Γιώργο Στάμου που είχε την επίβλεψη της συγκεκριμένης διπλωματικής εργασίας για την εμπιστοσύνη του και την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον και σύγχρονο αντικείμενο.

Επίσης, ένα μεγάλο ευχαριστώ στον διδάκτορα κ. Αλέξανδρο Χορταρά που είχε την επίβλεψη του θέματος, για την αμέριστη βοήθεια του και το ενδιαφέρον του καθ' όλη τη διάρκεια της παρούσας μελέτης. Από την πρώτη στιγμή ήταν δίπλα μου και πάντα πρόθυμος να με καθοδηγήσει και να με βοηθήσει κάθε φορά που αντιμετώπιζα κάποια δυσκολία, θυσιάζοντας πολύτιμο προσωπικό χρόνο.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου, τη μητέρα μου και τον αδερφό μου, οι οποίοι ήταν πάντα δίπλα μου στις δύσκολες στιγμές, προκειμένου να με ενθαρρύνουν και να με στηρίζουν.

Πίνακας περιεχομένων

1

Εισαγωγή

Ο Σημασιολογικός Ιστός (Web 3.0) θεωρείται το μέλλον του διαδικτύου. Βασίζεται σε τεχνολογίες που ήδη υπάρχουν αλλά και σε νέες τεχνολογίες οι οποίες αναπτύσσονται με τη βοήθεια της κοινότητας. Δεδομένου ότι σκοπεύει να είναι μια μεγάλη βάση όπου τα δεδομένα από διαφορετικά πεδία θα συνδέονται μεταξύ τους, αναμένεται να παίξει μεγάλο ρόλο στη ζωή μας τα επόμενα χρόνια. Μερικά από τα πεδία στα οποία αναμένεται να έχει μεγαλύτερη επίδραση είναι στην υγεία, στην παιδεία και στις επιχειρήσεις. Υπάρχουν ήδη πολλές προσπάθειες από εταιρίες, ερευνητές και μη κερδοσκοπικές οργανώσεις για να παραγάγουν πρότυπα οντολογιών, κυρίως για τα παραπάνω πεδία, για να υπάρχουν κοινές γλώσσες και περισσότερα δεδομένα τα οποία μπορούν να συνδυαστούν για καλύτερα αποτελέσματα.

Για τη συσταδοποίηση ενός συνόλου δεδομένων απαιτείται συνήθως ο υπολογισμός ενός συνόλου χαρακτηριστικών που να περιγράφουν με ικανοποιητικό τρόπο καθένα από τα δεδομένα. Τα είδη των χαρακτηριστικών εξαρτώνται ασφαλώς από το είδος των δεδομένων και από την επιδιωκόμενη συσταδοποίηση. Θα παράδειγμα, για απλά κειμενικά δεδομένα ως χαρακτηριστικά χρησιμοποιούνται συνήθως, ύστερα από κατάλληλη προεπεξεργασία, οι λέξεις που εμφανίζονται στα κείμενα. Στην περίπτωση όμως που τα δεδομένα είναι σημασιολογικά περιγεγραμμένα, δηλαδή συμμετέχουν ως στιγμιότυπα στη γνώση που αναπαρίσταται από κάποια οντολογία, το σύνολο των χαρακτηριστικών μπορεί να διευρυνθεί ώστε να συμπεριληφθεί η πληροφορία που προέρχεται από την οντολογία. Δεδομένου ότι οι έννοιες της οντολογίας στις οποίες συμμετέχει ένα αντικείμενο αποτελούν κατ' εξοχήν φορμαλιστική περιγραφή του αντικειμένου, η χρήση αυτού του είδους των χαρακτηριστικών αναμένεται να αποδειχθεί ιδιαίτερα χρήσιμη κατά τη διαδικασία της συσταδοποίησης.

Στο **Κεφάλαιο 2** παρουσιάζεται το θεωρητικό υπόβαθρο όλων των τεχνικών και αλγορίθμων που χρησιμοποιήθηκαν σε αυτή την εργασία. Στο **Κεφάλαιο 3** αναλύεται η υλοποίηση της εφαρμογής συσταδοποίησης σημασιολογικών δεδομένων. Στο **Κεφάλαιο 4** παρουσιάζεται μια απεικόνιση των αποτελεσμάτων της εργασίας, όπως θα τα έβλεπε ο χρήστης μέσω της ιστοσελίδας που δημιουργήθηκε για το σκοπό αυτό. Στο **Κεφάλαιο 5** τέλος, γίνεται μια σύγκριση της διαδικασίας που ακολουθήθηκε σε αυτή την εργασία έναντι μια ανάλυσης που αγνοεί τα οντολογικά δεδομένα.

2

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτό το κεφάλαιο αναλύεται το θεωρητικό υπόβαθρο όλων των τεχνικών που χρησιμοποιούνται στην εργασία αυτή. Η κατανόηση τους από τον αναγνώστη είναι αναγκαία πριν την παρουσίαση της υλοποίησης και των αποτελεσμάτων της εφαρμογής.

2.1 Σημασιολογικός Ιστός (Semantic Web)

Ο σημασιολογικός Ιστός είναι μια επέκταση του σημερινού Ιστού, που θα φέρει δομή στο ουσιαστικό περιεχόμενο των ιστοσελίδων. Η λογική πίσω από αυτό είναι ότι η δημοσιευμένη πληροφορία θα περιέχει μετα-δεδομένα που θα είναι κοινά για όλους, θα μπορούν να γίνονται κατανοητά και από μηχανές, οι οποίες θα βοηθήσουν στην καλύτερη συλλογή και επεξεργασία τους.

Ο Σημασιολογικός Ιστός είναι εμπνευσμένος από ένα όραμα για το σημερινό Διαδίκτυο το οποίο βρίσκεται στο προσκήνιο από την σύλληψη του Διαδικτύου. Ο Tim Berners-Lee είναι ο πρώτος που οραματίστηκε το Διαδίκτυο να περιέχει πλούσιες περιγραφές των εγγράφων και συνδέσμους μεταξύ τους. Ωστόσο, σε μία προσπάθεια να παρέχει ένα απλό, εύχρηστο και εύρωστο λειτουργικό σύστημα που μπορεί να χρησιμοποιηθεί από όλους, αυτές οι ιδέες παραμερίστηκαν και οι πιο απλές τέθηκαν σε εφαρμογή για να κυοφορήσουν το σημερινό Διαδίκτυο.

Το μεγαλύτερο όραμα εκφράστηκε σε ένα άρθρο γραμμένο από τους Tim Berners-Lee, Jim Hendler και Ora Lassila στο περιοδικό Scientific American το 2001. Στο άρθρο αυτό παρέχουν το όραμα ενός κόσμου που αντί οι άνθρωποι να ερευνούν και να χάνονται μέσα στην πληροφορία ή να διαπραγματεύονται μεταξύ τους άμεσα για να εκτελέσουν έργα ρουτίνας όπως προγραμματισμό ραντεβού, εύρεση εγγράφων κ.λπ., το Διαδίκτυο θα μπορεί να το κάνει αυτό για αυτούς. Αυτό μπορεί να γίνει παρέχοντας επαρκείς πληροφορίες σχετικά με τους πόρους στο Διαδίκτυο καθώς και εργαλεία για τη χρησιμοποίηση των πληροφοριών αυτών έτσι ώστε οι εφαρμογές να μπορούν να βρουν τα σωστά πράγματα και να πάρουν τις σωστές αποφάσεις. Όπως λέει συγκεκριμένα το άρθρο: « Ο Σημασιολογικός Ιστός θα φέρει δομή στο περιεχόμενο των ιστοσελίδων, δημιουργώντας ένα περιβάλλον όπου οι πράκτορες-εφαρμογές θα μπορούν να μεταφέρονται από σελίδα σε σελίδα και να εκτελούν εκλεπτυσμένα έργα για τους χρήστες.». Το ίδιο άρθρο γράφει για το Σημασιολογικό Ιστό ότι : « είναι μία επέκταση του τωρινού Διαδικτύου στην οποία οι πληροφορίες έχουν καλώς ορισμένη σημασία, επιτρέποντας στους υπολογιστές και στους ανθρώπους να συνεργάζονται».

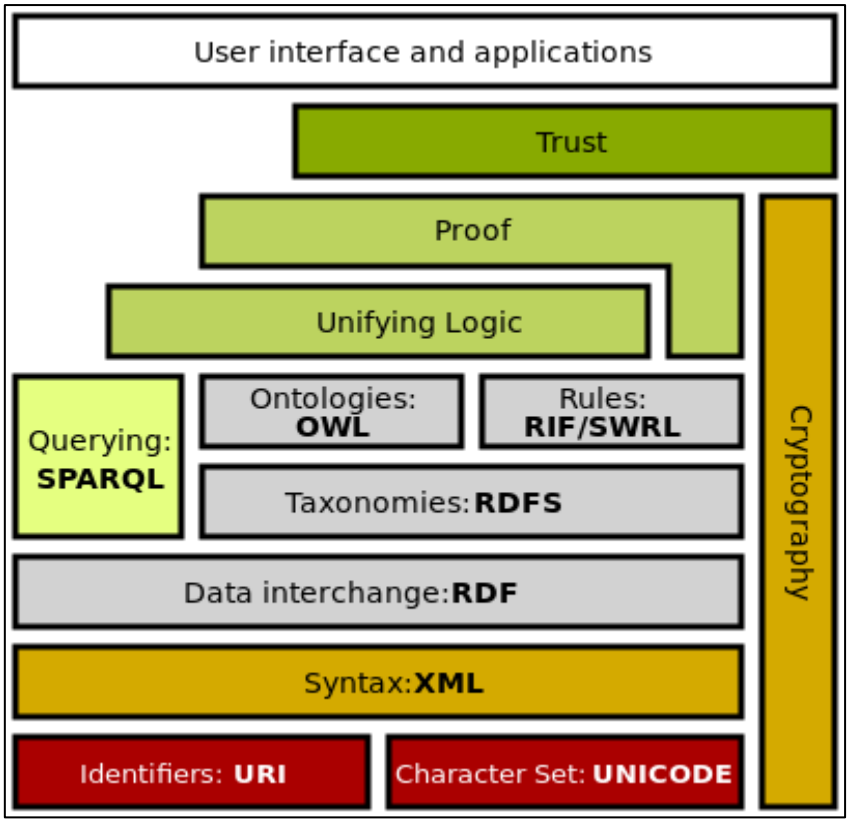
Ο Σημασιολογικός Ιστός βασίζεται σε τεχνολογίες που ήδη υπάρχουν (URI και XML) αλλά και σε νέες τεχνολογίες (RDF, RDFS, OWL, κα.), οι οποίες αναπτύσσονται με την βοήθεια της κοινότητας. Δεδομένου ότι ο νέος Ιστός σκοπεύει να είναι μια μεγάλη βάση όπου δεδομένα από διαφορετικά πεδία θα συνδέονται μεταξύ τους, αναμένεται να παίξει μεγάλο ρόλο στη ζωή μας.

Κάποιες από τις προκλήσεις που έχει να αντιμετωπίσει το Σημασιολογικό Διαδίκτυο είναι η απεραντοσύνη, η ασάφεια, η αβεβαιότητα, η ασυνέπεια και η εξαπάτηση. Αυτόματα συστήματα λογικής θα πρέπει να αντιμετωπίσουν όλα αυτά τα προβλήματα έτσι ώστε να μπορέσουν να ανταπεξέλθουν σε όλα αυτά που υπόσχεται το Σημασιολογικό Διαδίκτυο.

- Απεραντοσύνη: Το παγκόσμιο διαδίκτυο περιέχει πάρα πολλές δισεκατομμύρια σελίδες. Ο ιατρικός όρος οντολογία SNOMED CT περιέχει από μόνος του 370.000 ονόματα κλάσεων και η υπάρχουσα τεχνολογία δεν έχει καταφέρει ακόμα να μειώσει όλους τους σημασιολογικούς διπλούς όρους. Οποιοδήποτε λογικό σύστημα θα πρέπει να έρθει αντιμέτωπο με έναν πραγματικά τεράστιο αριθμό από δεδομένα εισαγωγής.
- Ασάφεια: Όροι όπως "νέος" ή "ψηλός" είναι ασαφής. Αυτό προέρχεται από την ασάφεια των ερωτημάτων του χρήστη, από έννοιες που εμφανίζονται από παρόχους περιεχομένων, από ταίριασμα των όρων των ερωτημάτων με τους όρους των παρόχων και από την προσπάθεια συνδυασμού διαφορετικών βάσεων γνώσης με επικαλυπτόμενες αλλά διακριτά διαφορετικές έννοιες. Η ασαφής λογική είναι η πιο κοινή τεχνική με την οποία αντιμετωπίζουμε την ασάφεια.
- Αβεβαιότητα: Τέτοιοι όροι είναι ακριβής με αβέβαιες τιμές. Για παράδειγμα, ένας ασθενής μπορεί να εμφανίσει ένα σύνολο από συμπτώματα τα οποία ανταποκρίνονται σε έναν αριθμό από διαφορετικές διαγνώσεις όπου η κάθε μία έχει διαφορετική πιθανότητα να είναι αληθής. Οι πιθανοτικές λογικές τεχνικές εφαρμόζονται γενικότερα για να διευθετήσουν την αβεβαιότητα.
- Ασυνέπεια: Αυτές είναι λογικές αντιφάσεις οι οποίες εμφανίζονται εντελή κατά την διάρκεια ανάπτυξης μεγάλων οντολογιών ή και όταν οντολογίες από διαφορετικές πηγές ενώνονται. Ο παραγωγικός συλλογισμός αποτυγχάνει καταστροφικά όταν αντιμετωπίζει την ασυνέπεια.
- Εξαπάτηση: Αυτό συμβαίνει όταν ο δημιουργός της πληροφορίας εσκεμμένα παραπληροφορεί τον καταναλωτή για την πληροφορία. Οι κρυπτογραφικές τεχνικές είναι αυτές που χρησιμοποιούνται για να μειώσουν αυτήν την απειλή.

Η λίστα με τις προκλήσεις είναι περισσότερο ενδεικτική παρά εξαντλητική και εστιάζει στις προκλήσεις για τα στάδια της "ενωτικής λογικής" και της "απόδειξης" του Σημασιολογικού διαδικτύου. Το World Wide Web Consortium (W3C) Incubator Group for Uncertainty Reasoning for the World Wide Web (URW3-XG) στην τελική αναφορά του ομαδοποιεί αυτά τα προβλήματα στην κατηγορία της "αβεβαιότητας".

Η αρχιτεκτονική του Σημασιολογικού Ιστού αποτελείται από μια στοίβα πρωτοκόλλων, όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 1. Αρχιτεκτονική Σημασιολογικού Ιστού

Αυτά τα πρωτόκολλα μπορούν να περιγραφούν ως εξής:

- Η XML (Extensible Markup Language) παρέχει μια στοιχειώδη σύνταξη για τη δομή του περιεχομένου σε αρχεία, χωρίς να συσχετίζει κάποια σημασιολογία με το περιεχόμενο αυτό. Η XML δεν είναι προς το παρόν αναγκαίο συστατικό των τεχνολογιών του Σημασιολογικού Ιστού, καθώς υπάρχουν εναλλακτικοί τρόποι σύνταξης, όπως η RDF Turtle.
- Η XML Schema είναι μια γλώσσα για την παροχή και τους περιορισμούς της δομής και του περιεχομένου των στοιχείων μέσα στα XML αρχεία.
- Η RDF είναι μια απλή γλώσσα που εκφράζει μοντέλα δεδομένων, που αναφέρονται σε αντικείμενα και τις μεταξύ τους σχέσεις. Έχει πολλές μορφές σύνταξης, όπως RDF/XML, N3, Turtle, και είναι ένα θεμελιώδες πρότυπο του Σημασιολογικού Ιστού.
- Η RDF Schema επεκτείνει το RDF και είναι ένα λεξιλόγιο για την περιγραφή κλάσεων και ιδιοτήτων των RDF resources, με σημασιολογία για την ιεραρχία αυτών των κλάσεων και ιδιοτήτων.
- Η OWL προσθέτει επιπλέον λεξιλόγιο για την περιγραφή κλάσεων και ιδιοτήτων.
- Η SPARQL είναι μια γλώσσα ερωτήσεων για βάσεις δεδομένων του Σημασιολογικού Ιστού.
- Η RIF (Rule Interchange Format) είναι μια XML γλώσσα για τη διατύπωση κανόνων του Ιστού που μπορούν να εκτελέσουν οι μηχανές. Παρέχει πολλές εκδόσεις, που καλούνται διάλεκτοι.

- Τα στρώματα της ενοποιημένης λογικής και της απόδειξης βοηθούν στην εξαγωγή αληθινών συμπερασμάτων. Δεν είναι πλήρως υλοποιημένα, σε αντίθεση με όλα τα υπόλοιπα που είναι πλήρως καθιερωμένα πρότυπα.
- Η εμπιστοσύνη, αφορά μέσα για την παροχή αυθεντικοποίησης της ταυτότητας και την απόδειξη της αξιοπιστίας των δεδομένων, των υπηρεσιών και των πρακτόρων. Μπορεί να είναι εφικτή μέσω των ήδη διαδεδομένων ψηφιακών υπογραφών.

2.1.1 Οντολογία (Ontology)

Οι οντολογίες αποτελούν τον πυρήνα του Σημασιολογικού Ιστού. Η οντολογία μπορεί να οριστεί ως «ένας επίσημος, σαφής και λεπτομερής ορισμός ενός πεδίου γνώσης» (Thomas Gruber, 1993). Εναλλακτικά μπορούμε να πούμε ότι μια οντολογία περιγράφει μια οντότητα με αυστηρό τρόπο και διευκρινίζει τις σχέσεις της με άλλες οντότητες του ίδιου μοντέλου.

Οι σύγχρονες οντολογίες κωδικοποιούνται με χρήση γλωσσών οντολογίας, όπως η OWL. Εμφανίζουν πολλές δομικές ομοιότητες, άσχετα από τη γλώσσα στην οποία είναι γραμμένες. Τα βασικά συστατικά των οντολογιών είναι:

- Στιγμιότυπα (individuals/instances): Είναι τα βασικά συστατικά μιας οντολογίας. Μπορεί να περιέχουν απτά αντικείμενα, όπως ανθρώπους, ζώα και πλανήτες, αλλά και αφηρημένα μέρη όπως λέξεις και νούμερα.
- Κλάσεις (classes): Μπορούν να οριστούν είτε ως αφηρημένες ομάδες, σύνολα ή συλλογές από αντικείμενα, είτε ως αφηρημένα αντικείμενα που θέτουν περιορισμούς στις τιμές κάποιων παραμέτρων ως όρο για τη συμμετοχή σε αυτά. Μπορούν να κατηγοριοποιούν στιγμιότυπα ή άλλες κλάσεις. Οι οντολογίες ποικίλουν όσον αφορά το αν κλάσεις μπορούν να περιέχουν άλλες κλάσεις, αν μια κλάση ανήκει στον εαυτό της, αν υπάρχει μια καθολική κλάση (που περιέχει τα πάντα) κ.ο.κ.
- Χαρακτηριστικά (attributes): Διαστάσεις, ιδιότητες, χαρακτηριστικά, γνωρίσματα, παράμετροι που έχουν τα στιγμιότυπα και οι κλάσεις. Όταν ένα χαρακτηριστικό σχετίζεται με ένα αντικείμενο, εκφράζει ένα γεγονός αποκλειστικά για το αντικείμενο αυτό. Όταν αναφέρεται σε μια κλάση, ισχύει για όλα τα μέλη αυτής.
- Σχέσεις (relations): Δείχνουν πως αντικείμενα, δηλαδή στιγμιότυπα και κλάσεις, σχετίζονται με άλλα αντικείμενα στην οντολογία.
- Συναρτήσεις (functions): Ιδιαίτερες μορφές σχέσεων, που παράγουν κάποιο χαρακτηριστικό με βάση κάποια άλλα διαφορετικά χαρακτηριστικά.
- Περιορισμοί (restrictions): Επίσημα διατυπωμένες περιγραφές για το τι πρέπει να ισχύει ώστε κάποιος ισχυρισμός να γίνει δεκτός ως είσοδος.
- Κανόνες (rules): Δηλώσεις σε μορφή Αν-Τότε πρότασης, που περιγράφουν τα λογικά συμπεράσματα που μπορούν να εξαχθούν από έναν ισχυρισμό συγκεκριμένης μορφής.
- Αξιώματα (Axioms): Ισχυρισμοί σε λογική μορφή που όλοι μαζί αποτελούν τη θεωρία που η οντολογία περιγράφει στο πεδίο εφαρμογής της. Μπορούν να

είναι προτάσεις που προέρχονται από a priori γνώση, αλλά και γνώση που παράγεται από αξιωματικές δηλώσεις.

- Γεγονότα (events): Η αλλαγή χαρακτηριστικών ή σχέσεων.

2.1.2 OWL

Όπως έχουμε προαναφέρει, ο Σημασιολογικός Ιστός αποτελεί ένα όραμα για το μέλλον του Διαδικτύου, στο οποίο η πληροφορία αποκτά σαφή σημασιολογία, καθιστώντας την ευκόλως επεξεργάσιμη από υπολογιστικά συστήματα. Ο Σημασιολογικός Ιστός κτίζεται πάνω στην ικανότητα της XML να ορίζει προσαρμοσμένα σχήματα ετικετών και στην ευέλικτη προσέγγιση της RDF στην αναπαράσταση δεδομένων. Η XML, όμως αποτελεί απλά μία σύνταξη για δομημένα έγγραφα, χωρίς να επιβάλλει σημασιολογικούς περιορισμούς στη σημασιολογία αυτών των εγγράφων. Αν και το XML Schema αποτελεί μία γλώσσα που περιορίζει τη δομή των XML εγγράφων, επεκτείνει απλώς την XML με τύπους δεδομένων. Η RDF, από την άλλη πλευρά, αποτελεί ένα μοντέλο δεδομένων για αντικείμενα («πόρους») και των συσχετίσεών τους, που παρέχει δομές για την έκφραση απλής σημασιολογίας, διατυπωμένης με το συντακτικό της XML, ενώ το RDF Schema που χρησιμοποιείται είναι ένα λεξιλόγιο για την περιγραφή ιδιοτήτων και κλάσεων για δικτυακούς πόρους, με τη δυνατότητα έκφρασης απλών ιεραρχιών γενίκευσης ιδιοτήτων και κλάσεων. Το πρώτο επίπεδο πάνω από την RDF που απαιτείται για το Σημασιολογικό Ιστό είναι μία γλώσσα οντολογιών, η οποία θα μπορεί να περιγράψει τυπικά τη σημασιολογία της ορολογίας που χρησιμοποιείται σε δικτυακά έγγραφα. Αυτή η αναπαράσταση όρων και των αλληλοσυσχετίσεων τους αποκαλείται οντολογία. Εάν τα υπολογιστικά συστήματα αναμένονται να εκτελέσουν χρήσιμες συλλογιστικές διεργασίες σε αυτά τα έγγραφα, η γλώσσα που θα χρησιμοποιηθεί θα πρέπει να εκτείνεται πέρα από τη βασική σημασιολογία που μπορεί να εκφράσει ένα RDF σχήμα.

Η OWL (Web Ontology Language) σχεδιάστηκε για να αντιμετωπίσει αυτή την ανάγκη και αποτελεί μέρος των προτάσεων του W3C που σχετίζονται με τον Σημασιολογικό Ιστό. Αποτελεί βελτίωση της γλώσσας DAML+OIL, ενσωματώνοντας εμπειρία από το σχεδιασμό και την εφαρμογή της, και είναι σχεδιασμένη για χρήση από εφαρμογές που πρέπει να επεξεργαστούν το περιεχόμενο της πληροφορίας, αντί απλώς να το προβάλλουν στο χρήστη. Η OWL υποστηρίζει τη δυνατότητα διατύπωσης πλουσιότερης σημασιολογίας από τις γλώσσες XML, RDF, και RDF Schema, παρέχοντας επιπλέον λεξιλόγιο και τυπική σημασιολογία για την περιγραφή των ιδιοτήτων και των κλάσεων. Μεταξύ άλλων είναι σε θέση να εκφράσει σχέσεις μεταξύ κλάσεων (π.χ. μη αλληλοεπικάλυψη), πληθικότητα (π.χ. «ακριβώς ένα»), ισοδυναμία, πλουσιότερους τύπους ιδιοτήτων και χαρακτηριστικά ιδιοτήτων (π.χ. συμμετρία).

Η OWL σχεδιάστηκε για εφαρμογές που επεξεργάζονται το περιεχόμενο των πληροφοριών και όχι μόνο να τις εκθέτουν στους ανθρώπους. Η τυποποίηση των οντολογιών σε γλώσσα OWL θα κάνει τα δεδομένα στο Web να κατανοούνται από μηχανές και να χρησιμοποιούνται ξανά σε εφαρμογές. Η OWL βασίζεται στην XML και στο RDF-RDF Shema και τα επεκτείνει παρέχοντας επιπλέον λεξιλόγιο και

τυπικούς ορισμούς για την περιγραφή ιδιοτήτων και κλάσεων με στόχο την διευκόλυνση της μηχανικής ερμηνείας των πληροφοριών του Παγκόσμιου Ιστού. Η γλώσσα OWL παρέχει ένα περιγραφικό τρόπο να ορίζει τις έννοιες. Οι πολύπλοκες έννοιες μπορούν να οριστούν με την βοήθεια απλούστερων εννοιών. Το λογικό μοντέλο στο οποίο βασίζεται η OWL μπορεί να ελέγξει το κατά πόσο οι ορισμοί μιας οντολογίας είναι ακριβής και κατά πόσο οι έννοιες είναι συμβατές με τους ορισμούς που δόθηκαν. Η OWL έχει τρεις υπογλώσσες: την OWL-Lite, την OWL-DL και την OWL –Full. Η διαφορά μεταξύ των τριών υπογλωσσών είναι το επίπεδο εκφραστικότητας που διαθέτει η κάθε μια. Η λιγότερο εκφραστική είναι η OWL-Lite και αντίστοιχα η περισσότερο εκφραστική είναι η OWL-Full. Η κάθε μία όμως μπορεί να θεωρηθεί επέκταση της προηγούμενης.

OWL - Lite

Η OWL-Lite προορίζεται σε περιπτώσεις αναπαράστασης απλών ιεραρχικών κλάσεων και ιδιοτήτων των κλάσεων που υπόκεινται σε απλούς περιορισμούς. Η πολυπλοκότητα της είναι αρκετά χαμηλότερη από αυτή των άλλων δύο υπογλωσσών και άρα μπορούμε να κατασκευάσουμε υπολογιστικά εργαλεία πιο εύκολα.

OWL - DL

Βασίζεται στην Περιγραφική Λογική η οποία αποτελεί υποσύνολο της Λογικής Πρώτης Τάξης και άρα είναι κατάλληλη για περιπτώσεις όπου χρειάζεται μέγιστη εκφραστικότητα. Περιλαμβάνει όλες τις γλωσσικές δομές της OWL, οι οποίες όμως μπορούν να χρησιμοποιηθούν υπό συγκεκριμένους περιορισμούς. Μια κλάση μπορεί να είναι υποκλάση πολλών κλάσεων (κληρονομικότητα) αλλά μια δεν μπορεί να αποτελεί πραγμάτωση μιας άλλης.

OWL - Full

Απευθύνεται σε χρήστες για τους οποίους είναι πιο σημαντική η δυνατότητα για μέγιστη εκφραστικότητα ή υπολογιστική πληρότητα των ισχυρισμών τους. Μια κλάση μπορεί να θεωρηθεί ταυτόχρονα και σύνολο ατόμων και μεμονωμένο άτομο. Επίσης επιτρέπει την επέκταση λεξιλογίου OWL και του RDF σχήμα προκειμένου να καλυφθούν ανάγκες. Υπάρχει βέβαια περίπτωση ατέρμονης αναδρομής των ορισμών κάτι που οδηγεί τον αυτόματο συμπερασμό σε OWL – Full οντολογιών να μην είναι εφικτός.

Η επόμενη εικόνα παρουσιάζει ένα παράδειγμα σύνταξης σε OWL.

```
<Ontology ontologyIRI="http://example.com/tea.owl" ...>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Declaration>
    <Class IRI="Tea"/>
  </Declaration>
</Ontology>
```

Εικόνα 2. Παράδειγμα σύνταξης OWL.

2.1.3 RDF

Το RDF (Resource Description Framework) είναι ένα σύνολο προδιαγραφών του World Wide Web Consortium (W3C) που αρχικά σχεδιάστηκε ως μοντέλο διαχείρισης μετα-δεδομένων. Πλέον χρησιμοποιείται ευρέως ως μια γενική μέθοδος για την εννοιολογική περιγραφή και τη μοντελοποίηση πληροφοριών στο διαδίκτυο. Παρέχει τη γλώσσα για την αναπαράσταση των διαδικτυακών πόρων (web resources) οι οποίοι περιλαμβάνουν οποιαδήποτε οντότητα μπορεί να ονομαστεί και για την οποία μπορεί να γίνει αναφορά στο διαδίκτυο, χωρίς να σημαίνει απαραίτητα ότι είναι δυνατή η προσπέλαση της μέσα από αυτό.

Το RDF αποτελεί μέλος των προτύπων για τη διαχείριση μετα-δεδομένων στον Παγκόσμιο Ιστό και απαρτίζει τη βάση για την κωδικοποίηση, ανταλλαγή, επεξεργασία και επαναχρησιμοποίηση μετα-δεδομένων. Η βασική επιδίωξη του RDF είναι να επιτρέψει τον ορισμό της σημασιολογίας που εγκλείεται σε πληροφοριακούς πόρους με τυπικό, διαλειτουργικό και αναγνώσιμο τρόπο. Αυτό επιτυγχάνεται μέσω ενός μηχανισμού για την περιγραφή πληροφοριακών πόρων, ο οποίος δεν κάνει καμία υπόθεση για τη φύση του συγκεκριμένου πεδίου εφαρμογής ή τη δομή του εγγράφου που περιέχει την πληροφορία. Εκτός από τη διαλειτουργικότητα μεταξύ συστημάτων το RDF στοχεύει στην επαναχρησιμοποίηση, διαμοιρασμό και επεκτασιμότητα των μετα-δεδομένων και κατά συνέπεια, στην αυτόματη επεξεργασία πόρων που ανταλλάσσονται μέσω του Διαδικτύου.

Το RDF μοντέλο δεδομένων είναι παρόμοιο με τις κλασσικές προσεγγίσεις στην εννοιολογική περιγραφή, όπως τα διαγράμματα οντοτήτων-συσχετίσεων και κλάσεων, καθώς βασίζεται στην ιδέα της δημιουργίας δηλώσεων για διαδικτυακούς πόρους με τη μορφή εκφράσεων «αντικείμενο-κατηγορούμενο-αντικείμενο». Αυτές οι εκφράσεις είναι γνωστές ως τριάδες ή τριπλέτες (triples). Το υποκείμενο δηλώνει τον πόρο, και το κατηγορούμενο δείχνει χαρακτηριστικά ή γνωρίσματα του πόρου και εκφράζει μια σχέση ανάμεσα στο υποκείμενο και το αντικείμενο . Ένα παράδειγμα μιας RDF triple είναι η εξής:

«Lord of the Rings» συγγραφέας «John R.R.Tolkien».

Ο όρος «Lord of the Rings» αποτελεί το υποκείμενο, το κατηγορούμενο είναι το «συγγραφέας», και το αντικείμενο είναι ο όρος «John R.R.Tolkien».

Αυτός ο μηχανισμός περιγραφής πόρων είναι βασικό συστατικό της δραστηριότητας του Σημασιολογικού Ιστού, ένα επαναστατικό στάδιο στο οποίο αυτοματοποιημένο λογισμικό θα μπορεί να αποθηκεύει, ανταλλάσσει και χρησιμοποιεί πληροφορίες ,αναγνωρίσιμες από τις μηχανές, διαμοιρασμένες στο διαδίκτυο. Αυτό με τη σειρά του θα επιτρέπει στους χρήστες να ασχολούνται με τις πληροφορίες με μεγαλύτερη αποτελεσματικότητα και βεβαιότητα. Το απλό μοντέλο δεδομένων του RDF και η ικανότητα του να αναπαριστά διαφορετικές αφηρημένες έννοιες έχει οδηγήσει στην αυξανόμενη χρήση του και σε δραστηριότητες άσχετες με τον Σημασιολογικό Ιστό.

Μια συλλογή από δηλώσεις RDF έμφυτα αναπαριστά έναν επισημασμένο, κατευθυνόμενο γράφο. Γι αυτό, ένα μοντέλο δεδομένων βασισμένο στο RDF ταιριάζει περισσότερο σε κάποια είδη αναπαράστασης γνώσης απ’ ότι στ σχεσιακό μοντέλο ή άλλα οντολογικά μοντέλα.

Το RDF είναι ένα αφηρημένο μοντέλο με πολλές μορφές συντακτικού (μορφές αρχείων), και επομένως ο συγκεκριμένος τρόπος που ένας πόρος ή μια τριάδα κωδικοποιείται εξαρτάται από τη μορφή που εφαρμόζεται.

Οι κυριότερες μορφές του RDF είναι οι εξής:

- **Turtle:** Μια συμπαγής, φιλική προς τον άνθρωπο μορφή.
- **N-Triples:** Απλή, εύκολα προσπελάσιμη μορφή, λιγότερο συμπαγής από την Turtle.
- **N-Quads:** Υπερσύνολο της N-Triples, για τη σύνταξη πολλαπλών RDF γράφων.
- **JSON-LD**: Μορφή βασισμένο σε JSON.
- **RDF/XML**: Μια σύνταξη βασισμένη σε XML. Ήταν το πρώτο καθιερωμένο format για αρχεία RDF.

Το RDF/XML πολλές φορές καλείται παραπλανητικά και απλά RDF, μιας και ήταν ιστορικά το πρώτο W3C πρότυπο RDF. Ωστόσο, είναι σημαντικό να διαχωρίζουμε τη μορφή RDF/XML από το αφηρημένο μοντέλο RDF. Αν και η μορφή RDF/XML χρησιμοποιείται ακόμα, άλλες μορφές γίνονται ολοένα και πιο δημοφιλείς γιατί είναι πιο φιλικές προς τον άνθρωπο, και επίσης καθώς κάποιοι γράφοι δεν μπορούν να αναπαρασταθούν σε RDF/XML, λόγω περιορισμών του συντακτικού της XML.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#" xmlns:eric="http://www.w3.org/People/EM/contact#"
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:mailbox rdf:resource="mailto:e.miller123(at)example"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:personalTitle>Dr.</contact:personalTitle>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <rdf:type rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person"/>
  </rdf:Description>
</rdf:RDF>
```

Εικόνα 3. Παράδειγμα συντακτικού RDF/XML.

2.1.3.1 RDF Λεξιλόγιο

Ο ορισμός των γνωρισμάτων/σχέσεων που χρησιμοποιούνται για την περιγραφή των πόρων και της σημασιολογίας που φέρουν, επιτυγχάνεται μέσω του RDFS. Ένα RDF σχήμα (schema) αποτελείται από τις δηλώσεις κλάσεων, γνωρισμάτων και σχέσεων μεταξύ των κλάσεων, όπου μία κλάση χρησιμοποιείται για να ομαδοποιήσει λογικά ομοειδείς πόρους. Με άλλα λόγια, ο μηχανισμός των RDF σχημάτων προσφέρει ένα σύστημα τύπων για τα RDF μοντέλα, ένα λεξιλόγιο για τους έγκυρους όρους που μπορούν να χρησιμοποιηθούν για την περιγραφή των πληροφοριακών πόρων. Τα λεξιλόγια αυτά δημιουργούνται ανεξάρτητα, ενώ κοινότητες χρηστών μπορούν να επαναχρησιμοποιήσουν λεξιλόγια άλλων χρηστών είτε αυτούσια είτε εκλεπτύνοντάς τα σύμφωνα με τις ανάγκες της εκάστοτε κοινότητας. Ένας πολύ σημαντικός μηχανισμός για την ανάπτυξη RDF σχημάτων και τη δημιουργία RDF περιγραφών είναι ο μηχανισμός ονοματοδοσίας XML (XML namespace), ο οποίος επιτρέπει την

επαναχρησιμοποίηση όρων από διαφορετικά σχήματα. Ένας χώρος ονοματοδοσίας είναι ένα σύνολο από κλάσεις και/ή ιδιότητες στον οποίον αποδίδεται ένα μοναδικό αναγνωριστικό (URI). Έτσι, ένα RDF σχήμα αποτελεί ένα χώρο ονοματοδοσίας που προσδιορίζεται μοναδικά από ένα URI. Με τη χρήση ενός XML χώρου ονοματοδοσίας, περιγραφικοί όροι (δηλαδή ονόματα κλάσεων ή/και ιδιοτήτων) αναγνωρίζονται μοναδικά από το URI του σχήματος, στο οποίο ορίζονται (το οποίο παίζει το ρόλο ενός προθέματος) ως κανονικοί πόροι στο Διαδίκτυο, γεγονός που επιτρέπει την επαναχρησιμοποίησή τους.

Το λεξιλόγιο που ορίζεται από τις προδιαγραφές του RDF είναι το εξής:

Κλάσεις

rdf

rdf:XMLLiteral	Η κλάση των XML literal values.
rdf:Property	Η κλάση των ιδιοτήτων.
rdf:Statement	Η κλάση των RDF δηλώσεων.
rdf:Alt, rdf:Bag, rdf:Seq	Υποκλάσεις της rdf:Container, συλλογές εναλλακτικών, συλλογές μη-ταξινομημένων και ταξινομημένων αντικειμένων αντίστοιχα.
rdf:List	Η κλάση των RDF λιστών.
rdf:nil	Στιγμιότυπο της rdf:List, αναπαριστά την κενή λίστα.

rdfs

rdfs:Resource	Η κλάση των πόρων, ουσιαστικά τα πάντα
rdfs:Literal	Η κλάση των literal values, όπως ακέραιοι και strings
rdfs:Class	Η κλάση των κλάσεων
rdfs:Datatype	Η κλάση των RDF τύπων δεδομένων
rdfs:Container	Η κλάση των RDF συλλογών
rdfs:ContainerMembershipProperty	Η κλάση των ιδιοτήτων μέλους των συλλογών. Όλες είναι υποκατηγορίες της rdfs:member

Ιδιότητες

rdf

rdf:type	Στιγμιότυπο της rdf:Property, χρησιμοποιείται για να δηλώσει ότι ένας πόρος είναι στιγμιότυπο μιας κλάσης.
rdf:first	Το πρώτο στοιχείο στην σχετική RDF λίστα
rdf:rest	Τα υπόλοιπα στοιχεία της σχετικής RDF λίστας
rdf:value	Χαρακτηριστική ιδιότητα για δομημένες τιμές
rdf:subject	Το υποκείμενο της σχετικής rdf δήλωσης
rdf:predicate	Το κατηγορούμενο της σχετικής rdf δήλωσης
rdf:object	Το αντικείμενο της σχετικής rdf δήλωσης

rdf

rdfs:subClassOf	Το στοιχείο είναι υποκλάση μιας κλάσης
rdfs:subPropertyOf	Το στοιχείο είναι υπο-ιδιότητα μιας ιδιότητας
rdfs:domain	Το πεδίο της σχετικής ιδιότητας
rdfs:range	Η εμβέλεια της σχετικής ιδιότητας
rdfs:label	Ένα ανθρώπινα κατανοητό όνομα για το στοιχείο
rdfs:comment	Μια περιγραφή του σχετικού πόρου
rdfs:member	Ένα μέλος του σχετικού πόρου
rdfs:seeAlso	Επιπλέον πληροφορίες για το σχετικό πόρο
rdfs:isDefinedBy	Ο ορισμός του σχετικού πόρου

Αυτό το λεξιλόγιο χρησιμοποιείται ως βάση για το RDF Schema, όπου και επεκτείνεται.

2.1.3.2 Αναγνώριση πόρων

Το υποκείμενο μιας δήλωσης RDF είναι είτε ένα URI (Uniform Resource Identifier) είτε ένας κενός κόμβος. Και τα δυο υποδηλώνουν πόρους. Πόροι που υποδεικνύονται από κενούς κόμβους ονομάζονται ανώνυμοι κόμβοι και δεν είναι άμεσα αναγνωρίσιμοι από την δήλωση RDF. Το κατηγορούμενο είναι ένα URI που επίσης υποδηλώνει κάποιο πόρο, ο οποίος αντιπροσωπεύει μια σχέση. Το αντικείμενο τέλος μπορεί να είναι ένα URI, κενός κόμβος ή μια συμβολοσειρά.

Σε εφαρμογές Σημασιολογικού Ιστού, και σε δημοφιλείς εφαρμογές του RDF, οι πόροι τείνουν να αναπαρίστανται από URIs που σκόπιμα υποδηλώνουν πραγματικά δεδομένα του διαδικτύου, στα οποία έχουν και πρόσβαση. Το RDF ωστόσο δεν περιορίζεται γενικά στην περιγραφή πόρων του διαδικτύου. Στην πραγματικότητα, ένα URI δεν χρειάζεται καν να παραπέμπει κάπου. Για παράδειγμα ένα URI της μορφής «http:» που χρησιμοποιείται σε ένα RDF δεν είναι ανάγκη να αναπαριστά κάποιον πόρο προσπελάσιμο μέσω HTTP. Μπορεί να μην αναπαριστά απολύτως τίποτα.

Επομένως, οι χρήστες του RDF πρέπει να συμφωνούν στη σημασιολογία των αναγνωριστικών των πόρων. Μια τέτοια συμφωνία δεν είναι έμφυτη στο RDF, αν και υπάρχουν κάποια επιβλεπόμενα λεξιλόγια που χρησιμοποιούνται, όπως το Dublin Core Metadata. Ο σκοπός της δημοσίευσης στο διαδίκτυο οντολογιών βασισμένων σε RDF είναι για η καθιέρωση των επιδιωκόμενων σημασιών των αναγνωριστικών που χρησιμοποιούνται για την έκφραση δεδομένων στο RDF. Για παράδειγμα το URI

http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#Merlot

προορίζεται από τους δημιουργούς του να αναφέρεται στην κλάση που περιέχει όλα τα κρασιά Merlot όλων των οινοποιών. Στιγμιότυπα του URI αναπαριστούν το καθένα μια κλάση με τα κρασιά που παράγει ένας οινοποιός. Αυτός ο ορισμό εκφράζεται από την OWL οντολογία, η οποία είναι ένα RDF αρχείο, στην οποία συναντάται. Χωρίς προσεκτική ανάλυση του ορισμού, ένα στιγμιότυπο του παραπάνω URI θα μπορούσε να ερμηνευτεί ως κάτι τελείως διαφορετικό.

2.1.3.3 N-Triples

Η N-Triples είναι μια από τις μορφές της RDF για αποθήκευση και μετάδοση δεδομένων. Είναι line-based, μορφή συντακτικού RDF γράφων απλού κειμένου, και υποκατηγορία της Turtle.

Σχεδιάστηκε για να είναι πιο απλή από τις μορφές Notation 3 και Turtle, και επομένως πιο εύκολή στην προσπέλαση και δημιουργία από λογισμικό. Ωστόσο δεν έχει κάποιες από τις συντομεύσεις που έχουν άλλες μορφές RDF συντακτικού, και γι αυτό μπορεί να είναι δύσκολη στην ανάγνωση, και να χρειάζεται χειρόγραφη διατύπωση μεγάλου όγκου δεδομένων.

Ένας RDF γράφος μπορεί να αναπαρασταθεί με το συντακτικό N-Triples με πολύ λίγους διαφορετικούς τρόπους. Αυτό το καθιστά πολύ χρήσιμο στην παροχή απαντήσεων ενός μοντέλου για είσοδο ενός συνόλου από testcases.

Όσον αφορά τους κανόνες του συντακτικού αυτού, κάθε γραμμή έχει τη μορφή ενός σχολίου ή μιας δήλωσης. Μια δήλωση αποτελείται όπως είναι αναμενόμενο από τα τρία γνωστά μέρη του RDF, το υποκείμενο, τα κατηγορούμενο και το αντικείμενο, τα οποία χωρίζονται με whitespaces. Η γραμμή καταλήγει με μια τελεία.

Τα υποκείμενα μπορούν να είναι είτε URIs είτε κενοί κόμβοι, και τα αντικείμενα URIs, κενοί κόμβοι ή literals. Τα URIs περιβάλλονται από τους χαρακτήρες ‘<’ και ‘>’. Οι κενοί κόμβοι αναπαρίστανται από μια αλφαριθμητική συμβολοσειρά, που ξεκινάει με τους χαρακτήρες ‘_:’ . Τα literals αναπαρίστανται με ASCII συμβολοσειρές, που περικλείονται σε χαρακτήρες «”», και προαιρετικά καταλήγουν

με μια ένδειξη γλώσσας ή τύπου δεδομένων. Οι δείκτες γλώσσας είναι της μορφής «@(RFC 3066 Language tag)», ενώ οι δείκτες τύπου δεδομένων της μορφής «^^(URI)».

Τα σχόλια τέλος, είναι γραμμές που ξεκινάνε με το σύμβολο '#'.

Ένα παράδειγμα συντακτικού N-Triples είναι το ακόλουθο.

```
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ↵
   <http://xmlns.com/foaf/0.1/Document> .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://purl.org/dc/terms/title> "N-Triples"@en-US .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://xmlns.com/foaf/0.1/maker> _:art .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://xmlns.com/foaf/0.1/maker> _:dave .
_:art <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
_:art <http://xmlns.com/foaf/0.1/name> "Art Barstow".
_:dave <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
_:dave <http://xmlns.com/foaf/0.1/name> "Dave Beckett".
```

Εικόνα 4. Παράδειγμα δομής συντακτικού N-Triples.

2.1.4 SPARQL

Η SPARQL (SPARQL Protocol And RDF Query Language) είναι μια γλώσσα ερωτημάτων RDF, δηλαδή μια γλώσσα σημασιολογικών ερωτημάτων για βάσεις δεδομένων, ικανή να ανακτήσει και να διαχειριστεί δεδομένα αποθηκευμένα σε RDF μορφή. Έγινε πρότυπο από το RDF Data Working Group (DAWG) του World Wide Web Consortium και θεωρείται ένα από τα βασικά κομμάτια του Σημασιολογικού Ιστού.

Η SPARQL επιτρέπει τα ερωτήματα να αποτελούνται από τριάδες, συζεύξεις, διαζεύξεις, και προαιρετικά μοτίβα. Αυτές οι τριάδες που αποτελούν την βάση δεδομένων έχουν τη μορφή «υποκείμενο-κατηγορούμενο-αντικείμενο».

Τα RDF δεδομένα μπορούν επίσης να θεωρηθούν σε όρους σχεσιακής SQL βάσης δεδομένων ως ένας πίνακας με τρεις στήλες, τη στήλη υποκειμένων, κατηγορούμενων και αντικειμένων. Αντίθετα με τις σχεσιακές βάσεις δεδομένων, η στήλη αντικειμένων είναι ετερογενής, και ο τύπος δεδομένων κάθε κελιού είτε προσδιορίζεται από μια οντολογία είτε υπονοείται από την τιμή του αντίστοιχου κατηγορούμενου. Εναλλακτικά, συγκριτικά πάλι με τη σχεσιακή SQL, όλες οι τριάδες για ένα συγκεκριμένο υποκείμενο μπορούν να αναπαρασταθούν σαν μια γραμμή, με το υποκείμενο να είναι το βασικό κλειδί, και κάθε πιθανό κατηγορούμενο να είναι μια στήλη, με το αντικείμενο να είναι η τιμή μέσα στο κελί. Ωστόσο, η SPARQL/RDF γίνεται πιο εύκολη και αποδοτική για στήλες που θα μπορούσαν να περιέχουν πολλαπλές τιμές, και που η στήλη αυτή καθεαυτή θα μπορούσε να συνδεδεμένη μεταβλητή στην ερώτηση, και όχι αυστηρά ορισμένη.

Η SPARQL λοιπόν, παρέχει ένα πλήρες σύνολο από αναλυτικές λειτουργίες ερωτημάτων, όπως JOIN, SORT, AGGREGATE για δεδομένα των οποίων το διάγραμμα (schema) είναι εγγενώς μέρος των δεδομένων, αντί να χρειάζεται ένα ξεχωριστό ορισμό. Οι πληροφορίες του διαγράμματος (δηλαδή η οντολογία) ωστόσο

συχνά παρέχεται εξωτερικά ώστε διαφορετικά σύνολα δεδομένων να μπορούν να ενωθούν χωρίς ασάφειες. Επιπρόσθετα, η SPARQL παρέχει ειδική σύνταξη για διάσχιση γράφων για δεδομένα που μπορούν να θεωρηθούν ως γράφος.

Το παρακάτω παράδειγμα δείχνει ένα απλό ερώτημα που αξιοποιεί τον ορισμό της οντολογίας foaf, που καλείται συχνά «friend-of-a-friend» οντολογία. Συγκεκριμένα, το ερώτημα επιστρέφει ονόματα και mails κάθε ατόμου στο σύνολο δεδομένων.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}
```

Εικόνα 5. Παράδειγμα SPARQL query.

Η αναφορά στο υποκείμενο σε αυτό το ερώτημα γίνεται με τη μεταβλητή «?person» καθαρά για λόγους κατανόησης. Μιας και το πρώτο στοιχείο κάθε τριάδας είναι πάντα το υποκείμενο, θα μπορούσε να χρησιμοποιηθεί οποιοδήποτε όνομα μεταβλητής. Ο μόνος περιορισμός είναι αυτό το όνομα να είναι κοινό σε όλες τις γραμμές του ερωτήματος ώστε να μπορέσει να γίνει η σύνδεση των τριάδων με ίδιο υποκείμενο.

Το αποτέλεσμα είναι ένα σύνολο γραμμών της μορφής «?person, ?name, ?email». το συγκεκριμένο ερώτημα επιστρέφει μόνο τα ?name και ?email, ίσως γιατί το ?person είναι συνήθως ένα πολύπλοκο URI, καθόλου human-friendly.

Αυτό το ερώτημα μπορεί να διαμοιραστεί σε πολλά SPARQL endpoints, τα οποία είναι υπηρεσίες που δέχονται ερωτήματα SPARQL και επιστρέφουν αποτελέσματα, και τα αποτελέσματα να υπολογιστούν και να συγκεντρωθούν. Αυτή η διαδικασία είναι γνωστή ως ενωτικά ερωτήματα.

Μορφές ερωτημάτων

Στη περίπτωση ερωτημάτων που διαβάζουν δεδομένα από βάσεις, η SPARQL προσδιορίζει τέσσερεις διαφορετικές παραλλαγές ερωτημάτων που η καθεμία εξυπηρετεί διαφορετικό σκοπό:

- **SELECT query:** Χρησιμοποιείται για την εξαγωγή ακατέργαστων τιμών από ένα SPARQL endpoint. Τα αποτελέσματα επιστρέφονται με τη μορφή ενός πίνακα.
- **CONSTRUCT query:** Χρησιμοποιείται για την εξαγωγή πληροφοριών από ένα SPARQL endpoint και την μετατροπή των αποτελεσμάτων σε RDF μορφή.
- **ASK query:** Επιστρέφει ένα απλό True/False αποτέλεσμα για ένα ερώτημα σε SPARQL endpoint.

- **DESCRIBE query:** Χρησιμοποιείται για την εξαγωγή ενός RDF γράφου από ένα SPARQL endpoint, για τα περιεχόμενα του οποίου αποφασίζει το endpoint αν είναι χρήσιμη πληροφορία.

Καθεμία από αυτές τις μορφές ερωτημάτων έχει ένα τμήμα «WHERE» που περιορίζει το ερώτημα, εκτός από το DESCRIBE query, όπου αυτό είναι προαιρετικό.

2.2 Επεξεργασία φυσικού λόγου (Natural Language Processing, NLP)

Η επεξεργασία φυσικού λόγου είναι ένα πεδίο της επιστήμης υπολογιστών, της τεχνητής νοημοσύνης και της υπολογιστικής γλωσσολογίας που ασχολείται με την αλληλεπίδραση ανάμεσα σε υπολογιστές και τις ανθρώπινες (φυσικές) γλώσσες.

Οι σύγχρονοι NLP αλγόριθμοι βασίζονται στη μηχανική μάθηση. Ενώ οι παλιότερες απόπειρες στη επεξεργασία φυσικού λόγου περιστρέφονταν γύρω από τη κωδικοποίηση μεγάλων ομάδων από κανόνες με το χέρι, η μηχανική μάθηση προτείνει τη χρήση πιο γενικών αλγορίθμων μάθησης για την αυτοματοποιημένη μάθηση αυτών των κανόνων μέσα από την ανάλυση μεγάλων σορών από τυπικά ρεαλιστικά παραδείγματα. Ένας σορός είναι ένα σύνολο εγγράφων, ή ακόμα και προτάσεων, που έχουν επισημανθεί με τις σωστές τιμές προς μάθηση.

Οι αλγόριθμοι μηχανικής μάθησης που εφαρμόζονται σε προβλήματα επεξεργασίας φυσικού λόγου παίρνουν ως είσοδο ένα μεγάλο σύνολο από χαρακτηριστικά, που παράγονται από τα δεδομένα εισόδου. Στη συνέχεια, βάσει στατιστικών μοντέλων, παίρνουν πιθανολογικές αποφάσεις και αναθέτουν βάρη σε κάθε χαρακτηριστικό. Αυτά τα μοντέλα έχουν το πλεονέκτημα ότι μπορούν να εκφράσουν τη σχετική βεβαιότητα πολλών διαφορετικών πιθανών απαντήσεων, παράγοντας πιο αξιόπιστα αποτελέσματα όταν ένα τέτοιο μοντέλο αποτελεί κομμάτι ενός μεγαλύτερου συστήματος.

Συστήματα που βασίζονται στη μηχανική μάθηση έχουν πολλά πλεονεκτήματα έναντι των ιδιόχειρων κανόνων:

- Οι διαδικασίες μάθησης που χρησιμοποιούνται στη μηχανική μάθηση επικεντρώνονται αυτόματα στις πιο κοινές περιπτώσεις, κάτι που δεν είναι πάντα προφανές όταν οι κανόνες γράφονται με το χέρι.
- Χρησιμοποιούν αλγόριθμους στατιστικών συμπερασμάτων που παράγουν μοντέλα που είναι ευσταθή για ασυνήθιστες (λέξεις και δομές που δεν έχουν συναντηθεί προηγουμένως) ή λανθασμένες (λέξεις με ορθογραφικά λάθη) εισόδους.
- Είναι πιο ακριβή, και μπορούν να αυξήσουν αυτή την ακρίβεια απλά με την επεξεργασία περισσότερων παραδειγμάτων εισόδου. Αντίθετα, τα συστήματα με ιδιόχειρους κανόνες μπορούν να γίνουν πιο ακριβή μόνο με τη σύνταξη μεγαλύτερων, πιο πολύπλοκων κανόνων. Υπάρχει ένα ρεαλιστικό όριο στην πολυπλοκότητα τους, ενώ στα αυτοματοποιημένα συστήματα το μόνο που

αυξάνεται για περισσότερο όγκο δεδομένων εισόδου είναι ο απαιτούμενος χρόνος εργασίας, και όχι η πολυπλοκότητα που μένει σχετικά σταθερή.

2.2.1 Λημματοποίηση (lemmatisation)

Η λημματοποίηση είναι μια διαδεδομένη τεχνική που εφαρμόζεται στην επεξεργασία φυσικού λόγου. Πρόκειται για τη διαδικασία κατά την οποία οι διαφορετικές μορφές μιας λέξης, είτε αυτές είναι διαφορετικές κλίσεις, πρόσωπα, πτώσεις, γένη ή χρόνοι, ομαδοποιούνται ώστε να αναλυθούν ως μια, μοναδική λέξη.

Η λημματοποίηση, ως κομμάτι μιας υπολογιστικής διαδικασίας, είναι ένα πολύ απαιτητικό πρόβλημα, που μπορεί να περιλαμβάνει πολύπλοκες διαδικασίες, όπως την κατανόηση του νοήματος μιας πρότασης ή την ταυτοποίηση τι μέρους το λόγου είναι μια λέξη σε αυτή την πρόταση. Ως εκ τούτου, η υλοποίηση ενός εργαλείου λημματοποίησης για κάθε διαφορετική ανθρώπινη γλώσσα μπορεί να αποδειχθεί πολύ δύσκολη.

Για παράδειγμα, στην Αγγλική γλώσσα το ρήμα “talk” μπορεί να εμφανιστεί και ως “talk”, “talked”, “talks”, “talking” και με άλλες μορφές. Η λημματοποίηση θα επιστρέφει τη βασική μορφή της λέξης , που είναι το “talk”. Η βασική μορφή μιας λέξης κατά γενικό κανόνα είναι αυτή που εμφανίζεται στα λεξικά της συγκεκριμένης γλώσσας, και καλείται «λήμμα» της.

Σε ένα άλλο παράδειγμα, η πρόταση “The boy’s cars are different color” θα επιστρέψει κάτι της μορφής :

String s[] = {boy, car, be, differ, color}.

Η λημματοποίηση είναι πολύ παρόμοια διαδικασία με μια άλλη τεχνική της επεξεργασίας φυσικού λόγου, το stemming. Η διαφορά είναι ότι στο stemming δεν λαμβάνονται υπόψιν τα συμφραζόμενα του κειμένου, και επομένως δεν μπορεί να διακρίνει ανάμεσα σε λέξεις που έχουν διαφορετικό νόημα ανάλογα με το μέρος του λόγου που είναι. Για παράδειγμα η λέξη “meeting” μπορεί να αναφέρεται στο ουσιαστικό (“this is a meeting”), είτε στο απαρέμφατο του ρήματος “meet” (“we are meeting soon”). Η λημματοποίηση αντίθετα, θα εντοπίσει αυτή τη διαφορά και θα επιστρέψει διαφορετικό αποτέλεσμα ανάλογα την περίπτωση.

2.3 Μηχανική μάθηση (machine learning)

Η μηχανική μάθηση είναι μια περιοχή της τεχνητής νοημοσύνης η οποία αφορά αλγορίθμους και μεθόδους που επιτρέπουν στους υπολογιστές να «εκπαιδεύονται». Με τη μηχανική μάθηση καθίσταται εφικτή η κατασκευή προσαρμόσιμων προγραμμάτων υπολογιστών τα οποία λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι τη διαίσθηση των μηχανικών που τα προγραμμάτισαν. Η μηχανική μάθηση επικαλύπτεται σημαντικά με τη στατιστική, αφού και τα δυο πεδία μελετούν την ανάλυση δεδομένων.

Βασική παράμετρος στην κατηγοριοποίηση ων αλγορίθμων μηχανικής μάθησης είναι το επιθυμητό αποτέλεσμα.

Στην επιβλεπόμενη μάθηση (supervised learning), υπάρχει ένα επιπλέον σύνολο δεδομένων, τα δεδομένα εκπαίδευσης, που μπορούν να θεωρηθούν και ως παραδείγματα. Κάθε αντικείμενο των δεδομένων εκπαίδευσης αποτελείται από ένα ζεύγος δεδομένων, ένα εισόδου (συνήθως είναι διάνυσμα) και μια επιθυμητή τιμή εξόδου. Ένας αλγόριθμος επιβλεπόμενης μάθησης αναλύει τα δεδομένα εκπαίδευσης και παράγει μια συνάρτηση, που χρησιμοποιείται για την κατανομή των νέων δεδομένων εισόδου. Στο ιδανικό σενάριο, ο αλγόριθμος θα ταξινομήσει σωστά ακόμα και περιπτώσεις εισόδου που δεν έχει ξαναδεί, και που δεν υπήρχαν στα δεδομένα εκπαίδευσης.

Αντίθετα, στη μη-επιβλεπόμενη μάθηση (unsupervised learning) δεν υπάρχει κάποιο σύνολο δεδομένων εκπαίδευσης, και ως εκ τούτου δεν υπάρχει feedback από το περιβάλλον της μηχανής για να εκτιμηθεί άμεσα η επιτυχία ή η αποτυχία μια λύσης. Η διαδικασία εύρεσης της λύσης έγκειται στον εντοπισμό μοτίβων στο χώρο των δεδομένων.

Τέλος, στην ενισχυτική μάθηση (reinforcement learning), σκοπός του συστήματος μάθησης είναι να μεγιστοποιήσει μια συνάρτηση του αριθμητικού σήματος ενίσχυσης (ανταμοιβή), για παράδειγμα την αναμενόμενη τιμή του σήματος ενίσχυσης στο επόμενο βήμα. το σύστημα δεν καθοδηγείται από κάποιον εξωτερικό επιβλέποντα για το ποια ενέργεια θα πρέπει να ακολουθήσει αλλά πρέπει να ανακαλύψει μόνο του ποιες ενέργειες είναι αυτές που θα το αποφέρουν το μεγαλύτερο κέρδος.

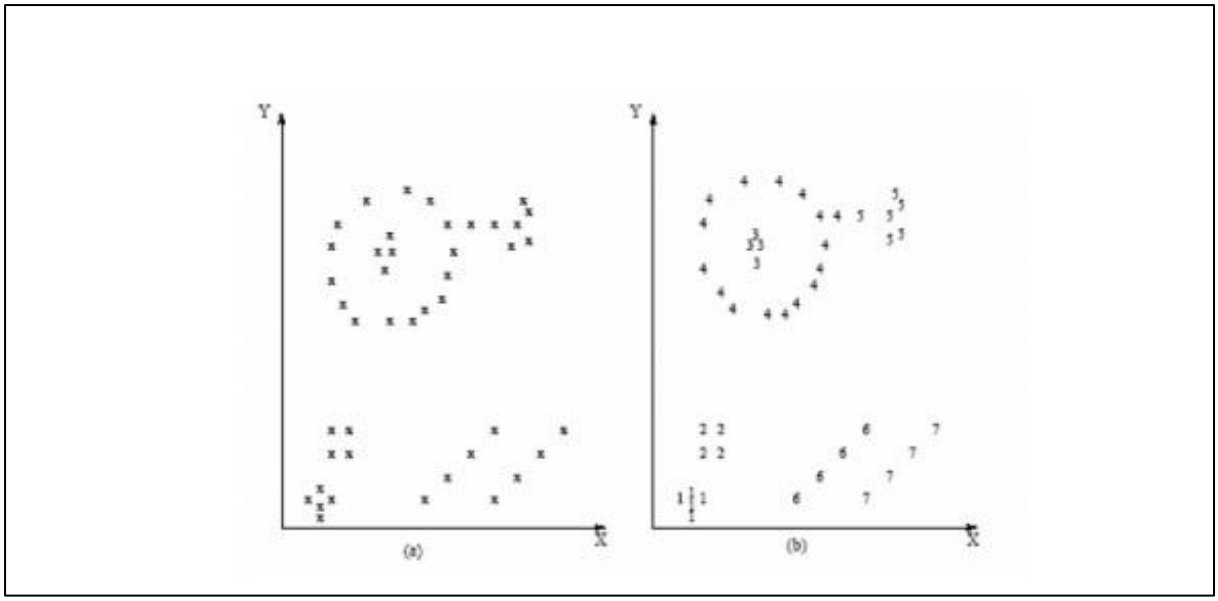
Κατ’ εξοχήν παράδειγμα μη-επιβλεπόμενης μάθησης είναι η συσταδοποίηση.

2.3.1 Συσταδοποίηση Δεδομένων (Data Clustering)

Η συσταδοποίηση ή ομαδοποίηση είναι η διαδικασία της ομαδοποίησης ενός συνόλου αντικειμένων με τέτοιο τρόπο, ώστε αντικείμενα που βρίσκονται στην ίδια ομάδα (συστάδα ή cluster) να είναι πιο όμοια (με κάποιο κριτήριο) μεταξύ τους, απ’ ότι με τα αντικείμενα των άλλων ομάδων.

Η διαφορά της συσταδοποίησης δεδομένων από την ταξινόμηση δεδομένων (data classification) είναι ότι στην ταξινόμηση οι ομάδες στις οποίες θα τοποθετηθούν τα δεδομένα είναι προκαθορισμένες. Αυτό σημαίνει ότι είναι εκ των προτέρων γνωστός ο αριθμός των ομάδων, τα ονόματα και οι ταυτότητες τους. Είναι και αυτό ένα σύστημα μάθησης μιας και οι ετικέτες που δίνονται από τα διαθέσιμα πρότυπα χρησιμοποιούνται ώστε να μάθει το σύστημα ταξινόμησης την περιγραφή κάθε κλάσης και να είναι σε θέση να ταξινομήσει ένα νέο πρότυπο. Αντίθετα, στη συσταδοποίηση δεδομένων τονίζεται ιδιαίτερα ότι οι ομάδες δεν προϋπάρχουν αλλά αποφασίζονται από τον αλγόριθμο κατά δυναμικό τρόπο. Στη συσταδοποίηση δηλαδή, υπάρχει ένα σύνολο δεδομένων το οποίο πρέπει να διαχειριστεί ώστε από αυτό να προκύψουν δυναμικά τα clusters (είναι δηλαδή data driven). Κάθε ένα από τα clusters διατηρεί ένα κέντρο, συνήθως το πιο κεντρικό στοιχείο της.

Ένα παράδειγμα δίνεται στην ακόλουθη εικόνα, όπου αριστερά παρουσιάζεται το αρχικό σύνολο των στοιχείων πριν τη συσταδοποίηση και δεξιά η καταχώρηση των στοιχείων σε clusters.



Εικόνα 6. Συσταδοποίηση δεδομένων

Η συσταδοποίηση δεν είναι κάποιος συγκεκριμένος αλγόριθμος, αλλά η γενικότερη μεθοδολογία. Μπορεί να επιτευχθεί με διάφορους αλγορίθμους, κάποιους από τους οποίους θα αναφέρουμε παρακάτω. Οι αλγόριθμοι αυτοί μπορεί να παρουσιάζουν σημαντικές διαφορές όσον αφορά το πώς ορίζουν την έννοια του cluster και το πόσο αποδοτικά τους βρίσκουν.

Για την επίλυση ενός προβλήματος συσταδοποίησης, συνήθως ακολουθούνται τα παρακάτω βήματα:

- Αναπαράσταση των προτύπων (επιλεκτικά μπορεί να γίνει εξαγωγή των χαρακτηριστικών και επιλογή).
- Καθορισμός μιας μετρικής, ενδεικτικής της γειτνίασης των προτύπων, ανάλογα με τον τύπο δεδομένων.
- Τεχνική συσταδοποίησης των δεδομένων.
- Αφαίρεση δεδομένων (προαιρετική).
- Αξιολόγηση του τελικού αποτελέσματος.

Είναι χαρακτηριστικό ότι πρόκειται για διαδικασία με επανατροφοδότηση. Το αποτέλεσμα της διαδικασίας επανατροφοδοτείται στο σύστημα, το οποίο συνδυάζοντας το αποτέλεσμα αυτό με τις υπόλοιπες εισόδους, προχωράει στην εξαγωγή χαρακτηριστικών και στον υπολογισμό των σχέσεων ομοιότητας, με στόχο την τελική εξαγωγή των ομάδων.

Η αναπαράσταση των προτύπων αναφέρεται στο πλήθος των κλάσεων, το πλήθος των διαθέσιμων προτύπων και το πλήθος, τύπο και κλίμακα των χαρακτηριστικών που είναι διαθέσιμα στον συγκεκριμένο αλγόριθμο συσταδοποίησης. Ωστόσο μερικά από τα προηγούμενα δεν είναι πάντα διαθέσιμα. Ενδιαφέρον παρουσιάζει η διαδικασία της επιλογής χαρακτηριστικών κατά την οποία επιλέγονται τα πιο σημαντικά χαρακτηριστικά των στοιχείων τα οποία θα χρησιμοποιηθούν στο clustering. Επιπλέον, η διαδικασία της εξαγωγής χαρακτηριστικών χρησιμοποιεί έναν ή περισσότερους μετασχηματισμούς των χαρακτηριστικών εισόδου, για τη παραγωγή

άλλων, νέων, τα οποία πιθανόν να είναι πιο ενδιαφέροντα. Οποιαδήποτε από τις τεχνικές αυτές, μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός συνόλου με τα πιο κατάλληλα χαρακτηριστικά, που θα χρησιμοποιηθεί για την αναπαράσταση των στοιχείων που προορίζονται για συσταδοποίηση.

Η γειτνίαση των προτύπων συνήθως μετριέται με βάση μια συνάρτηση απόστασης που ορίζεται για ζεύγη προτύπων. Η πιο απλή συνάρτηση απόστασης είναι η Ευκλείδεια. Η συνάρτηση απόστασης η οποία επιλέγεται, αποτελεί κάθε φορά μέτρο της ομοιότητας μεταξύ των προτύπων. Με βάση αυτό το μέτρο γίνεται η καταχώρηση τους στο ίδιο ή σε διαφορετικό cluster. Το πόσο επιτυχημένο θεωρείται το αποτέλεσμα της συσταδοποίησης εξαρτάται από τα κριτήρια που θα χρησιμοποιηθούν για το διαχωρισμό των στοιχείων σε clusters. Η σωστή επιλογή αυτών των κριτηρίων είναι πολύ σημαντικό ζήτημα.

Το στάδιο της συσταδοποίησης είναι το κυρίως μέρος της διαδικασίας, και όπως είπαμε μπορεί να πραγματοποιηθεί με πολλούς διαφορετικούς αλγορίθμους, καθένας από τους οποίους μπορεί να έχει διαφορετικό αποτέλεσμα, είτε αυστηρό είτε ασαφές. Το μέτρο ομοιότητας που επιλέχτηκε στο προηγούμενο βήμα θα χρησιμοποιηθεί από τον αλγόριθμο που επιλέγεται. Οι αλγόριθμοι clustering είναι πολλοί και στηρίζονται σε διαφορετικές τεχνικές. Η επιλογή εξαρτάται από τη μορφή δεδομένων και από το χρήστη. Οι κύριες κατηγορίες των αλγορίθμων συσταδοποίησης είναι δύο, οι ιεραρχικές και οι διαμεριστικές.

Οι ιεραρχικοί αλγόριθμοι προσπαθούν να δημιουργήσουν μια ιεραρχία μεταξύ των σημείων που προορίζονται για ομαδοποίηση. Δημιουργούν ένα δενδρόγραμμα που υποδηλώνει το μέγεθος και τον αριθμό των clusters που δημιούργησαν. Κάθε κόμβος του δέντρου έχει παιδιά τα σημεία που συγχωνεύτηκαν στην ίδια ομάδα. Ανάλογα με αν βρίσκονται κοντά ή μακριά από τη ρίζα προκύπτουν λίγες ομάδες με πολλά σημεία ή πολλές ομάδες με λίγα σημεία αντίστοιχα. Οι ιεραρχικοί αλγόριθμοι χωρίζονται σε συσσωρευτικούς και διαιρετικούς. Οι συσσωρευτικοί ξεκινούν θεωρώντας ότι κάθε σημείο είναι από μόνο του ένα cluster που περιέχει μόνο τον εαυτό του και στη συνέχεια πραγματοποιούν συγχωνεύσεις. Οι διαιρετικοί αλγόριθμοι λειτουργούν αντίστροφα. Θεωρούν ότι αρχικά υπάρχει ένα cluster που περιέχει όλα τα σημεία και στη συνέχεια το διαιρούν σε μικρότερα.

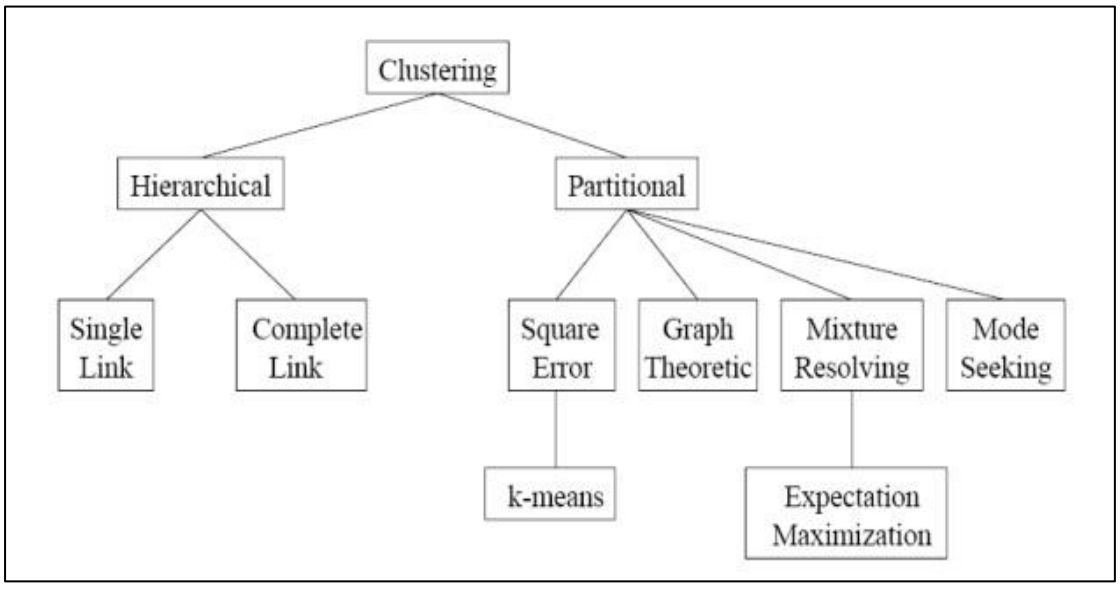
Οι διαμεριστικοί αλγόριθμοι χωρίζουν τα δεδομένα από την αρχή σε ένα δεδομένο αριθμό ομάδων και έπειτα βελτιστοποιούν το αποτέλεσμα. Και αυτοί χωρίζονται σε περαιτέρω κατηγορίες. Οι αλγόριθμοι που είναι βασισμένοι στην πυκνότητα δημιουργούν ομάδες με βάση την πυκνότητα των αντικειμένων στο χώρο. Ένα σημείο που ανήκει σε κάποια ομάδα θα πρέπει να έχει στη γειτονιά του (ορίζεται η ακτίνα της γειτονιάς σημείου) ένα συγκεκριμένο αριθμό από άλλα σημεία.

Μια άλλη κατηγοριοποίηση των clustering αλγορίθμων είναι σε αυστηρούς και ασαφείς. Οι αυστηροί αλγόριθμοι θεωρούν ότι τα σημεία ανήκουν κατά απόλυτο τρόπο στις ομάδες τους. Αυτό πρακτικά σημαίνει ότι δεν μπορεί ένα σημείο να βρίσκεται ταυτόχρονα σε πάνω από μια ομάδα. Αντίθετα, οι ασαφείς θεωρούν ότι τα σημεία ανήκουν σε όλες τις ομάδες σε κάποιο βαθμό. Διαθέτουν μια συνάρτηση συμμετοχής η οποία μας δίνει το βαθμό συμμετοχής του κάθε σημείου σε κάποια ομάδα. Προφανώς από μια ασαφή ομαδοποίηση μπορεί να προκύψει αυστηρή ομαδοποίηση.

Τέλος υπάρχουν οι αυξητικοί και οι μη αυξητικοί αλγόριθμοι. Στους αυξητικούς, το σύνολο των δεδομένων που τίθεται προς συσταδοποίηση προσέρχεται σταδιακά, ένα-ένα, ή κατά ομάδες. Έτσι, γίνεται ομαδοποίηση χωρίς ο αλγόριθμος να γνωρίζει εκ των προτέρων όλο το σύνολο των δεδομένων (on line προβλήματα clustering). Στους μη αυξητικούς αλγορίθμους, το σύνολο των δεδομένων είναι γνωστό εξ αρχής.

Η αφαίρεση δεδομένων, είναι η διαδικασία η οποία έχει ως αποτέλεσμα μια απλή και συμπαγή αναπαράσταση του συνόλου των δεδομένων. Ο όρος απλή αναπαράσταση μπορεί να εξηγηθεί είτε από την οπτική γωνία της αυτοματοποιημένης ανάλυσης είτε από την οπτική γωνία του ανθρώπου. Στην πρώτη περίπτωση, το επιθυμητό για τα δεδομένα είναι να αναπαρίστανται με τέτοιο σαφή και απλό τρόπο, ώστε μια περαιτέρω υπολογιστική επεξεργασία να είναι εξίσου εφικτή. Στη δεύτερη περίπτωση, η πιο απλή αναπαράσταση δεδομένων τα κάνει πιο κατανοητά στους ειδικούς που πρόκειται να τα επεξεργαστούν και να εξάγουν συμπεράσματα. Συνήθως, η αφαίρεση δεδομένων στο clustering είναι μια συνοπτική αναπαράσταση κάθε cluster μέσω κάποιου αντιπροσώπου-πρωτότυπου στοιχείου που καλείται κεντροειδές (centroid).

Τέλος, στο στάδιο της αξιολόγησης του αποτελέσματος ελέγχεται η εγκυρότητα των ομάδων. Εδώ εξετάζεται αν το τελικό αποτέλεσμα του αλγορίθμου είναι επιτυχές, και επομένως ο αλγόριθμος είναι αξιόπιστος. Πρακτικά εξετάζεται αν οι ομάδες είναι αντιπροσωπευτικές σε σχέση με τα σημεία που έπρεπε να ομαδοποιηθούν, αν τα σημεία τελικά τοποθετήθηκαν στις κατάλληλες ομάδες κ.ο.κ. Η αξιολόγηση συνήθως γίνεται συγκρίνοντας τη ληφθείσα δομή με μια δεδομένη εκ των προτέρων δομή. Για την ανάλυση που γίνεται σε αυτό το στάδιο, χρησιμοποιείται ένα συγκεκριμένο κριτήριο βελτιστοποίησης, ανάλογα με το πρόβλημα που αντιμετωπίζεται κάθε φορά.



Εικόνα 7. Κατηγοριοποίηση των τεχνικών clustering

Η συσταδοποίηση επομένως είναι ένα πολυεπίπεδο πρόβλημα βελτιστοποίησης. Ο αποτελεσματικότερος αλγόριθμος συσταδοποίησης και οι σωστές ρυθμίσεις παραμέτρων (όπως η συνάρτηση απόστασης που θα χρησιμοποιηθεί, το όριο πυκνότητας ή ο αναμενόμενος αριθμός των clusters) εξαρτάται από το συγκεκριμένο

πρόβλημα, από τα δεδομένα εισόδου και τον σκοπό που θα εξυπηρετούν τα αποτελέσματα. Η συσταδοποίηση δεν είναι αυτοματοποιημένη εργασία, αλλά μια επαναληπτική, trial and error διαδικασία μάθησης, που περιλαμβάνει πειραματισμό, αποτυχίες, και αναπροσαρμογές παραμέτρων ωσότου να επιτευχθεί ένα αποδεκτό αποτέλεσμα.

Στη συνέχεια παρουσιάζεται το θεωρητικό υπόβαθρο των αλγορίθμων συσταδοποίησης που χρησιμοποιήθηκαν στην εργασία.

2.3.2 Χάρτης Αυτό-οργάνωσης (SOM)

Ο Χάρτης Αυτό-Οργάνωσης (SOM) ή Χάρτης Αυτό-Οργάνωσης Χαρακτηριστικών (SOFM) ή Χάρτης Kohonen (Kohonen Map) είναι ένα είδος τεχνητού νευρωνικού δικτύου που εκπαιδεύεται με μη-επιβλεπόμενη μάθηση και παράγει μια διακριτή, μικρών διαστάσεων (συνήθως δισδιάστατη) αναπαράσταση του χώρου των δεδομένων εισόδου, τον χάρτη.

Αυτό που ξεχωρίζει τα SOMs από τα υπόλοιπα νευρωνικά δίκτυα είναι ότι χρησιμοποιούν μια συνάρτηση γειτονιάς για να διατηρούν τις τοπολογικές ιδιότητες του χώρου εισόδου. Αυτό κάνει τα SOMs χρήσιμα στην οπτικοποίηση πολυδιάστατων δεδομένων.

Ένα SOM έχει δύο λειτουργίες, την εκπαίδευση (training) και τη χαρτογράφηση (mapping). Κατά την εκπαίδευση φτιάχνεται ο χάρτης χρησιμοποιώντας παραδείγματα εισόδου, με μια διαδικασία ανταγωνισμού. Στην χαρτογράφηση, κάθε νέο διάνυσμα εισόδου ταξινομείται αυτόματα.

Το SOM αποτελείται από συστατικά που λέγονται κόμβοι ή νευρώνες. Κάθε κόμβος έχει ένα διάνυσμα βάρους, με διάσταση ίδια με τα δεδομένα εισόδου, και μια θέση στο χάρτη. Οι πιο συνηθισμένες διατάξεις κόμβων είναι οι δισδιάστατες διατάξεις σε εξαγωνικό, σε ορθογώνιο, ή ακόμα και σε τοροειδές πλέγμα.

Το δίκτυο SOM βασίζεται στην ανταγωνιστική μάθηση. Οι νευρώνες εξόδου του δικτύου ανταγωνίζονται μεταξύ τους για το δικαίωμα ενεργοποίησης, με αποτέλεσμα μόνο ένας νευρώνας εξόδου, ή ένας νευρώνας ανά ομάδα, να είναι ενεργός ανά πάσα στιγμή. Ο νευρώνας εξόδου που νικάει στον ανταγωνισμό (νικητής νευρώνας) απολαμβάνει το καθεστώς «winner takes all».

Σε ένα αυτό-οργανούμενο χάρτη, οι νευρώνες τοποθετούνται σε στους κόμβους ενός πλέγματος το οποίο είναι συνήθως μονοδιάστατο ή δισδιάστατο. Υψηλότερης διαστατικότητας χάρτες είναι δυνατόν να δημιουργηθούν, αλλά δεν χρησιμοποιούνται ιδιαίτερα. Οι νευρώνες συντονίζονται επιλεκτικά σε διάφορα πρότυπα εισόδου (ερεθίσματα) ή κλάσεις προτύπων εισόδου, κατά την πορεία μιας διαδικασίας ανταγωνιστικής μάθησης. Οι θέσεις των νευρώνων που συντονίζονται με αυτό τον τρόπο (δηλαδή οι νικητές νευρώνες) διατάσσονται ο ένας σε σχέση με τον άλλο με τέτοιο τρόπο, έτσι ώστε να δημιουργείται ένα λογικό σύστημα συντεταγμένων για διαφορετικά χαρακτηριστικά εισόδου πάνω στο πλέγμα. Συνεπώς, ένας αυτό-οργανούμενος χάρτης χαρακτηρίζεται από το σχηματισμό ενός τοπογραφικού χάρτη αποτελούμενου από τα πρότυπα εισόδου, στον οποίο οι χωρικές θέσεις των νευρώνων

στο πλέγμα είναι ενδεικτικές των εσωτερικών στατιστικών χαρακτηριστικών που περιέχονται στα πρότυπα εισόδου.

Ο κύριος στόχος ενός SOM είναι να μετασχηματίζει ένα πρότυπο εισερχόμενου σήματος, τυχαίας διάστασης, σε ένα διακριτό χάρτη μιας ή δύο διαστάσεων και να εκτελεί αυτό το μετασχηματισμό προσαρμοστικά, με κάποιο τοπολογικά διατεταγμένο τρόπο.

Κάθε πρότυπο εισόδου που παρουσιάζεται στο δίκτυο αποτελείται τυπικά από μια «τοπικοποιημένη» περιοχή ή «σημείο» δραστηριότητας πάνω σε ένα «ήσυχο» υπόβαθρο. Η θέση και η φύση ενός τέτοιου σημείου συνήθως μεταβάλλεται από το ένα στιγμιότυπο του προτύπου εισόδου στο επόμενο. Συνεπώς, όλοι οι νευρώνες του δικτύου θα πρέπει να εκτίθενται σε επαρκή αριθμό διαφορετικών στιγμιότυπων του προτύπου εισόδου για να διασφαλιστεί ότι η διαδικασία αυτό-οργάνωσης θα μπορέσει να αναπτυχθεί σωστά.

Ο αλγόριθμος που είναι υπεύθυνος για το σχηματισμό του αυτό-οργανούμενου χάρτη ξεκινά αρχικοποιώντας τα συναπτικά βάρη στο δίκτυο. Αυτό μπορεί να γίνει αναθέτοντας τους μικρές τιμές, επιλεγμένες από μια γεννήτρια τυχαίων αριθμών. Κατ' αυτό τον τρόπο, δεν επιβάλλεται κάποια αρχική σειρά στο χάρτη χαρακτηριστικών. Αφού το δίκτυο αρχικοποιηθεί σωστά, υπάρχουν τρεις σημαντικές διαδικασίες που εμπλέκονται στο σχηματισμό του αυτό-οργανουμενου χάρτη:

1. Ανταγωνισμός. Για κάθε πρότυπο εισόδου, οι νευρώνες του δικτύου υπολογίζουν τις αντίστοιχες τιμές μιας συνάρτησης διάκρισης. Αυτή η συνάρτηση διάκρισης παρέχει τη βάση για τον ανταγωνισμό μεταξύ των νευρώνων. Ο συγκεκριμένος νευρώνας με τη μεγαλύτερη τιμή στη συνάρτηση διάκρισης δηλώνεται νικητής του ανταγωνισμού.
2. Συνεργασία. Ο νικητής νευρώνας καθορίζει τη χωρική θέση μιας τοπολογικής γειτονιάς διεγερμένων νευρώνων, παρέχοντας έτσι τη βάση για συνεργασία μεταξύ τέτοιων γειτονικών νευρώνων.
3. Προσαρμογή Συναπτικών Βαρών. Αυτό ο τελευταίος μηχανισμός επιτρέπει στους διεγερμένους νευρώνες να αυξάνουν τις τιμές της συνάρτησης διάκρισης σε σχέση με το πρότυπο εισόδου μέσω κατάλληλων προσαρμογών που εφαρμόζονται στα συναπτικά βάρη τους. Οι προσαρμογές που γίνονται είναι τέτοιες ώστε η απόκριση του νικητή νευρώνα στην επόμενη εφαρμογή ενός παρόμοιου προτύπου εισόδου να είναι βελτιωμένη.

Η Διαδικασία Ανταγωνισμού

Έστω m η διάσταση του χώρου εισόδου (δεδομένων). Έστω επίσης ότι ένα πρότυπο εισόδου (διάνυσμα) που επιλέγεται τυχαία από την είσοδο συμβολίζεται ως :

$$x = [x_1, x_2, \dots, x_m]^T \quad (1)$$

Το διάνυσμα συναπτικών βαρών κάθε νευρώνα του δικτύου έχει την ίδια διάσταση με το χώρο εισόδου. Έστω ότι το διάνυσμα συναπτικών βαρών του νευρώνα j συμβολίζεται ως :

$$w_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^T, j = 1,2, \dots, l \quad (2)$$

όπου l ο συνολικός αριθμός νευρώνων του δικτύου. Για να βρούμε τη βέλτιστη ταύτιση του διανύσματος εισόδου **x** με τα διανύσματα συναπτικών βαρών $\mathbf{w_j}$,

συγκρίνουμε τα εσωτερικά γινόμενα $\mathbf{w_j^T x}$ για j = 1,2,…,l και επιλέγουμε το μεγαλύτερο. Αυτή η μέθοδος υποθέτει ότι εφαρμόζεται το ίδιο κατώφλι σε όλους τους νευρώνες (το κατώφλι είναι το αρνητικό της πόλωσης). Έτσι, επιλέγοντας το νευρώνα με το μεγαλύτερο εσωτερικό γινόμενο $\mathbf{w_j^T x}$ ουσιαστικά θα έχουμε καθορίσει τη θέση όπου πρόκειται να κεντραριστεί η τοπολογική γειτονιά των διεγερμένων νευρώνων.

Είναι γνωστό, ότι το κριτήριο βέλτιστης ταύτισης, βάσει μεγιστοποίησης το εσωτερικού γινομένου $\mathbf{w_j^T x}$ ειναι μαθηματικά ισοδύναμο με την ελαχιστοποίηση της Ευκλείδειας απόστασης μεταξύ των διανυσμάτων **x** και $\mathbf{w_j}$, υπό τον όρο ότι το $\mathbf{w_j}$ έχει μοναδιαίο μήκος για όλα τα j. Εάν χρησιμοποιήσουμε το δείκτη **i(x)** για το νευρώνα που ταιριάζει καλύτερα με το διάνυσμα εισόδου **x**, μπορούμε κατόπιν να καθορίσουμε το **i(x)** εφαρμόζοντας της ακόλουθη συνθήκη, η οποία συνοψίζει και την ουσία της διαδικασίας ανταγωνισμού μεταξύ των νευρώνων:

$$i(x) = \arg\min\|x - w_j\|, \qquad j \in A \ (3)$$

όπου Α το πλέγμα των νευρώνων. Από την (3), το i(x) είναι το σημείο εστίασης της προσοχής επειδή θέλουμε να βρούμε την ταυτότητα του νευρώνα i. Ο Συγκεκριμένος νευρώνας i που ικανοποιεί αυτή τη συνθήκη αποκαλείται νευρώνας βέλτιστης ταύτισης, ή νευρώνας νικητής για το διάνυσμα εισόδου **x.**

Η Διαδικασία Συνεργασίας

Έστω ότι το $h_{j,i}$ συμβολίζει την τοπολογική γειτονιά που είναι κεντραρισμένη γύρων από το νικητή νευρώνα i και περικλείει ένα σύνολο διεγερμένων (συνεργαζόμενων) νευρώνων, ένα τυπικό δείγμα των οποίων συμβολίζεται ως j. Έστω επίσης ότι το $d_{j,i}$ συμβολίζει την πλευρική απόσταση μεταξύ του νικητή νευρώνα j και του διεγερμένου νευρώνα j. Τότε μπορούμε να υποθέσουμε ότι η τοπολογική γειτονιά $h_{j,i}$ είναι μια μονοκόρυφη συνάρτηση της πλευρικής απόστασης $d_{j,i}$, τέτοια ώστε να ικανοποιεί δυο απαιτήσεις :

1. Η τοπολογική γειτονιά $h_{j,i}$ είναι συμμετρική γύρω από το μέγιστο σημείο που ορίζεται από $d_{j,i} = 0$. Κοινώς, αποκτά τη μέγιστη τιμή της στο νικητή νευρώνα i για το οποίο η απόσταση $d_{j,i}$ είναι μηδέν.
2. Το πλάτος της τοπολογικής γειτονιάς $h_{j,i}$ μειώνεται μονοτονικά με την αύξηση της πλευρικής απόστασης $d_{j,i}$, φθίνοντας στο μηδέν για $d_{j,i} \rightarrow \infty$. Αυτή είναι μια αναγκαία συνθήκη για τη σύγκλιση.

Μια καλή επιλογή του $h_{j,i}$ που ικανοποιεί αυτές τις απαιτήσεις είναι η Γκαουσιανή συνάρτηση:

$$h_{j,i(x)} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right), j \in A \qquad (4)$$

η οποία είναι αναλλοίωτη μετατόπισης, δηλαδή ανεξάρτητη από τη θέση του νικητή νευρώνα. Η παράμετρος σ είναι το «ενεργό» εύρος της τοπολογικής γειτονιάς. Μετρά το βαθμό στον οποίο οι διεγερμένοι νευρώνες στην κοντινή περιοχή του νικητή νευρώνα συμμετέχουν στη διαδικασία μάθησης. Η χρήση μιας Γκαουσιανού τύπου τοπολογικής γειτονιάς κάνει τον αλγόριθμο SOM να συγκλίνει πιο γρήγορα απ’ ότι μια ορθογώνια τοπολογική γειτονιά.

Για να υπάρξει συνεργασία μεταξύ γειτονικών νευρώνων, είναι αναγκαίο η τοπολογική γειτονιά του $h_{j,I}$ να εξαρτάται από τη πλευρική απόσταση $d_{j,I}$ μεταξύ του νικητή νευρώνα i και του διεγερμένου νευρώνα j στο χώρο εξόδου, και όχι μόνο από κάποιο μέτρο απόστασης στον αρχικό χώρο εισόδου.

Αυτό ακριβώς δείχνει η Εξ.(4). Στην περίπτωση ενός μονοδιάστατου πλέγματος το $d_{j,i}$ είναι ένας ακέραιος ίσος με |j-i|. Στην περίπτωση δισδιάστατου πλέγματος, ορίζεται ως

$$d_{j,i}^2 = \|r_j - r_i\|^2 \qquad (5)$$

όπου το διακριτό διάνυσμα $\mathbf{r_j}$ ορίζει τη θέση του διεγερμένου νευρώνα j και το $\mathbf{r_i}$ τη θέση του νικητή νευρώνα i, αμφότερες εκ των οποίων μετριούνται στο διακριτό χώρο εξόδου.

Ένα άλλο μοναδικό χαρακτηριστικό του αλγόριθμου SOM είναι ότι το μέγεθοςτης τοπολογικής γειτονίας επιτρέπεται να συρρικνώνεται με το χρόνο. Αυτή η απαίτηση ικανοποιείται κάνοντας το εύρος σ της τοπολογικής συνάρτησης γειτονιάς $h_{j,I}$ να μειώνεται με το χρόνο. Μια δημοφιλής επιλογή για την εξάρτηση του σ από τον διακριτό χρόνο n είναι η εκθετική μείωση που περιγράφεται από την :

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \quad n = 0,1,2,\ldots \qquad (6)$$

όπου σ_0 είναι η τιμή του σ κατά την έναρξη του αλγορίθμου SOM και τ_1 είναι μια σταθερά χρόνου επιλεγόμενη από το σχεδιαστή. Αντίστοιχα, η συνάρτηση τοπολογικής γειτονιάς λαμβάνει μια δική της, μεταβαλλόμενη στο χρόνο μορφή, όπως υποδεικνύει η

$$h_{j,i(x)}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right) \quad n = 0,1,2,\ldots \qquad (7)$$

όπου το σ(n) ορίζεται από την Εξ.(6). Έτσι καθώς αυξάνεται ο διακριτός χρόνος n (δηλαδή ο αριθμός των επαναλήψεων), το εύρος σ(n) μειώνεται με εκθετικό ρυθμό και η τοπολογική γειτονιά συρρικνώνεται ανάλογα. Ωστόσο,, θα πρέπει να επισημάνουμε ότι η συνάρτηση γειτονιάς τελικά θα έχει και πάλι τιμή μονάδας για το νικητή νευρώνα i, εφόσον η απόσταση $d_{j,i}$ για το νευρώνα j υπολογίζεται στο χώρο του πλέγματος και συγκρίνεται με το νικητή νευρώνα i.

Η Διαδικασία Προσαρμογής

Η προσαρμοστική διαδικασία των συναπτικών βαρών είναι η τελευταία φάση του αυτό-οργανούμενου σχηματισμού ενός χάρτη χαρακτηριστικών. Για να είναι αυτό-οργανούμενο το δίκτυο, το διάνυσμα συναπτικών βαρών $\mathbf{w_j}$ του νευρώνα j του δικτύου πρέπει να προσαρμόζεται σε σχέση με το διάνυσμα εισόδου **x.**

Τροποποιούμε την Χεμπιανή υπόθεση συνπεριλαμβάνοντας τον όρο λησμόνησης $g(y_j)\mathbf{w_j}$, όπου $g(y_i)$ είναι κάποια βαθμωτή συνάρτηση της απόκρισης y_i. Η μόνη απαίτηση που επιβάλλεται στην συνάρτηση $g(y_i)$ είναι ο σταθερός όρος στο ανάπτυγμα Taylor της $g(y_i)$ να είναι μηδέν, έτσι ώστε να μπορούμε να γράψουμε

$$g(y_i) = 0 \quad \gamma\iota\alpha \quad y_i = 0 \qquad (8)$$

Δοθείσας μιας τέτοιας συνάρτησης, μπορούμε κατόπιν να εκφράσουμε την αλλαγή στο διάνυσμα βαρών του νευρώνα j ως εξής:

$$\Delta w_j = \eta y_i \boldsymbol{x} - g(y_i) w_j \qquad (9)$$

όπου η είναι η παράμετρος ρυθμού μάθησης του αλγορίθμου. Ο πρώτος όρος στη δεξιά πλευρά της Εξ.(9) είναι ο «Χεμπιανός» όρος, ενώ ο δεύτερος είναι ο όρος λησμόνησης. Για την ικανοποίηση της Εξ.(8) επιλέγουμε μια γραμμική συνάρτηση για την $g(y_i)$.

$$g(y_i) = \eta y_i \qquad (10)$$

Για ένα νικητή νευρώνα i(x), μπορούμε να απλοποιήσουμε περισσότερο την Εξ.(9) θέτοντας την απόκριση

$$y_i = h_{j,i(x)} \qquad (11)$$

Χρησιμοποιώντας τις Εξ.(10) και (11) παίρνουμε

$$\Delta \boldsymbol{w}_j = \eta h_{j,i(x)} (\boldsymbol{x} - \boldsymbol{w}_j) \qquad (12)$$

Τέλος, χρησιμοποιώντας φορμαλισμό διακριτού χρόνου, με δεδομένο το διάνυσμα συναπτικών βαρών $\mathbf{w}_j(n)$ του νευρώνα j τη χρονική στιγμή n, ορίζουμε το ενημερωμένο διάνυσμα βαρών $\mathbf{w}_j(n+1)$ τη χρονική στιγμή n+1 ως:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n) \left(\boldsymbol{x}(n) - \boldsymbol{w}_j(n) \right) \qquad (13)$$

σχέση η οποία εφαρμόζεται σε όλους τους νευρώνες του πλέγματος που βρίσκονται μέσα στην τοπολογική γειτονιά του νικητή νευρώνα. Η Εξ.(13) έχει ως αποτέλεσμα την μετακίνηση του διανύσματος των συναπτικών βαρών $\mathbf{w}_i$ του νικητή νευρώνα i προς το διάνυσμα εισόδου $\mathbf{x}$. Μετά από επαναλαμβανόμενες παρουσιάσεις των δεδομένων εκπαίδευσης, τα διανύσματα συναπτικών βαρών τείνουν να ακολουθούν την κατανομή των διανυσμάτων εισόδου λόγω της ενημέρωσης που λαμβάνει χώρα στη γειτονιά. Συνεπώς, ο αλγόριθμος οδηγεί σε μια τοπολογική διάταξη του χάρτη χαρακτηριστικών στο χώρο εισόδου, υπό την έννοια ότι οι νευρώνες που είναι γειτονικοί στο πλέγμα θα τείνουν να έχουν παρόμοια διανύσματα συναπτικών βαρών.

Η Εξ.(13) είναι ο επιθυμητός τύπος για τον υπολογισμό των συναπτικών βαρών του χάρτη χαρακτηριστικών. Επιπρόσθετα με αυτή την εξίσωση, ωστόσο, χρειαζόμαστε τον ευρετικό κανόνα της Εξ.(7) για την επιλογή της συνάρτησης γειτονιάς $h_{j,i(x)}(n)$.

Η παράμετρος ρυθμού μάθησης η(n) θα πρέπει επίσης να είναι μεταβαλλόμενη στο χρόνο, όπως υποδεικνύεται στην Εξ.(13). Συγκεκριμένα θα πρέπει να ξεκινά από κάποια αρχική τιμή η_0 και κατόπιν να μειώνεται βαθμιαία με το χρόνο n. Αυτή η απαίτηση μπορεί να ικανοποιηθεί με το ακόλουθο ευρετικό κανόνα:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \qquad n = 0,1,2,\ldots \qquad (14)$$

όπου τ_2 είναι μια άλλη σταθερά χρόνου του αλγορίθμου SOM. Σύμφωνα με αυτό το δεύτερο ευρετικό κανόνα, η παράμετρος ρυθμού μάθησης φθίνει εκθετικά με το χρόνο. Παρότι οι εκθετικά φθίνουσες εξισώσεις (6) και (14) για το εύρος της συνάρτησης γειτονιάς και την παράμετρο ρυθμού μάθησης αντίστοιχα μπορεί να μην είναι βέλτιστες, είναι συνήθως επαρκείς για το σχηματισμό του χάρτη χαρακτηριστικών με αυτό-οργανούμενο τρόπο.

2.3.3 K-means

Ο αλγόριθμος K-means είναι ένας εξαιρετικά δημοφιλής αλγόριθμος συσταδοποίησης. Αυτό οφείλεται στο ότι είναι σχετικά απλός στην υλοποίηση αλλά ταυτόχρονα πολύ αποτελεσματικός σε απόδοση. Ο αλγόριθμος μπορεί να περιγραφεί ως εξής:

Δοθέντος ενός συνόλου N παρατηρήσεων, ζητείται ο κωδικοποιητής C που αντιστοιχίζει αυτές τις παρατηρήσεις στα K clusters με τέτοιο τρόπο ώστε, μέσα σε κάθε συστάδα, ό μέσος όρος του μέτρου ανομοιότητας των αντιστοιχισμένων παρατηρήσεων ως προς το μέσο (mean) του cluster να ελαχιστοποιείται.

Έστω ότι το $\{x_i\}_{i=1}^{N}$ συμβολίζει ένα σύνολο πολυδιάστατων παρατηρήσεων το οποίο πρόκειται να διαμεριστεί σε ένα προτεινόμενο σύνολο K clusters, όπου το K είναι μικρότερο από τον αριθμό παρατηρήσεων N. έστω επίσης η σχέση:

$$j = C(i), \qquad i = 1,2, \dots, N \qquad (1)$$

που συμβολίζει ένα αντιστοιχιστή «πολλά-προς-ενα», που αποκαλείται κωδικοποιητής, ο οποίος αντιστοιχίζει την i-οστή παρατήρηση x_i στο j-οστό cluster σύμφωνα με έναν κανόνα που θα ορίσουμε. Για να υλοποιηθεί αυτή την κωδικοποίηση, χρειάζεται ένα μέτρο ομοιότητας μεταξύ κάθε ζεύγους διανυσμάτων x_i και $x_{i'}$, το οποίο συμβολίζεται ως $d(x_i,x_{i'})$. Όταν το μέτρο $d(x_i,x_{i'})$ είναι επαρκώς μικρό, αμφότερα τα x_i και $x_{i'}$ αντιστοιχίζονται στο ίδιο cluster. Διαφορετικά, αντιστοιχίζονται σε διαφορετικά clusters.

Για τη βελτιστοποίηση της διαδικασίας συσταδοποίησης, εισάγουμε τη συνάρτηση κόστους :

$$J(C) = \frac{1}{2}\sum_{j=1}^{K}\sum_{C(i)=j}\sum_{C(i')=j} d(x_i, x_{i'}) \qquad (2)$$

Για ένα προκαθορισμένο K, το ζητούμενο είναι να βρεθεί ο κωδικοποιητής C(i)=j για τον οποίο η συνάρτηση κόστους J(C) ελαχιστοποιείται. Επισημαίνεται ότι ο κωδικοποιητής C είναι άγνωστος, και σε αυτό οφείλεται η λειτουργική εξάρτηση ης συνάρτησης κόστους J από το C.

Στον αλγόριθμο K-means, χρησιμοποιεί το τετράγωνο της Ευκλείδειας νόρμας για τον ορισμό του μέτρου ομοιότητας μεταξύ των παρατηρήσεων x_i και $x_{i'}$, όπως αποδεικνύει η σχέση

$$d(x_i, x_{i'}) = \|x_i - x_{i'}\| \qquad (3)$$

Άρα αντικαθιστώντας την Εξ.(3) στην Εξ.(2) προκύπτει :

$$J(C) = \frac{1}{2}\sum_{j=1}^{K}\sum_{C(i)=j}\sum_{C(i')=j} \|x_i - x_{i'}\| \qquad (4)$$

Δυο σημαντικά σημεία είναι :

1. Το τετράγωνο της Ευκλείδειας απόστασης μεταξύ των παρατηρήσεων x_i και $x_{i'}$ είναι συμμετρικό, δηλαδή:

$$\|x_i - x_{i'}\|^2 = \|x_{i'} - x_i\|^2$$

2. Το εσωτερικό άθροισμα στην Εξ.(4) ερμηνεύεται ως εξής: Για ένα δεδομένο $x_{i'}$ ο κωδικοποιητής C αντιστοιχίζει στο cluster j όλες τις παρατηρήσεις $x_{i'}$ που είναι πλησιέστερα στην x_i. Εκτός από ένα συντελεστή κλιμάκωσης, το άθροισμα των παρατηρήσεων $x_{i'}$ που αντιστοιχίζεται είναι μια εκτίμηση του μέσου διανύσματος που αφορά το cluster j. Ο εν λόγω συντελεστής κλιμάκωσης είναι $1/N_j$, όπου N_j είναι ο αριθμός των σημείων δεδομένων μέσα στη συστάδα j.

Με βάση αυτά τα σημεία η Εξ.(4) μπορεί να απλοποιηθεί σε :

$$J(C) = \sum_{j=1}^{K} \sum_{C(i)=j} \left\| x_i - \hat{\mu}_j \right\|^2 \qquad (5)$$

όπου το $\hat{\mu}_j$ συμβολίζει το εκτιμώμενο μέσο διάνυσμα που σχετίζεται με τη συστάδα j. Ουσιαστικά, το μέσο $\hat{\mu}_j$ μπορεί να θεωρηθεί κέντρο της συστάδας j.

Για μια ερμηνεία της συνάρτησης κόστους J(C) που ορίζει η Εξ.(5), εκτός από ένα συντελεστή κλιμάκωσης $1/N_j$, το εσωτερικό άθροισμα σε αυτή την εξίσωση είναι μια εκτίμηση της διακύμανσης των παρατηρήσεων που σχετίζονται με το cluster j για ένα δεδομένο κωδικοποιητή C, όπως φαίνεται στην

$$\hat{\sigma}_j^2 = \sum_{C(i)=j} \left\| x_i - \hat{\mu}_j \right\|^2 \qquad (6)$$

Κατά συνέπεια, η συνάρτηση κόστους J(C) μπορεί να αντιμετωπιστεί ως ένα μέτρο της συνολικής διακύμανσης του cluster που προκύπτει από τις αντιστοιχίσεις όλων των N παρατηρήσεων στα K clusters που σχηματίζονται από το κωδικοποιητή C.

Για την ελαχιστοποίηση της συνάρτησης κόστους J(C) χρησιμοποιείται ένας αλγόριθμος επαναληπτικής κατάβασης, κάθε επανάληψη του οποίου χρησιμοποιεί μια διαδικασία βελτιστοποίησης δύο βημάτων. Το πρώτο βήμα χρησιμοποιεί τον κανόνα των πλησιέστερων γειτόνων (Nearest neighbor) της J(C) ως προς το μέσο διάνυσμα $\hat{\mu}_j$ για ένα δεδομένο κωδικοποιητή C. το δεύτερο βήμα ελαχιστοποιεί το εσωτερικό άθροισμα της Εξ.(5) ως προς τον κωδικοποιητή C για ένα δεδομένο μέσο διάνυσμα $\hat{\mu}_j$. Αυτή η επαναληπτική διαδικασία των δυο βημάτων συνεχίζεται μέχρι να επιτευχθεί σύγκλιση.

Επομένως, με μαθηματικούς όρους ο αλγόριθμος K-means εξελίσσεται σε δυο βήματα:

Βήμα 1: Για ένα δεδομένο κωδικοποιητή , η συνολική διακύμανση συστάδας ελαχιστοποιείται ως προς το σύνολο μέσων συστάδας $\{\hat{\mu}_j\}_{j=1}^{K}$. Δηλαδή εκτελείται η ακόλουθη ελαχιστοποίηση:

$$min_{\{\hat{\mu}_j\}_{j=1}^{K}} \sum_{j=1}^{K} \sum_{C(i)=j} \left\| x_i - \hat{\mu}_j \right\|^2 \qquad \textit{για δεδομένο } C$$

Βήμα 2: Αφού υπολογιστούν οι βελτιστοποιημένοι μέσοι των clusters $\{\hat{\mu}_j\}_{j=1}^{K}$ στο βήμα 1, στη συνέχεια βελτιστοποιούμε τον κωδικοποιητή ως εξής:

$$C(i) = \arg min_{1 \le j \le K} \left\| x_i - \hat{\mu}_j \right\|^2$$

Ξεκινώντας από κάποια αρχική επιλογή του κωδικοποιητή C, ο αλγόριθμος εναλλάσσεται μεταξύ αυτών των δυο βημάτων μέχρι να μην υπάρχει περαιτέρω αλλαγή στις αντιστοιχίσεις των συστάδων.

Κάθε ένα από αυτά τα δυο βήματα είναι σχεδιασμένο ώστε να μειώνει τη συνάρτηση κόστους J(C) με το δικό του τρόπο. Άρα η σύγκλιση του αλγορίθμου είναι διασφαλισμένη. Ωστόσο επειδή ο αλγόριθμος δεν διαθέτει ένα γενικό κριτήριο βελτιστότητας, το αποτέλεσμα μπορεί να συγκλίνει σε ένα κάποιο τοπικό ελάχιστο, δίνοντας μια υποβέλτιστη λύση. Παρόλα αυτά ο αλγόριθμος K-means έχει κάποια πρακτικά πλεονεκτήματα:

1. Είναι υπολογιστικά αποτελεσματικός, στο ότι η πολυπλοκότητα του είναι γραμμική ως προς τον αριθμό των clusters.
2. Όταν τα clusters επιδεικνύουν συμπαγή κατανομή στο χώρο δεδομένων, ανακτώνται πιστά από τον αλγόριθμο.

Τέλος, μια χρήσιμη παρατήρηση είναι ότι για την αρχικοποίηση του αλγορίθμου K-means, η συνιστώμενη διαδικασία είναι η εκκίνηση του αλγορίθμου με πολλές διαφορετικές τυχαίες επιλογές για τους μέσους $\{\hat{\mu}_j\}_{j=1}^{K}$ για το προτεινόμενο μέγεθος K και κατόπιν η επιλογή του συγκεκριμένου συνόλου για το οποίο η διπλή άθροιση στην Εξ.(5) λαμβάνει τη μικρότερη τιμή.

2.3.3.1 Εκτίμηση του αριθμού των clusters

Η εκτίμηση του αριθμού των clusters σε ένα σύνολο δεδομένων, η μεταβλητή που συμβολίζεται συνήθως με K, είναι ένα συνηθισμένο πρόβλημα στη συσταδοποίηση δεδομένων, και είναι τελείως ξεχωριστό πεδίο μελέτης από την ίδια τη διαδικασία της συσταδοποίησης.

Πολλοί αλγόριθμοι συσταδοποίησης, μεταξύ αυτών και ο αλγόριθμος K-means που αναφέρθηκε προηγουμένως, απαιτούν να είναι προκαθορισμένος ο αριθμός των clusters K. Η σωστή επιλογή του K είναι συχνά δύσκολη, και μπορεί να εξαρτάται τόσο από το σχήμα και την κλίμακα της κατανομής των στοιχείων του συνόλου στο χώρο δεδομένων, όσο και από τις επιλογές του ίδιου του χρήστη. Επιπλέον, η αύξηση του K χωρίς ποινή, θα μειώνει πάντα το ποσοστό σφάλματος στην τελική ομαδοποίηση. Η ακραία περίπτωση εντοπίζεται όταν κάθε δεδομένο είναι το μοναδικό στο cluster του, δίνοντας συνολικό σφάλμα ίσο με το μηδέν (δηλαδή όταν το K ισούται με τον μέγεθος του συνόλου δεδομένων N). Διαισθητικά επομένως, η βέλτιστη τιμή για το K θα επιτυγχάνει ισορροπία ανάμεσα στη μέγιστη συμπίεση των δεδομένων χρησιμοποιώντας ένα μόνο cluster, και τη μέγιστη ακρίβεια με την ανάθεση κάθε δεδομένου στο δικό του cluster. Αν μια τέτοια τιμή για το K δεν είναι προφανής από τις ιδιότητες του υπό εξέταση συνόλου δεδομένων, θα πρέπει κάπως να βρεθεί.

Κοιτάζοντας τη βιβλιογραφία στο συγκεκριμένο πρόβλημα, υπάρχουν πολλές μέθοδοι για τον υπολογισμό αυτής της τιμής, κλιμακούμενης πολυπλοκότητας. Ο πιο απλός είναι ο λεγόμενος «Κανόνας του Αντίχειρα» (Rule of Thumb), ο οποίος ευθέως θεωρεί πως η βέλτιστη τιμή για το K υπολογίζεται από τη σχέση:

$$K \approx \sqrt{\frac{N}{2}} \ , \qquad N \ o \ \alpha\rho\iota\theta\mu\acute{o}\varsigma \ \tau\omega\nu \ \delta\varepsilon\delta o\mu\acute{\varepsilon}\nu\omega\nu$$

Όπως είναι προφανές, αυτή η μέθοδος δεν λαμβάνει καθόλου υπόψιν τις ιδιαιτερότητες του εκάστοτε συνόλου δεδομένων που εξετάζεται, και είναι τελείως άκαμπτη.

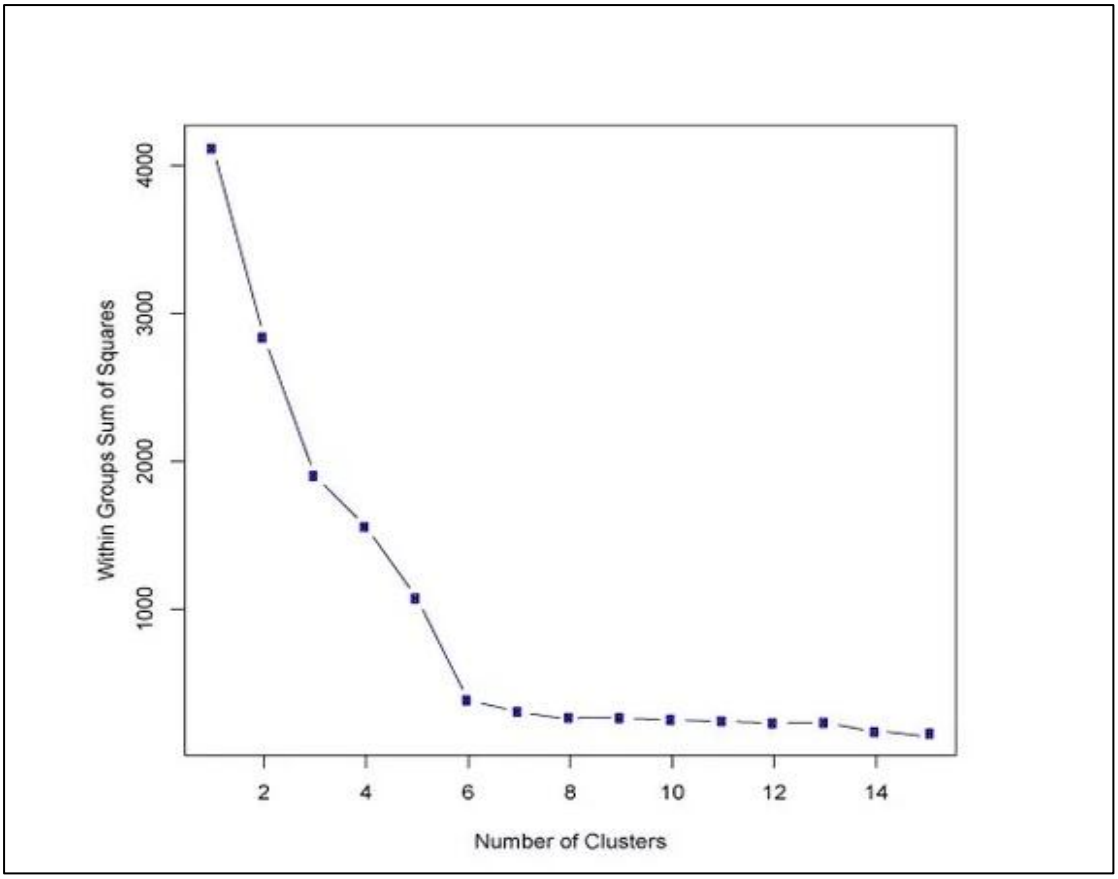
Στο άλλο άκρο, υπάρχουν ιδιαίτερα πολύπλοκες μέθοδοι που εμπλέκουν την έννοια της ευστάθειας ή θεωρητικές κριτήρια όπως τα Akaike information criterion (AIC) , Bayesian information criterion (BIC) και Deviance information criterion (DIC).

Σε αυτή την εργασία χρησιμοποιήθηκε η «μέθοδος του αγκώνα» (Elbow Method), η οποία λειτουργεί ως εξής :

Εκτελούμε τον αλγόριθμο συσταδοποίησης (συγκεκριμένα εδώ τον αλγόριθμο K-means) για πολλές τιμές του K, από πολύ μικρές έως πολύ μεγάλες. Για κάθε εκτέλεση, υπολογίζεται ένας δείκτης εκτίμησης των clusters. Εν προκειμένω, αυτός ο δείκτης είναι το άθροισμα του τετραγώνου των σφαλμάτων (SSE), που ορίζεται ως το άθροισμα των τετραγώνων των αποστάσεων ανάμεσα σε κάθε δεδομένο και τον κεντροειδή του cluster στον οποίο ανήκει. Δηλαδή :

$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} dist(x, c_i)^2$$

Στη συνέχεια αν κανείς σχεδιάσει τη γραφική παράσταση του SSE συναρτήσει του αριθμού των clusters K, θα παρατηρήσει ότι ο δείκτης SSE, δηλαδή το σφάλμα μειώνεται όσο το K αυξάνεται, το οποίο είναι αναμενόμενο με βάση τα όσα αναφέραμε παραπάνω. Θα παρατηρήσει επίσης ότι για κάποιο K, η γραφική παράσταση θα παρουσιάζει μια έντονη γωνία, εξ' ου και το όνομα της μεθόδου. Αυτή η τιμή του K είναι που επιλέγεται ως βέλτιστη για την εκτέλεση του αλγορίθμου. Ακολουθεί ένα παράδειγμα μιας τέτοια γραφικής παράστασης που δείχνει αυτή την απότομη μεταβολή:



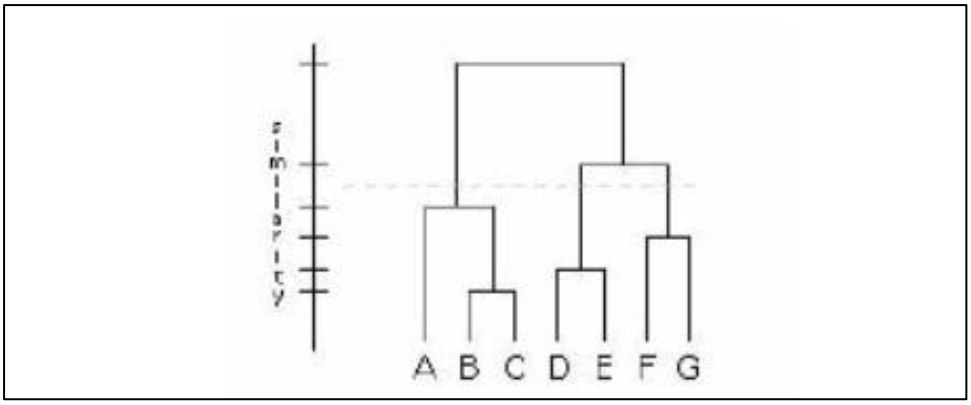
Εικόνα 8. Elbow method

Σε αυτό το παράδειγμα, η τιμή που θα επιλεχθεί είναι K = 6.

Είναι σημαντικό να λαμβάνεται υπόψιν ότι η μέθοδος αυτή είναι μια heuristic, και ως τέτοια μπορεί να κάποιες φορές να μην είναι τόσο αποτελεσματική. Μπορεί το γράφημα να παρουσιάζει παραπάνω από μια γωνίες, μπορεί να μην παρουσιάζει καμία. Παρόλα αυτά, μιας και περιλαμβάνει υπολογισμό του δείκτη SSE για διάφορες τιμές του K, δίνει την δυνατότητα στο χρήστη να κάνει μια καλή, προσωπική επιλογή για το K στις περιπτώσεις εκείνες που ο κανόνας δεν δίνει ξεκάθαρο αποτέλεσμα.

2.3.4 Ιεραρχική Συσταδοποίηση (Hierarchical Clustering)

Στην ιεραρχική συσταδοποίηση τα στιγμιότυπα δίνονται με μορφή δενδρογράμματος. Στα δενδρογράμματα αυτά, επιλέγεται ένα επίπεδο που θα «κλαδευτούν». Το σημείο που θα κλαδευτεί κάποιο δενδρόγραμμα δείχνει τον αριθμό των clusters που θα προκύψουν καθώς και τα σημεία που περιέχει το κάθε cluster. Οι ιεραρχικές τεχνικές είναι είτε συσσωρευτικές (bottom-up), είτε διαιρετικές (top-down). Παρακάτω φαίνεται ένα παράδειγμα δενδρογράμματος κομμένο σε ένα επιλεγμένο επίπεδο. Διαφαίνονται τρία clusters. Το πρώτο περιέχει τα σημεία A, B και C, το δεύτερο τα σημεία D και E, και το τρίτο τα σημεία F και G.

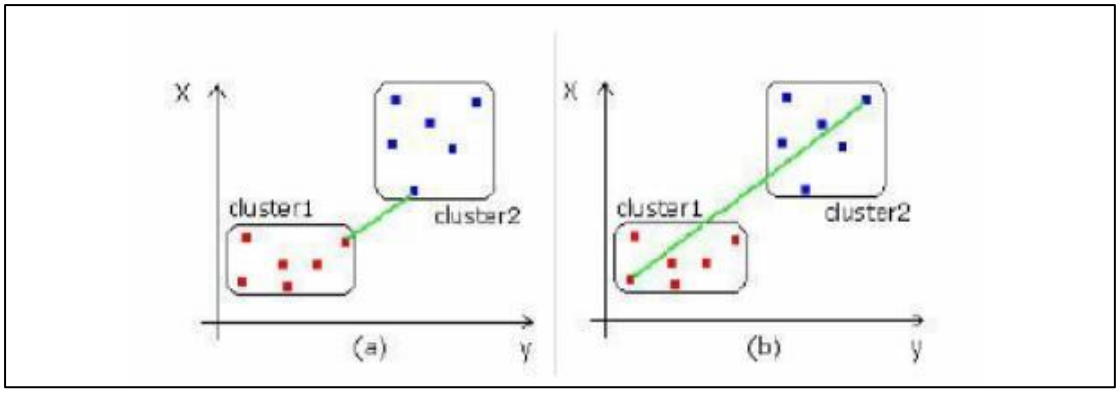


Εικόνα 9. Παράδειγμα δενδρογράμματος

Στη γενική της μορφή, η ιεραρχική συσταδοποίηση λειτουργεί ως εξής:

1. Αρχικά κάθε σημείο θεωρείται σαν μια ομάδα. Ν δηλαδή υπάρχει ένα σύνολο από Ν σημεία τα οποία πρέπει να ομαδοποιηθούν, τότε αρχικά υπάρχουν Ν ομάδες, που η καθεμία περιέχει ένα μόνο σημείο. Μετρώνται οι μεταξύ τους αποστάσεις.
2. Βρίσκεται το πιο κοντινό ζευγάρι ομάδων. Το ζευγάρι αυτό συγχωνεύεται σε ένα. Πλέον υπάρχει μια ομάδα λιγότερη.
3. Υπολογίζονται εκ νέου οι αποστάσεις των ομάδων μεταξύ τους.
4. Επαναλαμβάνονται τα βήματα 2 και 3 έως ότου και τα Ν σημεία να τοποθετηθούν σε μια και μοναδική ομάδα.
5. Τέλος, σχεδιάζεται το αντίστοιχο δενδρόγραμμα και επιλέγεται σε ποιο σημείο θα κλαδευτεί.

Το βήμα 3 μπορεί να πραγματοποιηθεί με διάφορους τρόπους. Στην ιεραρχική συσταδοποίηση απλού συνδέσμου (simple-linkage), θεωρείται ως απόσταση μεταξύ δύο ομάδων η μικρότερη απόσταση μεταξύ όλων των ζευγών των προτύπων με στοιχεία κι από τις δύο ομάδες. Στην συσταδοποίηση ολοκληρωμένου συνδέσμου (complete-linkage), θεωρείται ως απόσταση μεταξύ δύο ομάδων η μεγαλύτερη απόσταση μεταξύ όλων των ζευγών των προτύπων με στοιχεία κι από τις δύο ομάδες. Στην συσταδοποίηση μέσου συνδέσμου (average-linkage) τέλος, θεωρείται ως απόσταση μεταξύ δύο ομάδων η μέση απόσταση μεταξύ όλων των ζευγών των προτύπων με στοιχεία κι από τις δύο ομάδες. Παρακάτω φαίνονται οι αποστάσεις δυο ομάδων για simple-linkage και complete-linkage ομαδοποίηση.



Εικόνα 10. (α) Παράδειγμα απόστασης simple linkage. (b) Παράδειγμα απόστασης complete linkage.

Στα πλεονεκτήματα της ιεραρχικής συσταδοποίησης είναι ότι τερματίζει σχετικά γρήγορα, δεν απαιτεί να είναι γνωστός ο αριθμός των clusters εκ των προτέρων, και έχει ενσωματωμένη ευελιξία χάρις στη δυνατότητα που παρέχουν να επιλέγεται το επίπεδο τομής του δενδρογράμματος. Επίσης, έχουν ευκολία χειρισμού κάθε τύπου μέτρου απόστασης ή ομοιότητας, και μπορούν να εφαρμοστούν σε πολλούς τύπους δεδομένων και όχι μόνο για δεδομένα που περιέχουν ισοτροπικά clusters (π.χ. σε clusters με μορφή αλυσίδας, ομόκεντρους). Τέλος η ιεραρχική συσταδοποίηση παράγει καλές οπτικοποιήσεις των αποτελεσμάτων της κατά τη διάρκεια της εκτέλεσης της.

Τα μειονεκτήματα της είναι ότι διακρίνεται από αοριστία στον ορισμό των κριτηρίων τερματισμού, έχει μεγαλύτερες απαιτήσεις σε υπολογιστική ισχύ (οι αλγόριθμοι σύνδεσης έχουν υπολογιστική πολυπλοκότητα $O(N^2)$), και δεν επιστέφει ποτέ σε ένα ήδη κατασκευασμένο, ενδιάμεσο cluster για να το βελτιώσει.

2.4 Ομοιότητα συνημίτονου (Cosine similarity)

Η ομοιότητα συνημίτονου είναι ένας δείκτης συνάφειας ανάμεσα σε δυο διανύσματα, βασισμένος στη γωνία ανάμεσα στα διανύσματα στο χώρο χαρακτηριστικών τους. Η τιμή του κυμαίνεται στο διάστημα [-1,1], καθώς όπως δηλώνει και το όνομα του είναι ουσιαστικά ένα συνημίτονο. Είναι επομένως ένας δείκτης φοράς και όχι πλάτους/μεγέθους. Δυο διανύσματα με ίδια φορά έχουν δείκτη ομοιότητας συνημίτονου ίσο με 1, δυο διανύσματα κάθετα μεταξύ τους (δηλαδή με γωνία 90^o) έχουν δείκτη ομοιότητας συνημίτονου ίσο με 0, και τέλος δύο διανύσματα διαμετρικά αντίθετα (δηλαδή με γωνία 180^o) έχουν δείκτη ομοιότητας συνημίτονου ίσο με -1. Συνήθως αυτό ο δείκτης χρησιμοποιείται στο θετικό χώρο, όπου το αποτέλεσμα δεσμεύεται στο διάστημα [0,1].

Το συνημίτονο δύο διανυσμάτων μπορεί να βρεθεί ως γνωστόν από τον τύπο του εσωτερικού γινομένου. Δοθέντων των διανυσμάτων A και B, ο δείκτης ομοιότητας συνημίτονου ,cos(θ), υπολογίζεται από τον τύπο :

$$cosine\ similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

όπου προφανώς $A \cdot B$ είναι το εσωτερικό γινόμενο των δυο διανυσμάτων, και $\|A\|\|B\|$ είναι το γινόμενο των μέτρων τους.

Τα όρια που αναφέρθηκαν παραπάνω ισχύουν για οποιοδήποτε αριθμό διαστάσεων, και μάλιστα ο δείκτης εφαρμόζεται κυρίως σε περιπτώσεις πολυδιάστατων διανυσμάτων. Για παράδειγμα, στην εξόρυξη κειμένου, σε κάθε όρο ανατίθεται μια διαφορετική διάσταση, και το αρχείο χαρακτηρίζεται από ένα διάνυσμα, κάθε διάσταση του οποίου δείχνει πόσες φορές περιέχεται στο αρχείο ο αντίστοιχος όρος. Ο δείκτης ομοιότητας συνημίτονου τότε μπορεί να δείξει πόσο σχετικά είναι δυο αρχεία όσον αφορά το περιεχόμενο τους. Είναι προφανές ότι σε αυτή την περίπτωση, ο δείκτης θα είναι στο διάστημα [0,1], αφού οποιαδήποτε διάσταση δεν μπορεί να έχει αρνητική τιμή, μιας και αναφέρεται σε συχνότητα εμφάνισης όρου σε κείμενο. Άρα και η γωνία ανάμεσα σε δυο οποιαδήποτε αρχεία δεν μπορεί να ξεπερνά τις 90^o.

Εάν τα διανύσματα κανονικοποιηθούν αφαιρώντας το μέσο διάνυσμα (δηλαδή $A - \bar{A}$), ο δείκτης ονομάζεται κεντραρισμένη ομοιότητα συνημίτονου και είναι ισοδύναμος με τον Συντελεστή Συσχέτισης Pearson.

Ο δείκτης ομοιότητας συνημίτονου σχετίζεται με την Ευκλείδεια απόσταση ως εξής. Αν $\|A - B\|$ συμβολίζει την Ευκλείδεια απόσταση, παρατηρούμε ότι

$$\|A - B\|^2 = (A - B)^T(A - B) = \|A\|^2 + \|B\|^2 - 2A^TB$$

που για κανονικοποιημένα διανύσματα έχουμε $\|A\|^2 = \|B\|^2 = 1$, άρα

$$\|A - B\|^2 = 2(1 - \cos(A, B))$$

Τέλος, υπάρχει η χαλαρή ομοιότητα συνημίτονου (soft cosine similarity), ένας δείκτης που υπολογίζει ομοιότητα ανάμεσα σε ζεύγη από χαρακτηριστικά. Σε αντίθεση με τον κανονικό δείκτη ομοιότητας συνημίτονου που θεωρεί τα χαρακτηριστικά του μοντέλου διανυσματικού χώρου (VSM) ανεξάρτητα ή τελείως διαφορετικά, ο δείκτης χαλαρής ομοιότητας συνημίτονου προτείνει να υπολογίζεται η ομοιότητα των χαρακτηριστικών του VSM, και αυτό επιτρέπει τη γενίκευση των εννοιών του μέτρου συνημίτονου και της ομοιότητας.

Για παράδειγμα, στο πεδίο της επεξεργασίας φυσικού λόγου, η έννοια της ομοιότητας ανάμεσα σε χαρακτηριστικά είναι αρκετά διαισθητική. Χαρακτηριστικά όπως λέξεις ή n-grams (ακολουθίες n συνεχόμενων όρων από κάποια πρόταση) μπορούν να είναι αρκετά παρεμφερή, αν και τυπικά θεωρούνται τελείως διαφορετικά στο VSM. Παραδείγματος χάριν, οι λέξεις "play" και "game" είναι διαφορετικές και αντιστοιχίζονται σε διαφορετικές διαστάσεις στο VSM, αν και είναι προφανές ότι σχετίζονται σημασιολογικά.

Για τον υπολογισμό του δείκτη χαλαρής ομοιότητας συνημίτονου, ορίζεται ο πίνακας ομοιότητας μεταξύ χαρακτηριστικών s, όπου s_{ij} = similarity(feature-i, feature-j).

Για δυο διανύσματα N διαστάσεων a και b, ο τύπος υπολογισμού είναι:

$$soft\ cosine(a, b) = \frac{\sum_{i,j}^{N} s_{ij} a_i b_j}{\sqrt{\sum_{i,j}^{N} s_{ij} a_i a_j} \sqrt{\sum_{i,j}^{N} s_{ij} b_i b_j}}$$

Αν δεν υπάρχει ομοιότητα ανάμεσα στα χαρακτηριστικά ($s_{ii}=1$,$s_{ij}=0$ για $i \neq j$) ο δείκτης χαλαρής ομοιότητας συνημίτονου ανάγεται στον παραδοσιακό δείκτη ομοιότητας συνημίτονου.

3

ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή υλοποιήθηκε σε Java, στο Eclipse IDE. Ένα τμήμα της, αυτό της συσταδοποίησης υλοποιήθηκε στο περιβάλλον της MATLAB, με χρήση της αντίστοιχης γλώσσας, καθώς παρέχει πολλές ευκολίες στη συγκεκριμένη διαδικασία. Τέλος για τη παρουσίαση των αποτελεσμάτων της εφαρμογής δημιουργήθηκε μια ιστοσελίδα, συνδεδεμένη με μια βάση δεδομένων στην οποία φορτώθηκαν τα αποτελέσματα.

Όπως αναφέρθηκε και στην περίληψη της διπλωματικής εργασίας, ο σκοπός της εφαρμογής είναι η ομαδοποίηση ενός συνόλου ημιδομημένων δεδομένων χρησιμοποιώντας τις πληροφορίες μιας οντολογίας για την επίτευξη μεγαλύτερης ακρίβειας στα αποτελέσματα.

Εν προκειμένω, τα δεδομένα μας ήταν το σενάριο της κινηματογραφικής ταινίας «Bangkok Dangerous (2008)». Αυτό ήταν δοσμένο σε δύο διαφορετικές μορφές. Η πρώτη ήταν το κανονικό σενάριο στην κειμενική του μορφή, όπως θα το έπαιρνε ένας συντελεστής που εργάστηκε στην ταινία, και έχει την εξής μορφή:

BANGKOK DANGEROUS CC&SL REEL 1 - P. 2

SCENE #	FOOTAGE / DESCRIPTION DIALOGUE / M&E	TITLE #	START	END	DURATION	DIALOGUE
8	104+02 INT. CAFÉ - NIGHT WS of people in a café. JOE (V.O.) *My job takes me to a lot of places.* *It's got its down sides.*	2	105+00	108+10	3+10	JOE (V.O.) *My job takes me to a lot of places.*
		3	109+06	112+00	2+10	JOE (CONT'D.) (V.O.) *It's got its down sides.*
9	112+02 FS. Joe sitting alone, at a table in LT BG	4	112+14	116+10	3+12	JOE (CONT'D.) (V.O.) *I sleep alone. I eat alone.*

Εικόνα 11. Τμήμα του σεναρίου

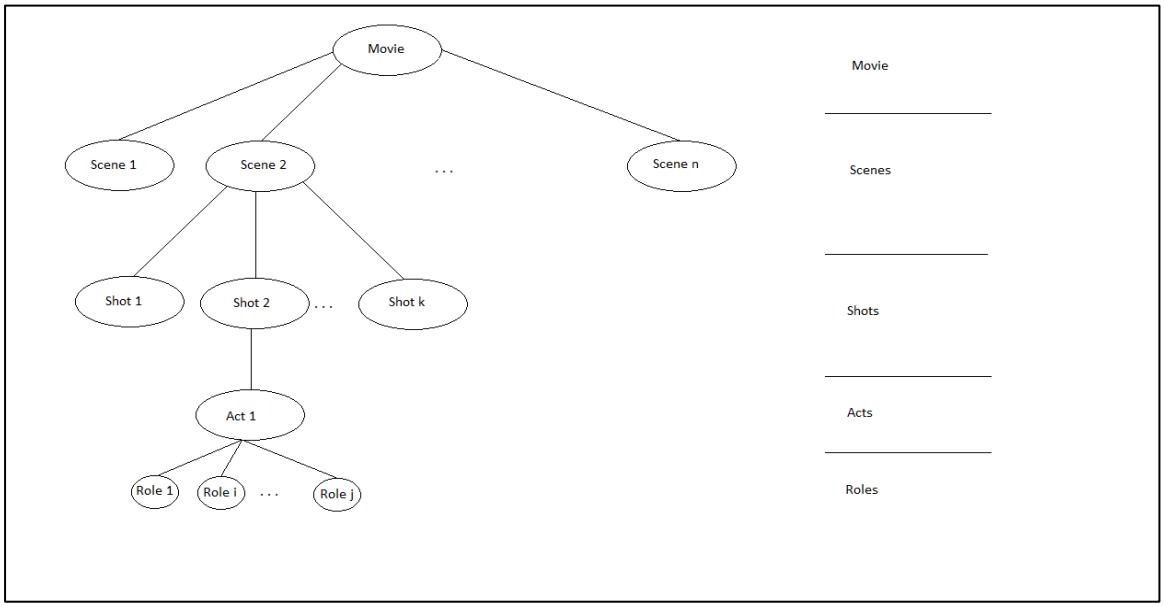
Όπως είναι προφανές, αυτή η μορφή είναι εντελώς ακατάλληλη για την εργασία που έχουμε να κάνουμε, και επομένως χρειάζεται μια προεπεξεργασία πριν προχωρήσουμε. Αυτή είναι η δεύτερη μορφή, και προέκυψε από κάποιο άλλο project. Εμείς σε αυτή την εργασία το θεωρούμε ως έτοιμο δεδομένο. Το δεύτερο αυτό αρχείο είναι ένα ABox βασισμένο στο σενάριο της ταινίας. Περιέχει τα δεδομένα σε μορφή RDF, και συγκεκριμένα σε N-Triples RDF format. Ένα τμήμα του φαίνεται στην επόμενη εικόνα.

```
<http://image.ntua.gr/scriptontology/MOV_12794_SH0742> <http://image.ntua.gr/scriptontology/isNextOf> <http://image.ntua.gr/scriptontology/MOV_12794_SH0741> .
<http://image.ntua.gr/scriptontology/MOV_12794_SC0063> <http://image.ntua.gr/scriptontology/hasPart> <http://image.ntua.gr/scriptontology/MOV_12794_SH0742> .
<http://image.ntua.gr/scriptontology/MOV_12794_SH0742> <http://image.ntua.gr/scriptontology/startTime> "00:30:33.16"^^<http://www.w3.org/2001/XMLSchema#time> .
<http://image.ntua.gr/scriptontology/MOV_12794_SH0742> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://image.ntua.gr/scriptontology/CloseUpShot> .
<http://image.ntua.gr/scriptontology/MOV_12794_SH0742> <http://image.ntua.gr/scriptontology/description> "Kong stops Joe's arm."@en .
<http://image.ntua.gr/scriptontology/MOV_12794_SH0743> <http://image.ntua.gr/scriptontology/isNextOf> <http://image.ntua.gr/scriptontology/MOV_12794_SH0742> .
<http://image.ntua.gr/scriptontology/MOV_12794_SC0063> <http://image.ntua.gr/scriptontology/hasPart> <http://image.ntua.gr/scriptontology/MOV_12794_SH0743> .
<http://image.ntua.gr/scriptontology/MOV_12794_SH0743> <http://image.ntua.gr/scriptontology/startTime> "00:30:33.44"^^<http://www.w3.org/2001/XMLSchema#time> .
```

Εικόνα 12. Τμήμα του ABox

Τέλος, έχουμε ως δεδομένο το αρχείο της οντολογίας, «ontology.owl», βάση της οποίας δημιουργήθηκε το παραπάνω ABox. Αυτή η οντολογία μπορεί να θεωρηθεί ως το πρότυπο με το οποίο οποιοδήποτε σενάριο ταινίας μετατρέπεται σε ένα αντίστοιχο ABox, θεωρώντας βέβαια ιδανικά ότι όλα τα σενάρια ακολουθούν τους ίδιους κανόνες συγγραφής. Η οντολογία θα χρησιμοποιηθεί και σε αυτή την εφαρμογή με τρόπο που θα αναφερθεί πιο μετά.

Με μελέτη των δεδομένων, βλέπουμε ότι ακολουθείται μια συγκεκριμένη αρχιτεκτονική. Η συγκεκριμένη ταινία (και κάθε ταινία) αποτελείται από σκηνές (scenes), οι σκηνές αποτελούνται από πλάνα (shots). Σε κάθε πλάνο μπορεί να περιέχονται ερμηνείες (acts), καθεμία από τις οποίες περιλαμβάνει ένα υποσύνολο των χαρακτήρων (roles) της ταινίας.



Εικόνα 13. Δομή του σεναρίου

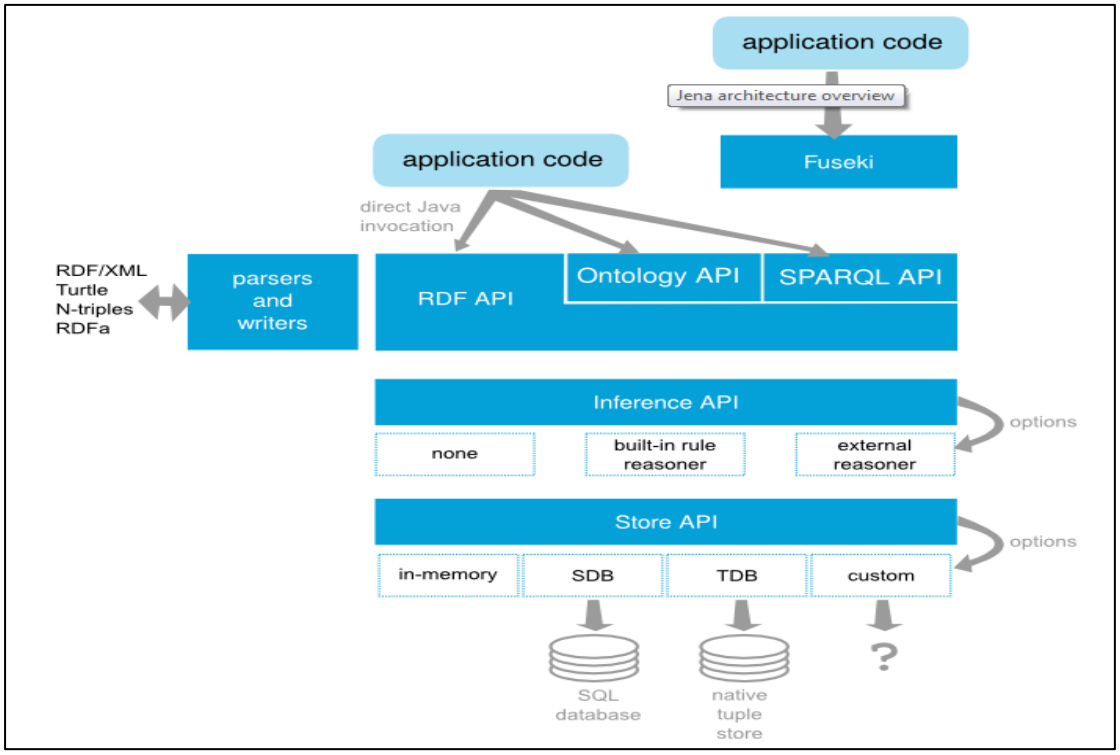
Η ομαδοποίηση που θα πραγματοποιηθεί από την εφαρμογή θα γίνει πάνω στο σύνολο των πλάνων της ταινίας.

Η εφαρμογή αποτελείται από τρία επιμέρους τμήματα. Το πρώτο είναι η εξαγωγή των χαρακτηριστικών και η δημιουργία διανυσμάτων, το δεύτερο η συσταδοποίηση των διανυσμάτων, και το τρίτο η παρουσίαση των αποτελεσμάτων.

3.1 Δημιουργία διανυσμάτων

Για να ομαδοποιήσουμε τα πλάνα με κάποιον αλγόριθμο clustering θα πρέπει πρώτα να τα μετατρέψουμε σε κάποια πιο κατάλληλη μορφή. Αυτό θα γίνει δημιουργώντας ένα διάνυσμα (vector) για κάθε πλάνο. Αυτό το διάνυσμα θα περιέχει όλες τις πληροφορίες του πλάνου, όπως αυτές δίνονται από τα δεδομένα μας. Οι πληροφορίες θα προέρχονται από δυο πηγές. Αφενός από την οντολογία, αφετέρου από τις κειμενικά τμήματα. Ο αριθμός των διαστάσεων των διανυσμάτων θα είναι προφανώς ίδιος για όλα τα πλάνα, και θα είναι ίσος με το πλήθος των χαρακτηριστικών που θα εξάγουμε (feature extraction) από τις πληροφορίες του πλάνου.

Για την ανάγνωση και επεξεργασία των δεδομένων του ABox, χρησιμοποιήθηκε το Java framework Apache Jena, έκδοση 2.1.2.1. Το Jena είναι ένα δωρεάν και open source framework για Java για την ανάπτυξη εφαρμογών Σημασιολογικού Ιστού και Διασυνδεδεμένων Δεδομένων. Αποτελείται από διάφορα APIs που αλληλεπιδρούν για την επεξεργασία RDF δεδομένων.



Εικόνα 14. Η αλληλεπίδραση ανάμεσα στα διαφορετικά APIs του Jena

3.1.1 Εξαγωγή χαρακτηριστικών από οντολογία

Για τα δεδομένα του N-Triples RDF αρχείου χρησιμοποιήθηκε η δομή δεδομένων *Model*, η οποία περιέχεται στο Jena. Το μοντέλο υποδηλώνει έναν RDF γράφο, ο οποίος περιέχει ένα σύνολο RDF κόμβων που συνδέονται με επισημασμένες σχέσεις. Κάθε σχέση είναι μονόδρομη. Έτσι η τριάδα

Example:idj foaf:name “Ian”

Μπορεί να διαβαστεί ως «Ο πόρος Example:idj έχει ιδιότητα foaf:name με τιμή "Ian". Είναι προφανές ότι το αντίθετο δεν ισχύει. Μαθηματικά, αυτό κάνει το μοντέλο ένα στιγμιότυπο ενός κατευθυνόμενου γράφου.

Στη Java χρησιμοποιούμε την κλάση Model ως τον κύριο περιέκτη για πληροφορίες RDF σε μορφή γράφου. Η κλάση αυτή είναι σχεδιασμένη να έχει πλούσιο API, με πολλές μεθόδους ώστε να διευκολύνεται η ανάπτυξη RDF προγραμμάτων και εφαρμογών. Επίσης, το Model παρέχει μια αφαίρεση (abstraction) πάνω στους διάφορους τρόπους αποθήκευσης RDF κόμβων και σχέσεων.

Στην εφαρμογή λοιπόν, φορτώνουμε σε ένα μοντέλο τις τριάδες των RDF δεδομένων με την εντολή:

Model myModel = FileManager.get().loadModel(filename);

Στη συνέχεια, πρέπει να φορτωθεί η οντολογία. Για αυτό το σκοπό, θα δημιουργήσουμε ένα μοντέλο οντολογίας. Το μοντέλο οντολογίας είναι μια επέκταση του μοντέλου RDF της Jena που χρησιμοποιήθηκε από πάνω, και παρέχει κάποιες επιπλέον δυνατότητες για τον χειρισμό οντολογιών. Η εντολή που χρησιμοποιήθηκε είναι η:

OntModel myOntology = ModelFactory.createOntologyModel(<model spec>);

Το model spec που επιλέγουμε είναι το OWL_MEM_MINI_RULE_INF. Αυτό το specification χρησιμοποιεί την mini OWL inference engine, για να εντοπίσει επιπλέον συνεπαγωγές ανάμεσα στις κλάσεις τις οντολογίας, με χρήση reasoning. Ο OWLMini reasoned υποστηρίζει τις ίδιες λειτουργίες με τον default reasoner του Jena, με τη διαφορά ότι παραλείπει τις ευθείς συνεπαγωγές από περιορισμούς minCardinality και someValuesFrom, αποφεύγοντας κάποιες άπειρες επεκτάσεις.

Έχοντας την οντολογία πλέον περασμένη στο μοντέλο, πρέπει να καταγραφούν όλες οι υποκατηγορίες των τριών κύριων οντοτήτων, οι οποίες είναι οι Shot, Scene και Act. Προφανώς όλες οι υποκατηγορίες αυτές θα είναι παιδιά της κύριας κλάσης αν φανταστούμε την οντολογία ως ένα σύνολο γράφων. Χρησιμοποιώντας την εντολή της Jena, listSubClasses(), δημιουργούμε τις λίστες shotClassList και sceneClassList που περιέχουν όλα τα είδη Shot και Scene αντίστοιχα. Παράλληλα, κρατάμε σε ξεχωριστή λίστα, τις συσχετίσεις ανάμεσα στις υποκλάσεις του Shot, δηλαδή ποιος είναι παιδί ποιου. Κάτι τέτοιο δεν είναι αναγκαίο για τις υποκλάσεις του Scene, καθώς παρατηρούμε ότι όλες είναι απευθείας παιδιά, δηλαδή το δέντρο του έχει ύψος 2.

Με μια γρήγορη εξέταση, βλέπουμε ότι στα δεδομένα του RDF αρχείου η μόνη κατηγορία Act που συναντάται είναι η SpeakingAct. Ως εκ τούτου δεν θα μπορέσει το δεδομένο αυτό να βοηθήσει στην διαδικασία της ομαδοποίησης. Επομένως παραλείπεται η προαναφερθείσα διαδικασία για την οντότητα Act. Σε περίπτωση διαφορετικών δεδομένων, εάν χρειαστεί, είναι εύκολη η προσθήκη ενός παρόμοιου τμήματος κώδικα που θα λειτουργεί όπως τα αντίστοιχα για τις οντότητες Shot και Scene.

Αυτό που πρέπει να γίνει στη συνέχεια είναι να εντοπίσουμε όλα τα πλάνα της ταινίας που βρίσκονται στα δεδομένα με δυναμικό τρόπο. Για έναν άνθρωπο αυτό είναι απλή διαδικασία που μπορεί να γίνει με το μάτι, αλλά για μια μηχανή είναι πιο πολύπλοκο. Θα μπορούσε να γίνει με ρητή δήλωση από μέρους μας, των

προγραμματιστών, αλλά αυτό αφενός απαιτεί περισσότερο χειρόγραφο κώδικα, και αφετέρου δεν καλύπτει την περίπτωση που νέα πλάνα προστίθενται στα δεδομένα, ακόμα και αν αυτά ακολουθούν τους σωστούς κανόνες συντακτικού!

Αυτό που κάνουμε επομένως είναι ένα query στα δεδομένα, με το SPARQL API της Jena. Η γενική μορφή που ακολουθείται για να υποβληθεί ένα SPARQL query επί των RDF δεδομένων του N-Triples αρχείου είναι η εξής:

```
import com.hp.hpl.jena.query.* ;
Model model = ... ;
String queryString = " .... " ;
Query query = QueryFactory.create(queryString) ;
try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
  ResultSet results = qexec.execSelect() ;
  for ( ; results.hasNext() ; )
  {
    QuerySolution soln = results.nextSolution() ;
    RDFNode x = soln.get("varName") ;       // Get a result variable by name.
    Resource r = soln.getResource("VarR") ; // Get a result variable - must be a resource
    Literal l = soln.getLiteral("VarL") ;   // Get a result variable - must be a literal
  }
}
```

Εικόνα 6. Υπόδειγμα SPARQL query στην Jena

Πιο συγκεκριμένα, το query που χρησιμοποιούμε εδώ έχει την εξής σύνταξη:

```
String queryStringShot =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT *" +
        "WHERE {" +
        "          ?x rdf:type ?z ." +
        "}";
```

Το query επιστρέφει όλες τις τριάδες, για τις οποίες το κατηγορούμενο είναι το resource URI <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>.

Κρατάμε τα αντικείμενα αυτών των τριάδων, και τα διασταυρώνουμε με τη λίστα των υποκατηγοριών του Shot που δημιουργήσαμε παραπάνω. Κρατάμε μια τριάδα αν το αντικείμενο της βρίσκεται μέσα στη λίστα αυτή.

Τώρα που όλα τα πλάνα της ταινίας έχουν εντοπιστεί, μπορούμε να προχωρήσουμε στη διαδικασία της δημιουργίας των διανυσμάτων. Εξετάζουμε το κάθε πλάνο ξεχωριστά με μια επαναληπτική διαδικασία, και για καθένα από αυτά κάνουμε την παρακάτω διαδικασία.

Αρχικά δημιουργούμε τέσσερεις προσωρινές λίστες, τις temp1, temp2, temp3 και temp4.

Η λίστα temp1 έχει μέγεθος ίσο με την προαναφερθείσα shotClassList, και θα χρησιμεύσει για να κρατήσουμε την πληροφορία που θα δώσει η οντολογία για το είδος του πλάνου.

Η λίστα temp2 επίσης, έχει μέγεθος ίσο με την προαναφερθείσα sceneClassList, και θα χρησιμεύσει για να κρατήσουμε την πληροφορία που θα δώσει η οντολογία για το είδος της σκηνής. Όπως θα δούμε στη συνέχεια, η πληροφορία αυτή θα είναι διπλή, δηλαδή κάθε σκηνή θα ανήκει σε δυο είδη.

Η λίστα temp3 έχει τόσα στοιχεία όσα είναι και οι χαρακτήρες της ταινίας, και θα δείχνει ποιοι χαρακτήρες, και κατ' επέκταση ηθοποιοί, εμφανίζονται στο πλάνο που εξετάζεται.

Η λίστα temp4 τέλος, θα περιέχει την ανάλυση όλης της κειμενικής πληροφορίας που παρέχεται για το πλάνο. Για το κομμάτι αυτό θα μιλήσουμε στην επόμενη ενότητα.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι οι πληροφορίες για κάποιο πλάνο στα RDF δεδομένα, προέρχονται από τρεις πηγές:

- Τις πληροφορίες για το ίδιο το πλάνο.
- Τις πληροφορίες της σκηνής στην οποία ανήκει το πλάνο.
- Τα acts που συμβαίνουν στο πλάνο.

Επομένως θα πραγματοποιήσουμε τρία queries στο RDF μοντέλο, ένα για καθεμία από τις παραπάνω πηγές πληροφοριών. Καθώς το id του πλάνου αλλάζει σε κάθε επανάληψη (δηλαδή το URI που το περιγράφει), θα πρέπει να χρησιμοποιείται στα queries μια μεταβλητή η οποία θα περιέχει κάθε φορά το URI του υπό εξέταση πλάνου. Αυτό το επιτυγχάνουμε ορίζοντας ένα χάρτη από τον οποίο θα διαβάζονται τα URIs των πλάνων, και φορτώνοντας σε αυτόν τα URIs όλων των πλάνων με τις εντολές:

QuerySolutionMap initialBindings = new QuerySolutionMap();

initialBindings.add("shot", shot);

Πληροφορίες πλάνου

Το πρώτο query, αυτό για το ίδιο το πλάνο, έχει την εξής σύνταξη.

```
String queryStringA =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT * " +
        "WHERE {" +
        "        ?shot ?y ?z ." +
        " }";
```

Το query αυτό επιστρέφει όλες τις τριάδες που έχουν για υποκείμενο το πλάνο που εξετάζουμε. Κάθε πλάνο στα RDF δεδομένα, έχει τις πληροφορίες όπως φαίνονται στην επόμενη εικόνα.

```
<MOV_12794_SH0741>  < isNextOf>      < MOV_12794_SH0740> .
<MOV_12794_SC0063>  < hasPart>       < MOV_12794_SH0741> .
<MOV_12794_SH0741>  < startTime>     "00:30:32.76".
<MOV_12794_SH0741>  <type>           < LowAngleShot> .
<MOV_12794_SH0741>  <type>           < MediumShot>   .
<MOV_12794_SH0741>  < description>   "Suddenly Joe swings the blade"@en.
```

Εικόνα 16. Παράδειγμα πλάνου στα RDF δεδομένα

Άρα οι τέσσερις περιπτώσεις κατηγορούμενου που συναντάμε στην επεξεργασία των αποτελεσμάτων του query είναι τα resources isNextOf, startTime, type και description. Ανάλογα με το κατηγορούμενο της τριάδας κάνουμε μια συγκεκριμένη ενέργεια:

- isNextOf: Η συγκεκριμένη πληροφορία δεν μπορεί να αξιοποιηθεί με κάποιο τρόπο, μιας και είναι προφανές ότι συνεχόμενα πλάνα θα έχουν κάποια συνάφεια. Ωστόσο δεν είναι αυτή η συνάφεια που θέλουμε να εντοπίσουμε με την ομαδοποίηση που θα ακολουθήσει.
- startTime: Κρατάμε τη χρονική στιγμή στην οποία κάθε πλάνο ξεκινάει σε μια λίστα. Θα χρησιμοποιηθούν σε επόμενο σημείο της εργασίας, στο στάδιο της παρουσίασης των αποτελεσμάτων της συσταδοποίησης.
- Type: Η βασική πληροφορία που θέλουμε από αυτό το query. Είναι το είδος (ή τα είδη) του πλάνου. Την χρησιμοποιούμε θέτοντας την τιμή 1 στη λίστα temp1 που ορίσαμε παραπάνω, στο στοιχείο που αντιπροσωπεύει την συγκεκριμένη υποκατηγορία της κλάσης Shot, αλλά και σε όλες τις υπερκλάσεις αυτής. Έτσι στο τέλος η λίστα temp1 είναι κάθε στοιχείο ίσο με το μηδέν, εκτός από τα στοιχεία εκείνα που αντιπροσωπεύουν τα είδη πλάνου στα οποία ανήκει το υπό εξέταση πλάνο. Στο παράδειγμα που φαίνεται στην Εικόνα 6, τιμή 1 θα έχουν τα στοιχεία που αντιπροσωπεύουν τις κλάσεις «LowAngleShot» και «MediumShot», καθώς και όλες οι υπερκλάσεις αυτών.
  Να τονίσουμε τέλος, ότι μιας και όλα τα είδη πλάνων είναι υποκλάσεις του Shot, όλα τα πλάνα θα έχουν τιμή 1 στη διάσταση που αντιπροσωπεύει το χαρακτηριστικό «Shot».
- Description: Πρόκειται για την περιγραφή του πλάνου και πληροφορία κειμενικού τύπου, και ο τρόπος αξιοποίησης της θα αναλυθεί στην επόμενη ενότητα.

Πληροφορίες σκηνής

Σειρά έχει η σκηνή στην οποία ανήκει το πλάνο. Το query που θα βοηθήσει σε αυτό έχει την παρακάτω μορφή:

```
String queryStringB =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT ?x ?y ?z " +
        "WHERE {" +
        "{       ?x movie:hasPart ?shot ." +
        "        ?x rdf:type ?z ." +
        "}" +
        "UNION {" +
        "        ?x movie:hasPart ?shot ." +
        "        ?x movie:description ?z .}" +
        " }";
```

Κατ' αντιστοιχία με το πλάνο, ένα παράδειγμα των δεδομένων που αναφέρονται σε μια σκηνή παρουσιάζονται στην επόμενη εικόνα.

```
< MOV_12794>    < hasPart>  < MOV_12794_SC0144> .
< MOV_12794_SC0144> < isNextOf> < MOV_12794_SC0143> .
< MOV_12794_SC0144> < startTime>    "01:13:08.20".
< MOV_12794_SC0144> < type>     < ExternalScene>    .
< MOV_12794_SC0144> < description>  "BOULEVARD"@en  .
< MOV_12794_SC0144> < type>     < DayScene> .
```

Εικόνα 7. Παράδειγμα σκηνής στα RDF δεδομένα.

Με τον τρόπο που συντάχθηκε το συγκεκριμένο query, δίνεται η δυνατότητα να κρατηθούν μόνο κάποια από τα δεδομένα αυτά της σκηνής, αυτά που όντως χρειάζονται. Αυτά είναι τα type και description. Και πάλι, πράττουμε διαφορετικά ανάλογα με το ποιο resource είναι το κατηγορούμενο της τριάδας:

- Type: Σε αυτή την περίπτωση, γίνονται οι αντίστοιχες διαδικασίες με την περίπτωση που το κατηγορούμενο ήταν το resource «type» στο query του πλάνου. Η διαφορά είναι ότι οι αλλαγές τιμών που κάνουμε είναι στη λίστα temp2, μιας και το είδος αφορά την υποκατηγορία της σκηνής. Να τονίσουμε ότι όλες οι σκηνές έχουν πάντα δύο είδη στα RDF δεδομένα. Το πρώτο θα μπορούσαμε να πούμε ότι αναφέρεται στο αν η σκηνή τραβήχτηκε μέρα ή νύχτα, και το δεύτερο αναφέρεται στο αν είναι σε εσωτερικό ή εξωτερικό χώρο. Άρα στο συγκεκριμένο παράδειγμα, τα μόνα στοιχεία της λίστας temp2 θα είναι αυτά που αναφέρονται στα χαρακτηριστικά «ExternalScene» και «DayScene».

- Description: Όπως και στο query του πλάνου, το αντικείμενο της τριάδας με κατηγορούμενο «description» αποτελεί την περιγραφή της σκηνής, είναι πληροφορία κειμενικής μορφής και θα χρησιμοποιηθεί στην επόμενη ενότητα.

Είναι προφανές, ότι αφού κάθε σκηνή περιλαμβάνει παραπάνω από ένα πλάνα, οι πληροφορίες αυτές θα αξιοποιηθούν πολλαπλές φορές. Αυτό ήδη, διαισθητικά και μόνο, δίνει την εντύπωση ότι θα οδηγήσει σε πιο αξιόπιστη ομαδοποίηση των πλάνων στη συνέχεια.

Τα acts του πλάνου

Τέλος θα αξιοποιηθούν οι πληροφορίες των acts που συμβαίνουν στο υπό εξέταση πλάνο. Το query που τις επιστρέφει είναι το εξής:

```
String queryStringC =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT ?x ?y ?z " +
        "WHERE {" +
        "       ?x movie:happensIn ?shot ." +
        "       ?x ?y ?z ." +
        " }";
```

Οι τριάδες RDF που επιστρέφει το query αυτό έχουν την παρακάτω μορφή.

```
<MOV_12794_AC289>   <type>  <SpeakingAct>   .
<MOV_12794_AC289>   <happensIn>     <MOV_12794_SH0916>  .
<MOV_12794_AC289>   <performedBy>   <MOV_12794_RL0> .
<MOV_12794_AC289>   <addressedTo>   <MOV_12794_RL17>    .
<MOV_12794_AC289>   <text>  "Ha, the whole thing."@en   .
```

Εικόνα 18. Παράδειγμα act στα RDF δεδομένα.

Οι κατηγορίες των δεδομένων είναι :

- Type: Όπως αναφέραμε και κατά τη διαδικασία της αναζήτησης των υποκλάσεων των Shot και Scene, η συγκεκριμένη πληροφορία στο σενάριο που εξετάζουμε είναι περιττή καθώς όλα τα acts έχουν τον ίδιο τύπο, τον «SpeakingAct». Επομένως δεν κάνουμε κάποια ενέργεια εδώ. Ωστόσο αν σε κάποιο άλλο σενάριο υπάρχει διαφοροποίηση, μπορεί να γίνει εύκολα μια

αντίστοιχη υλοποίηση με αυτές που πραγματοποιήσαμε στα δυο προηγούμενα queries.

- happensIn: Πρόκειται για πληροφορία που δεν αξιοποιείται κάπως, απλά επιβεβαιώνει ότι το query επέστρεψε έγκυρα αποτελέσματα.
- performedBy / adressedTo: Η βασική οντολογική πληροφορία του query. Αναφέρεται στους χαρακτήρες της ταινίας που συμμετέχουν στο act και επομένως στο πλάνο. Όπως έχει αναφερθεί και προηγουμένως, κάθε χαρακτήρας της ταινίας στα δεδομένα περιγράφεται από δύο στοιχεία, έναν αριθμό ή id και το όνομα του χαρακτήρα. Για παράδειγμα, υπάρχει ο χαρακτήρας με στοιχεία «RL10» και «Kong».
  Έτσι, οι ενέργειες που κάνουμε είναι: Πρώτον, θέτουμε ίση με 1 την τιμή του κ-στου στοιχείου της λίστας temp3, όπου κ είναι το id του ρόλου. Δεύτερον, κρατάμε το όνομα του χαρακτήρα και θα το αξιοποιήσουμε σαν κειμενική πληροφορία στην επόμενη ενότητα.
- Text: Πρόκειται για το διάλογο που γίνεται στο act. Θα αξιοποιηθεί σαν κειμενική πληροφορία μαζί με τις υπόλοιπες.

Σε αυτό το σημείο λοιπόν, με τις ενέργειες που περιγράψαμε σε αυτή την ενότητα, έχει αξιοποιηθεί η οντολογία που δόθηκε στα δεδομένα, για την εξαγωγή ενός συνόλου χαρακτηριστικών που, αν όλα πάνε σύμφωνα με το σχέδιο, θα προσφέρουν μεγαλύτερη ακρίβεια στα αποτελέσματα της συσταδοποίησης.

3.1.2 Εξαγωγή χαρακτηριστικών από κειμενικές πληροφορίες

Από τα προηγούμενα queries πάνω στο μοντέλο των RDF δεδομένων, συγκεντρώσαμε ένα σύνολο διαφορετικών κειμενικών πληροφοριών για το κάθε πλάνο. Αυτές είναι οι περιγραφές του πλάνου και της σκηνής στην οποία αυτό ανήκει, οι διάλογοι που γίνονται στα acts που συμβαίνουν στο πλάνο, καθώς και τα ονόματα των χαρακτήρων που εμφανίζονται σε αυτό.

Θα μπορούσε κάποιος να πει ότι αυτό το τελευταίο κομμάτι είναι περιττό, και ότι έχει καλυφθεί πλήρως το θέμα των χαρακτήρων από την οντολογία και τη δημιουργία της λίστας temp3. Αυτό δεν ισχύει, και θα φανεί από το εξής παράδειγμα. Έστω δυο διαφορετικά πλάνα. Το πρώτο, έστω πλάνο Α, έχει κάποιο act στο οποίο συμμετέχει ο j-στός χαρακτήρας της ταινίας, και αυτό αντικατοπτρίζεται στις πληροφορίες του. Στο δεύτερο, έστω πλάνο Β, δεν υπάρχει κάποια ρητή αναφορά ότι ο ίδιος χαρακτήρας συμμετέχει, αλλά το όνομα του εμφανίζεται είτε στην περιγραφή του πλάνου ή της σκηνής, είτε στο διάλογο ενός act του πλάνου. Εάν δεν κρατούσαμε και το όνομα του χαρακτήρα σε κειμενική μορφή, πιθανότατα δεν θα εντοπιζόταν συνάφεια ανάμεσα στα δυο πλάνα αυτά, που θα ήταν λάθος. Για αυτό το λόγο λοιπόν κρατάμε και αυτή την επιπλέον πληροφορία.

Τα ζεύγη «id χαρακτήρα»-«όνομα χαρακτήρα» τα βρίσκουμε με το ακόλουθο query.

```
String queryStringRole =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT ?y ?z " +
        "WHERE {" +
        "       ?x movie:hasRole ?y ." +
        "       ?y movie:name ?z ." +
        "}";
```

Επιστρέφοντας στην διαδικασία εξαγωγής χαρακτηριστικών, συγκεντρώνουμε όλες αυτές τις κειμενικές πληροφορίες σε ένα ενιαίο Java String. Για την περαιτέρω επεξεργασία τους χρειάζεται να τεθεί ένα μέτρο σύγκρισης, δηλαδή πως θα μπορέσει να γίνει σύγκριση ανάμεσα στα κείμενα δυο πλάνων, μιας και ένα κείμενο δεν είναι quantifiable μέγεθος.

Η διαδικασία που επιλέχθηκε σε αυτή την εφαρμογή είναι η λημματοποίηση. Το θεωρητικό της υπόβαθρο έχει ήδη αναλυθεί στο πρώτο κομμάτι της εργασίας. Για την υλοποίηση της χρησιμοποιήθηκε το Stanford CoreNLP API, που έχει αναπτυχθεί από το Stanford Natural Language Processing Group. Πρόκειται για ένα εργαλείο που προσφέρει πολλές από τις βασικές διαδικασίες της επεξεργασίας φυσικού λόγου, μεταξύ αυτών και η λημματοποίηση. Είναι ένα integrated framework, απλό στη χρήση του, αλλά και πολύ ευέλικτο και επεκτάσιμο.

Για τη διαδικασία της λημματοποίησης υλοποιήθηκε η κλάση StanfordLemmatizer, που περιέχει τη συνάρτηση lemmatize η οποία είναι υπεύθυνη για τη λημματοποίηση μιας συμβολοσειράς. Αυτή λειτουργεί ως εξής.

Αρχικά δημιουργείται ένα αντικείμενο ιδιοτήτων Properties, το οποίο περιέχει τα ονόματα των NLP διαδικασιών στις οποίες θα υποβληθεί η συμβολοσειρά. Αυτές είναι οι:

- Tokenize: διαχωρίζει το input κείμενο σε μια αλληλουχία από λεκτικά (tokens). Το αγγλικό μέρος παρέχει έναν PTB style tokenizer (Penn Treebank), ικανό να διαχειριστεί και «θορυβώδες» κείμενο.
- Ssplit: Χωρίζει μια αλληλουχία από λεκτικά σε προτάσεις.
- Pos: Επισημαίνει λεκτικά με μια μέρος-του-λόγου (part-of-speech, POS) ετικέτα, χρησιμοποιώντας tagger μέγιστης εντροπίας.
- Lemma: Παράγει τα λήμματα (βασικές μορφές) για όλα τα λεκτικά στο annotation.

Επίσης δημιουργείται ένα αντικείμενο Annotation με όρισμα το input String. Αυτό ουσιαστικά είναι μια εφαρμογή του CoreMap που «ξέρει» από κείμενα.

Στην συνέχεια δημιουργείται ένα StanfordCoreNLP pipeline που παίρνει τις προηγούμενες ιδιότητες ως annotations.

Τέλος, εφαρμόζονται όλες οι NLP διαδικασίες που ορίζουν οι ιδιότητες στο κείμενο, και στο τέλος επιστρέφονται τα λήμματα σε μια λίστα.

Η ιδέα πίσω από τη διαδικασία που γίνεται εδώ, είναι ότι για κάθε πλάνο θα δημιουργούμε μια λίστα που θα περιέχει όλες τις διαφορετικές λέξεις που υπάρχουν στα κειμενικά τμήματα του. Στη συνέχεια, αυτή η λίστα θα διασταυρώνεται με ένα λεξικό, μια μεγάλη λίστα δηλαδή που περιέχει όλες τις διαφορετικές λέξεις όλων των πλάνων μαζί, δηλαδή ολόκληρης ταινίας ουσιαστικά. Και σε ένα υπό-διάνυσμα, με διαστάσεις όσες και το μέγεθος του λεξικού, θα σημειώνεται ποιες λέξεις του λεξικού συναντώνται στο συγκεκριμένο πλάνο.

Για τη δημιουργία του προαναφερθέντος λεξικού, χρησιμοποιείται το παρακάτω query επί του μοντέλου των RDF δεδομένων.

```
String queryStringDictionary =
        "PREFIX movie: <http://image.ntua.gr/scriptontology/> " +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT ?z " +
        "WHERE {" +
        "{      ?x movie:description ?z .}" +
        "UNION" +
        "{      ?x movie:text ?z .}" +
        "UNION" +
        "{      ?x movie:name ?z .}" +
        "}";
```

Αυτό το query επιστρέφει όλες τις κειμενικές πληροφορίες που αναφέραμε στην αρχή της ενότητας, δηλαδή τις περιγραφές πλάνων και σκηνών, τους διαλόγους των acts, και τα ονόματα των χαρακτήρων, αλλά για ολόκληρη την ταινία.

Στη συνέχεια συνενώνουμε όλες τις πληροφορίες αυτές σε ένα ενιαίο κείμενο και προχωράμε στην επεξεργασία του. Αρχικά κάνουμε μια διαδικασία που ονομάζεται αφαίρεση αδιάφορων λέξεων (stop words removal). Σε αυτή αφαιρούμε από ένα κείμενο όλες τις λέξεις που δεν θα παίξουν κάποιο ρόλο στην επεξεργασία φυσικού λόγου. Αν και ως αδιάφορες λέξεις θεωρούνται οι πιο συνηθισμένες λέξεις μιας γλώσσας, δεν υπάρχει κάποια καθολικά αποδεκτή λίστα που να ορίζει ποιες ακριβώς είναι αυτές. Επομένως αφήνεται στον καθένα που κάνει μια τέτοια διαδικασία, να ορίσει ποιες λέξεις θα θεωρήσει ως αδιάφορες. Η πιο διαδεδομένη πρακτική είναι να θεωρούνται αδιάφορες λέξεις οι μικρές λέξεις όπως άρθρα, επιρρήματα και προθέσεις, ίσως και τα πολύ συνηθισμένα ρήματα όπως τα «be», «do» και «have». Αυτή είναι και η λογική που ακολουθήθηκε σε αυτή την εφαρμογή.

Έχοντας αφαιρέσει τις αδιάφορες λέξεις λοιπόν από το συγκεντρωτικό κείμενο, και φυσικά και τη στίξη, εφαρμόζουμε λημματοποίηση σε αυτό. Αυτό που μας επιστρέφεται είναι μια λίστα που έχει όλες τις λέξεις του κειμένου, αλλά στη βασική ή «λεξικολογική» τους μορφή. Για παράδειγμα, η λέξη «cars» του κειμένου θα μετατραπεί σε «car» στη λίστα. Ο κώδικας για αυτή την ενέργεια είναι :

```
StanfordLemmatizer slemm = new StanfordLemmatizer();
List<String> allLemmas = slemm.lemmatize(allText.toString());
```

Τέλος, κρατάμε μόνο τις διαφορετικές λέξεις της παραπάνω λίστας, δηλαδή διαγράφουμε όλα τα duplicate entries. Αν μια λέξη υπάρχει εμφανίζεται στη λίστα πάνω από μια φορές, κρατάμε την πρώτη της εμφάνιση και διαγράφουμε όλες τις υπόλοιπες.

Έτσι έχουμε φτιάξει το λεξικό που θα αποτελέσει τη βάση αναφοράς για όλα τα πλάνα.

Ακολούθως, σε ένα επαναληπτικό βρόγχο όμοιο με αυτόν που έτρεχε κατά την εξαγωγή χαρακτηριστικών από την οντολογία, εξετάζουμε τα πλάνα ένα προς ένα. Για κάθε πλάνο, εφαρμόζουμε τη συνάρτηση λημματοποίησης στο Java String που περιέχει τις κειμενικές πληροφορίες του, με τον ίδιο κώδικα που χρησιμοποιήσαμε για το λεξικό.

Για καθεμία από τις λέξεις του πλάνου, την αναζητούμε στο λεξικό – με δυαδική αναζήτηση για αποδοτικότητα. Αφού την εντοπίσουμε, κρατάμε τον δείκτη της, τη θέση δηλαδή που βρίσκεται στο λεξικό. Στη συνέχεια, θέτουμε την τιμή της αντίστοιχης θέσης της λίστας temp4 (που είχαμε ορίσει στην αρχή) ίση με 1. Είναι αυτονόητο ότι για να λειτουργήσει αυτό σωστά, η λίστα temp4 θα πρέπει να έχει μέγεθος όσο και το λεξικό.

Στο τέλος της διαδικασίας, για το υπό εξέταση πλάνο η λίστα temp4 θα έχει όλα τα στοιχεία της ίσα με το μηδέν, εκτός από αυτά που αντιπροσωπεύουν λέξεις που εμφανίστηκαν στα κειμενικά κομμάτια του πλάνου. Έτσι ολοκληρώνεται και η εξαγωγή χαρακτηριστικών από τις κειμενικές πληροφορίες των δεδομένων.

3.1.3 Μετάβαση στην συσταδοποίηση

Έχοντας συγκεντρώσει όλες τις πληροφορίες, όλα τα χαρακτηριστικά που θα μπορούσαν να εξαχθούν από τα δεδομένα πρέπει να προωθηθούν για συσταδοποίηση. Αυτό σημαίνει ότι η συνάρτηση της κλάσης που υλοποίησε όλη αυτή τη διαδικασία της δημιουργίας διανυσμάτων, η vectorize της κλάσης CreateVectors, πρέπει να επιστρέψει τα τελικά διανύσματα των πλάνων στην main συνάρτηση της βασικής κλάσης της εφαρμογής, την Thesis.java.

Ολόκληρος ο κώδικας όλων των κλάσεων της εφαρμογής μπορεί να βρεθεί στο παράρτημα Α.

Πριν επιστρέψει η συνάρτηση εξαγωγής χαρακτηριστικών, συνενώνουμε τις προσωρινές λίστες temp1 – temp4 στο τελικό διάνυσμα του πλάνου, με όνομα shotVector.

Η συνάρτηση δημιουργίας των διανυσμάτων επιστρέφει στη βασική κλάση ένα σύνολο δεδομένων. Μιας και στη Java αυτό δεν μπορεί να γίνει από μόνο του, γιατί οποιαδήποτε συνάρτηση μπορεί να επιστρέφει μία μόνο δομή δεδομένων, όποιου τύπου κι αν είναι αυτή, ορίζουμε μια νέα δομή δεδομένων, μια κλάση με όνομα myResult. Τα πεδία αυτής της κλάσης είναι:

- Μία λίστα διανυσμάτων, δηλαδή ουσιαστικά μια λίστα που περιέχει λίστες που περιέχουν ακεραίους.
- Μία δεύτερη λίστα, που είναι λίστα ακεραίων.
- Μια τρίτη λίστα, που περιέχει Strings.

Έτσι, η συνάρτηση δημιουργίας διανυσμάτων επιστρέφει ένα αντικείμενο τύπου myResult, τα πεδία του οποίου περιέχουν κατ' αντιστοιχία με τα αποπάνω:

- Τη λίστα με τα διανύσματα όλων των πλάνων της ταινίας, με όνομα finalVectors.
- Μια λίστα με δείκτες με όνομα indexes. Πρόκειται για τη λίστα που δείχνει με ποια σειρά βρίσκονται τα διανύσματα των πλάνων στην προηγούμενη λίστα. Αυτό είναι αναγκαίο γιατί τα πλάνα δεν διαβάζονται με τη σειρά. Το Apache Jena φορτώνει στο μοντέλο τις τριάδες των RDF δεδομένων με τυχαία σειρά, ή αν όχι τυχαία, σίγουρα μη σειριακή. Έτσι, αν για παράδειγμα της λίστας είναι ο ακέραιος αριθμός «1256», αυτό σημαίνει ότι το πρώτο πλάνο που εξετάστηκε ήταν το Πλάνο_1256, και επομένως το πρώτο διάνυσμα στη λίστα finalVectors, είναι το διάνυσμα που περιγράφει το Πλάνο_1256.
- Μια λίστα με ετικέτες, με όνομα labels. Αυτή η λίστα έχει ίδιες διαστάσεις με το διάνυσμα ενός πλάνου, και περιέχει τα ονόματα των χαρακτηριστικών που αποτελούν το διάνυσμα του πλάνου. Κοινώς, περιέχει όλα τα ονόματα των υποκατηγοριών πλάνου και σκηνής, τα ονόματα όλων των χαρακτήρων της ταινίας, και όλες τις λέξεις του λεξικού.

Πλέον είμαστε έτοιμοι να προχωρήσουμε στη συσταδοποίηση.

3.2 Συσταδοποίηση

Για τη συσταδοποίηση χρησιμοποιήθηκε το MATLAB, καθώς είναι ένα εργαλείο πολύ αποδοτικό και σχετικά απλό στη χρήση του. Στο MATLAB υπάρχουν ενσωματωμένες διάφορες μέθοδοι ομαδοποίησης δεδομένων. Από αυτές, επιλέξαμε να δοκιμάσουμε τρεις, τον Χάρτη Αυτό-Οργάνωσης (SOM), τον αλγόριθμο K-Means, και την ιεραρχική συσταδοποίηση.

Ως είσοδο, καθεμία από αυτές τις μεθόδους, πήρε τη λίστα των διανυσμάτων που κατασκευάστηκε στο προηγούμενο τμήμα της εφαρμογής. Αυτή η λίστα αποθηκεύτηκε σε ένα αρχείο κειμένου, από το οποίο την κάνει import το MATLAB.

3.2.1 Χάρτης Αυτό-Οργάνωσης

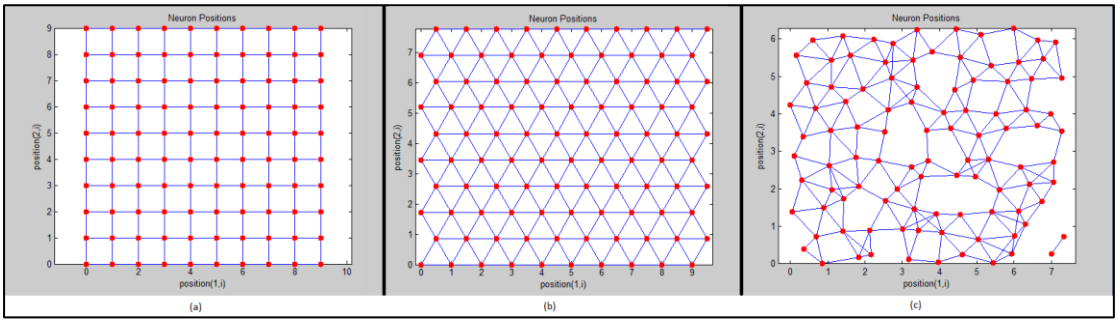
Ο Χάρτης Αυτό-Οργάνωσης μαθαίνει να ομαδοποιεί τα διανύσματα εισόδου βάσει του πως κατανέμονται στον χώρο εισόδου. Διαφέρει από τα ανταγωνιστικά στρώματα γιατί οι γειτονικοί νευρώνες μαθαίνουν να αναγνωρίζουν κομμάτια του χώρου εισόδου που γειτονεύουν. Επομένως ο SOM μαθαίνει τόσο από την κατανομή όσο και από τη τοπολογία των δεδομένων με τα οποία εκπαιδεύεται.

Ο κώδικας που χρησιμοποιήσαμε είναι ο εξής.

```
close all ; clear all;

filename = 'vectors.txt';
A = dlmread(filename);
inputs = transpose(A);

% Create a Self-Organizing Map
dimension1 = 6;
dimension2 = 6;
coverSteps = 100; %default 100
initNeighbor = 3 ; %default 3
topologyFcn = 'gridtop' ; %default 'hextop'
distanceFcn = 'linkdist'; %default 'linkdist'
net = selforgmap([dimension1 dimension2],
            coverSteps,initNeighbor,topologyFcn,distanceFcn);

% Train the Network
%%net.trainParam.epochs = 500;
[net,tr] = train(net,inputs);

% Test the Network
outputs = net(inputs);

% View the Network
view(net)
```

Υπάρχουν κάποιες μεταβλητές στη δημιουργία του SOM για τις οποίες πρέπει να γίνει μια επιλογή:

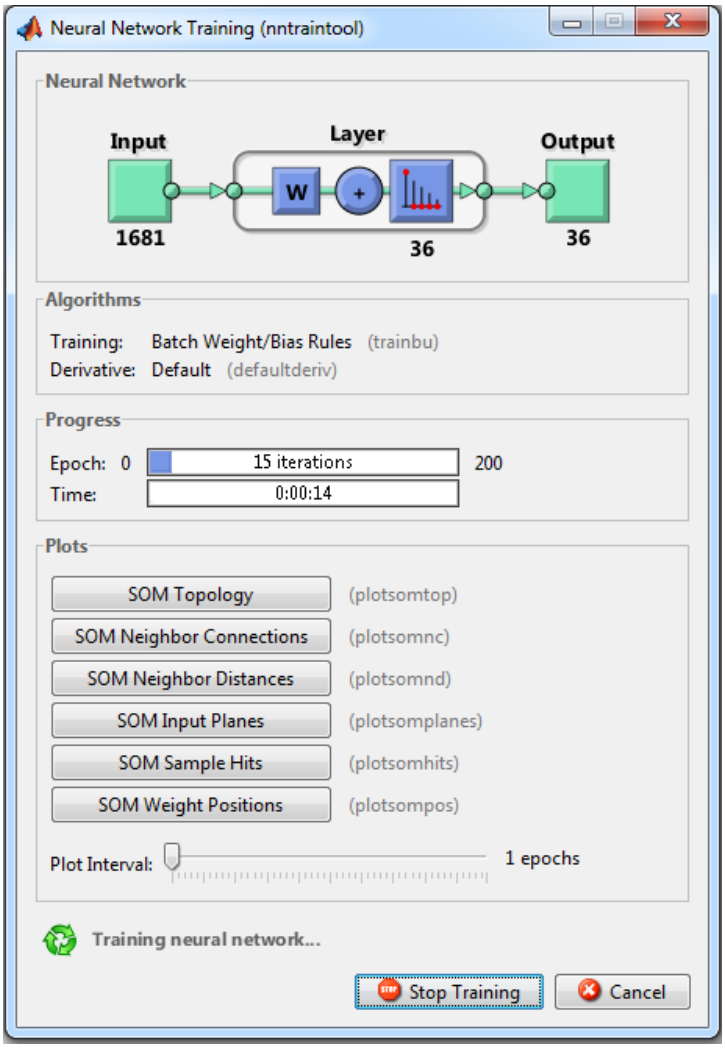
- Τοπολογίες: Οι επιλογές είναι οι “gridtop”, “hextop”, “randtop”, οι οποίες αντιπροσωπεύουν ορθογώνιο, εξαγωνικό, και τυχαίο N-διάστατο πλέγμα. Επίσης επιλέγονται ο αριθμός των διαστάσεων του πλέγματος και το μέγεθος τους. Εδώ επιλέχθηκε ένα ορθογώνιο πλέγμα 6x6 διαστάσεων.



Εικόνα 19. Παραδείγματα πλεγμάτων. (a) Ορθογώνιο (b) Εξαγωνικό (c) Τυχαίο

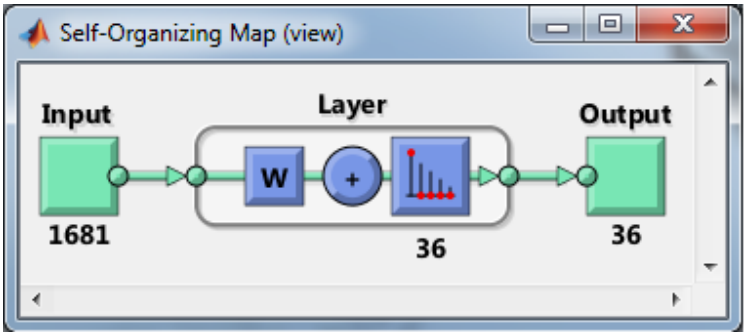
- Η συνάρτηση απόστασης. Προσφέρονται 4 επιλογές, οι συναρτήσεις “dist”, “linkdist”, “mandist” και “boxdist”. Επιλέξαμε την δεύτερη.
- Τα βήματα της φάσης οργάνωσης και το αρχικό μέγεθος γειτονιάς. Και για τα δυο χρησιμοποιήθηκε η default τιμή τους.
- Ο αριθμός των εποχών. Επιλέχθηκε πλήθος 200 εποχών.

Αφού δημιουργηθεί το δίκτυο, εμφανίζεται το παράθυρο εκπαίδευσης του νευρωνικού δικτύου, που δείχνει την πρόοδο της διαδικασίας.



Εικόνα 20. Εκπαίδευση νευρωνικού δικτύου

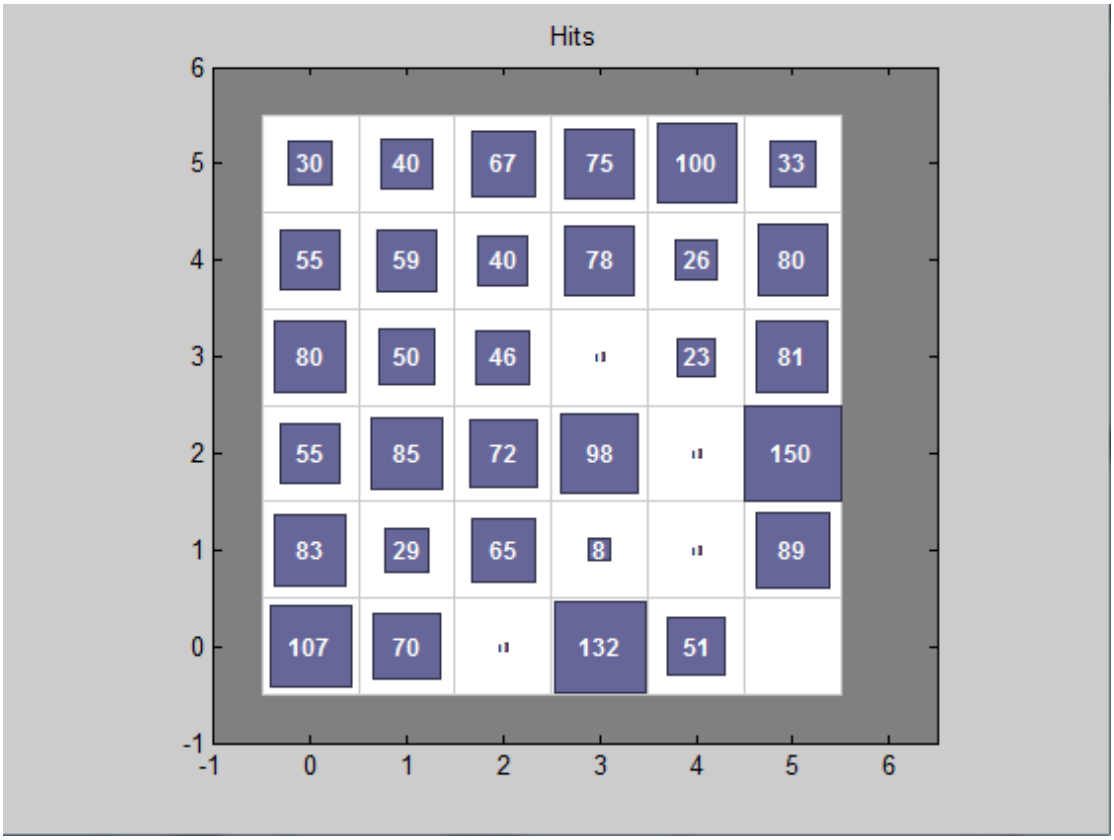
Αφού η διαδικασία ολοκληρωθεί, εμφανίζεται η αρχιτεκτονική του δικτύου που χρησιμοποιήθηκε.



Εικόνα 21. Αρχιτεκτονική SOM

Επίσης δίνεται η δυνατότητα σχεδιασμού κάποιων χαρακτηριστικών γραφικών παραστάσεων. Οι δύο βασικότερες είναι οι "SOM Sample Hits" και "SOM Neighbor Distance".

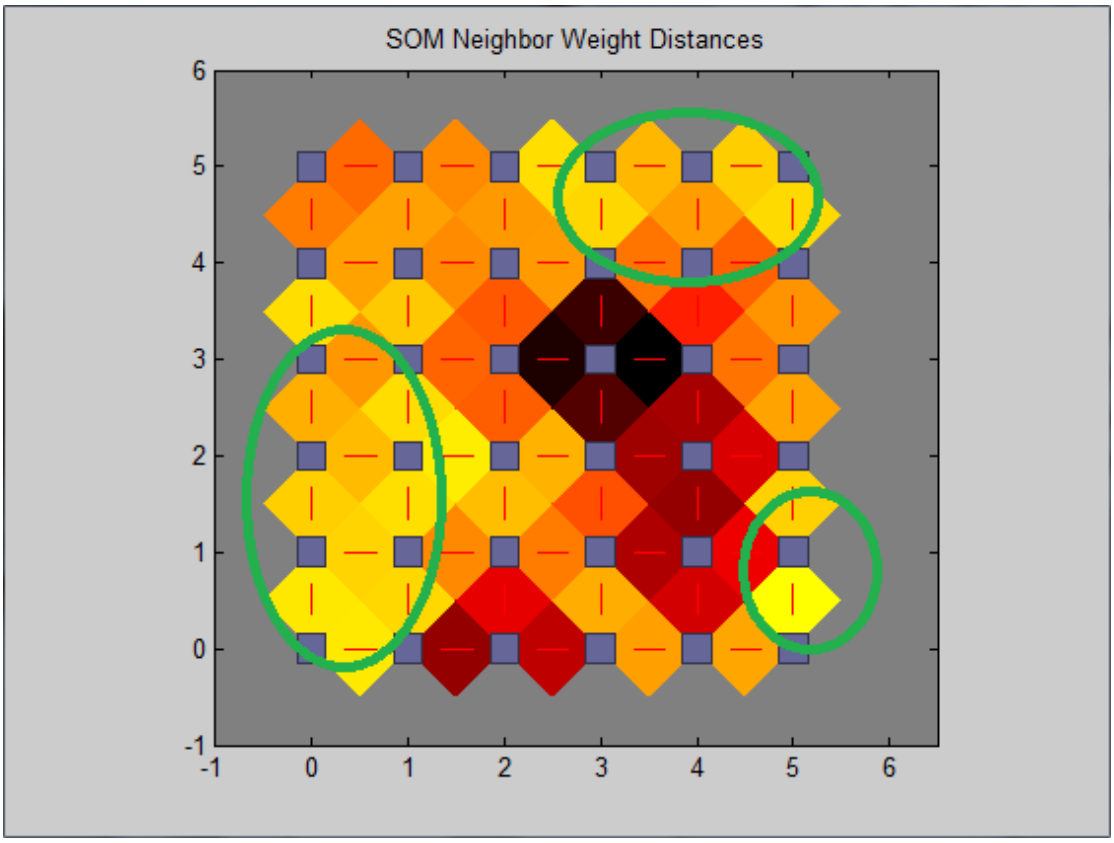
Η πρώτη δείχνει την τελική κατανομή των διανυσμάτων στο πλέγμα, δηλαδή πόσα διανύσματα ανατέθηκαν στον κάθε νευρώνα. Γενικά, επιζητείται μια ομοιόμορφη κατανομή στους νευρώνες. Αυτή η γραφική παράσταση για τα δεδομένα μας φαίνεται στην επόμενη εικόνα.



Εικόνα 22. SOM Sample Hits

Η δεύτερη παρουσιάζει την τελική απόσταση ανάμεσα στους νευρώνες. Οι νευρώνες αναπαρίστανται από τα μπλε εξάγωνα, και οι κόκκινες γραμμές συνδέουν τους γειτονικούς νευρώνες. Τα χρώματα ενδιάμεσα δείχνουν τις αποστάσεις, με τα σκούρα

χρώματα να συμβολίζουν τις μεγάλες αποστάσεις, και τα ανοιχτά τις μικρές. Το γράφημα αυτό για τα δεδομένα μας είναι το εξής.



Εικόνα 23. SOM Neighbor Distances

Παρατηρούμε ότι με βάση τις σκοτεινές περιοχές που εμφανίζονται, το δίκτυο έχει ταξινομήσει τα δεδομένα σε 3 κύριες περιοχές, τις οποίες δείχνουμε με τους πράσινους κύκλους.

3.2.2 K-Means Clustering

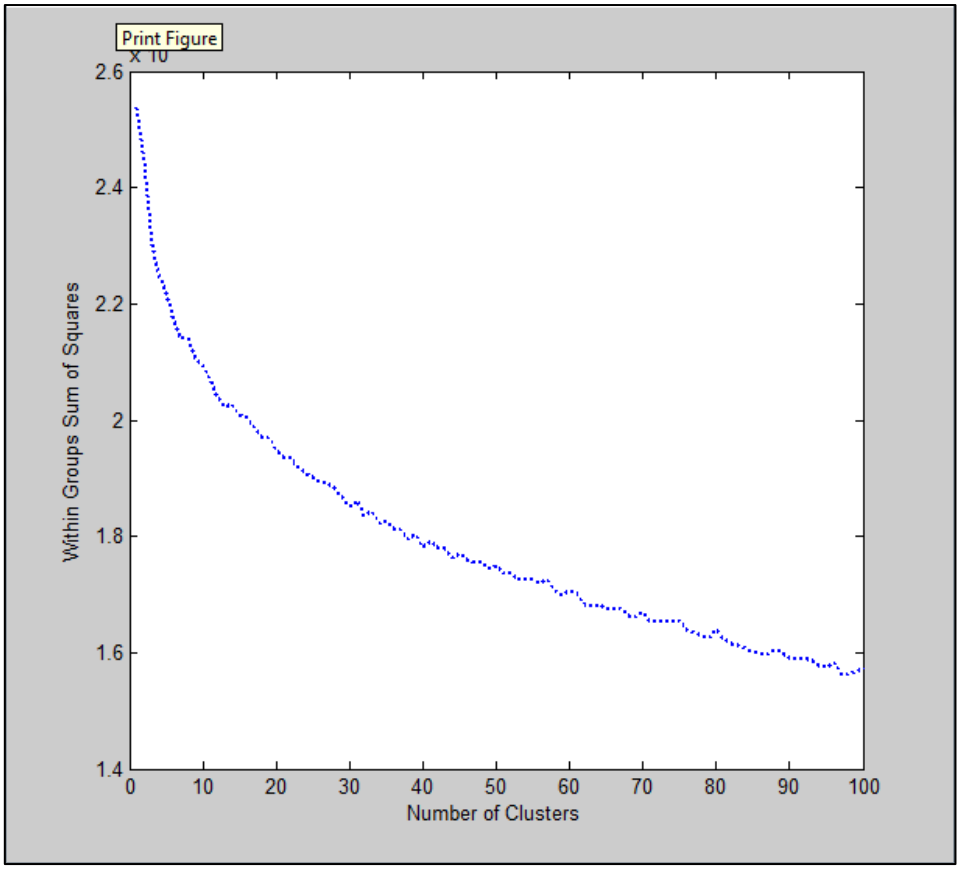
Ο αλγόριθμος K-Means είναι μια διαμεριστική μέθοδος, καθώς χωρίζει τα δεδομένα σε k ξεχωριστούς clusters. Δημιουργεί ένα επίπεδο από clusters, και λειτουργεί πάνω στις ίδιες τις παρατηρήσεις, και όχι πάνω στο ευρύτερο σύνολο των μέτρων ανομοιότητας, όπως πράττει η ιεραρχική συσταδοποίηση. Αυτό κάνει τον K-Means συχνά πιο κατάλληλο για μεγάλο όγκο δεδομένων, όπως στην περίπτωση μας.

Ο K-Means θεωρεί ότι κάθε παρατήρηση (δηλαδή κάθε διάνυσμα εισόδου) είναι ένα αντικείμενο που βρίσκεται κάπου στο χώρο. Βρίσκει μια διαμέριση στην οποία τα αντικείμενα που βρίσκονται στο ίδιο cluster είναι όσο το δυνατόν πιο κοντά μεταξύ τους, και ταυτόχρονα όσο πιο μακριά από τα αντικείμενα των υπόλοιπων clusters γίνεται. Κάθε cluster περιγράφεται από τα μέλη του, και από το κέντρο του, τον κεντροειδή. Ο κεντροειδής κάθε cluster είναι το σημείο για το οποίο το άθροισμα των αποστάσεων από όλα τα μέλη του ελαχιστοποιείται. Ο κεντροειδής δεν είναι απαραίτητο να συμπίπτει στο χώρο με κάποια από τις παρατηρήσεις.

Έχουμε ήδη αναλύσει πως δουλεύει ο αλγόριθμος K-Means, στο προηγούμενο κεφάλαιο, αυτό του θεωρητικού υπόβαθρου. Ο κώδικας σε MATLAB που χρησιμοποιήθηκε είναι ο παρακάτω.

```
close all ; clear all;

filename = 'vectors.txt';
A = dlmread(filename);
inputs = A;

k = 25;
[idx,C,sumd] = kmeans(inputs,k,'Display','final','Replicates',5);
```

Η βασική παράμετρος που έπρεπε να επιλεγεί είναι ο αριθμός των clusters k στους οποίους θα μοιραστούν τα δεδομένα. Ο αριθμός αυτός επιλέχθηκε χρησιμοποιώντας την «Elbow Method», η οποία αναλύθηκε στο προηγούμενο κεφάλαιο. Η γραφική παράσταση που προέκυψε από την μελέτη φαίνεται στην επόμενη εικόνα.



Εικόνα 24. Γραφική παράσταση για υπολογισμό κατάλληλου αριθμού clusters

Παρατηρούμε ότι το ζητούμενο σημείο εμφανίζεται για αριθμό clusters $k \approx 25$. Ως εκ τούτου, ο αλγόριθμος K-Means υλοποιήθηκε για αυτή ακριβώς την τιμή, όως φαίνεται και στον κώδικα.

Όσον αφορά τις υπόλοιπες παραμέτρους, επιλέχθηκε η default συνάρτηση απόστασης (distance), η Ευκλείδεια απόσταση (sqEuclidean), και η επίσης default ενέργεια σε

περίπτωση που κάποιο cluster χάσει όλες τις παρατηρήσεις του (emptyaction), που είναι να θεωρηθεί το συγκεκριμένο cluster ως σφάλμα (error). Επίσης επιλέχθηκε το ζεύγος "Display" – "final", ώστε να εμφανίζονται μόνο τα τελικά αποτελέσματα, και όχι τα αποτελέσματα κάθε βήματος.

Τέλος, ορίσαμε την παράμετρο "Replicates" στην τιμή 5. Αυτό σημαίνει ότι η διαδικασία του clustering έγινε 5 φορές. Αυτό έγινε διότι κάποιες φορές, ακόμα και για σχετικά απλά προβλήματα, ο αλγόριθμος K-Means υποπίπτει σε τοπικά ελάχιστα, στα οποία οποιαδήποτε ανατοποθέτηση μιας παρατήρησης σε διαφορετικό cluster αυξάνει το συνολικό άθροισμα της απόστασης σημείων – κεντροειδών. Ο αλγόριθμος επομένως τερματίζει σε εκείνο το βήμα, αν και μπορεί να βρεθεί καλύτερη τελική λύση. Με τη χρήση πολλαπλών replications, αυτό αποφεύγεται. Κάθε επανάληψη της διαδικασίας ξεκινάει από ένα διαφορετικό, τυχαία επιλεγμένο σύνολο αρχικών κεντροειδών, και η τελική λύση που επιστρέφεται είναι πάντα αυτή με το μικρότερο άθροισμα αποστάσεων από όλες τις επαναλήψεις.

Το αποτέλεσμα που εμφανίζεται είναι το παρακάτω.

```
41 iterations, total sum of distances = 15433.4

20 iterations, total sum of distances = 15570.6

33 iterations, total sum of distances = 15420.9

39 iterations, total sum of distances = 15652.7

28 iterations, total sum of distances = 15302.4

Best Total sum of distances = 15302.4
```

Τα αποτελέσματα του αλγορίθμου K-Means θα φανούν και στο Κεφάλαιο 5.

3.2.3 Ιεραρχική Συσταδοποίηση

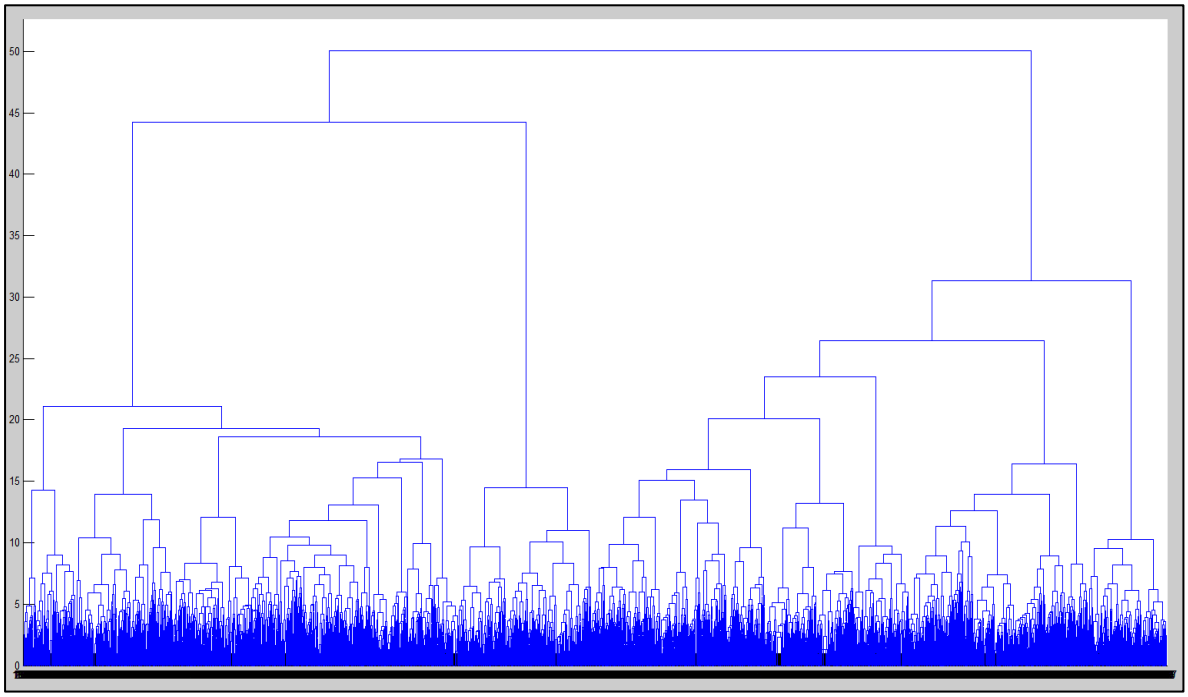
Ο κώδικας που χρησιμοποιήθηκε για την ιεραρχική συσταδοποίηση είναι ο παρακάτω.

```
close all ; clear all;

filename = 'vectors.txt';
A = dlmread(filename);
inputs = A;

T = clusterdata(inputs,'linkage','ward','maxclust',25);
```

Σε αυτή τη μέθοδο υπήρχαν δυο επιλογές. Η πρώτη ήταν να ορίσουμε ένα κατώφλι αποκοπής του δενδρογράμματος. Ωστόσο με δοκιμές παρατηρήθηκε ότι δεν μπορούσε να βρεθεί κατάλληλη τιμή για ικανοποιητική ομαδοποίηση των δεδομένων μας. Επομένως χρησιμοποιήθηκε η δεύτερη επιλογή, η οποία είναι να μην οριστεί κάποιο κατώφλι, αλλά ο μέγιστος αριθμός clusters που θέλουμε να προκύψουν, κατά παρόμοιο τρόπο με τον αλγόριθμο K-Means.

Έτσι, θέτοντας και τις παραμέτρους της απόστασης (distance) και απόστασης ανάμεσα σε clusters (linkage) στις συναρτήσεις ‘euclidean’ και ‘ward’ αντίστοιχα, προχωρήσαμε σε δοκιμές για την επιλογή του κατάλληλου αριθμού clusters.

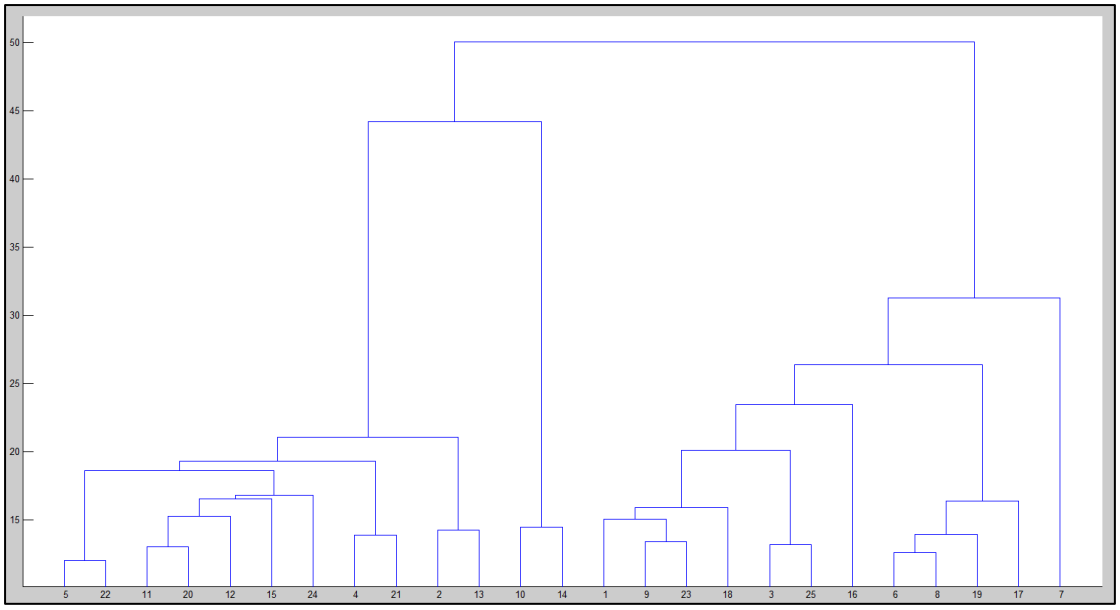
Στην ιεραρχική συσταδοποίηση, τα αποτελέσματα μπορούν να φανούν μέσω ενός δενδρογράμματος, που δείχνει το πώς τα δεδομένα ομαδοποιούνται. Το δενδρόγραμμα που προκύπτει στη δική μας περίπτωση είναι το παρακάτω.



Εικόνα 25. Το δενδρόγραμμα της ιεραρχικής συσταδοποίησης.

Τα πιο κάτω επίπεδα του δενδρογράμματος δεν φαίνονται καθαρά στην παραπάνω εικόνα, καθώς τα δεδομένα είναι πολλά, και ως εκ τούτου γίνονται πολλές συνδέσεις.

Στην επόμενη εικόνα φαίνεται το ίδιο δενδρόγραμμα, με τα χαμηλά επίπεδα αποκομμένα. Συγκεκριμένα εμφανίζονται οι ομαδοποιήσεις των τελευταίων 25 συστάδων που έχουν δημιουργηθεί από τα χαμηλότερα επίπεδα, όσα δηλαδή ζητήσαμε και στον κώδικα.



Εικόνα 26. Ανώτερα επίπεδα του δενδρογράμματος.

3.2.4 Αξιολόγηση Μεθόδων

Σε αυτό το σημείο θα πρέπει να γίνει μια αξιολόγηση των αποτελεσμάτων που παρήγαγαν οι τρεις διαφορετικές μέθοδοι συσταδοποίησης. Ωστόσο κάτι τέτοιο δεν είναι εύκολο, και αυτό οφείλεται στη μορφή των δεδομένων. Τα δεδομένα δεν παρουσιάζουν κάποια ιδιαιτερότητα και θα μπορούσαμε να πούμε ότι είναι σχετικά ομοιόμορφα κατανεμημένα στο διανυσματικό χώρο εισόδου. Ως εκ τούτου, όλες οι μέθοδοι επιστρέφουν αποτελέσματα τα οποία είναι παρεμφερή.

Ένα άλλο πρόβλημα είναι ότι δεν έχουμε κάποιο σύνολο παραδειγμάτων για σύγκριση. Δηλαδή, δεν διαθέτουμε κάποια ζεύγη εισόδων – επιθυμητών εξόδων, ώστε να μπορέσουμε να εκπαιδεύσουμε το δίκτυο μας. Επομένως η μόνη εκτίμηση που μπορούσε να γίνει ήταν εμπειρική, με το μάτι, μετά από πειραματισμούς και αλλαγές παραμέτρων.

Συμπερασματικά, όλες οι μέθοδοι παρήγαγαν παρόμοια αποτελέσματα, τα οποία είναι ταυτόχρονα και ικανοποιητικά. Για την τελική παρουσίαση επιλέχθηκε ο αλγόριθμος K-Means, γιατί ως μέθοδος ταιριάζει θεωρητικά καλύτερα στα δεδομένα του συγκεκριμένου προβλήματος, και τα αποτελέσματα του ήταν σχετικά καλύτερα, κατά την προσωπική μας γνώμη.

3.3 Παρουσίαση Αποτελεσμάτων

Έχοντας ολοκληρώσει την ομαδοποίηση των διανυσμάτων, κρατάμε την κατανομή τους στα clusters, δηλαδή σε ποιο cluster ανήκει το κάθε διάνυσμα.

Στη συνέχεια, υπολογίζουμε την ομοιότητα συνημίτονου (cosine similarity) ανάμεσα σε όλα τα διανύσματα. Η ομοιότητα συνημίτονου προσφέρει μια ενδιαφέρουσα οπτική γωνία. Ενώ σε καμία περίπτωση δεν μπορεί να αντικαταστήσει τη διαδικασία της συσταδοποίησης, παρουσιάζει μια σχέση συνάφειας ανάμεσα στα διανύσματα. Είναι δυνατόν, και θα φανεί και στα αποτελέσματα της εφαρμογής, δυο διανύσματα να είναι πιο «κοντά» από οποιοδήποτε άλλο ζεύγος διανυσμάτων, και παρόλα αυτά να ανήκουν σε διαφορετικά clusters. Αυτό δεν σημαίνει ότι κάποια από τις δυο μεθόδους κάνει λάθος, αλλά ότι απλά υπολογίζουν διαφορετικά πράγματα. Θα χρησιμοποιήσουμε την ομοιότητα συνημίτονου με αυτό λοιπόν τον τρόπο στην παρουσίαση των αποτελεσμάτων, ως έναν ξεχωριστό δείκτη, άσχετο με τη συσταδοποίηση.

Για την πιο σωστή και εύχρηστη αναπαράσταση των αποτελεσμάτων, δημιουργήθηκε η δομή δεδομένων "Cluster", η οποία όπως υποδηλώνει και το όνομα της, περιέχει όλες τις πληροφορίες κάποιου cluster που προκύπτει από τη διαδικασία της συσταδοποίησης. Τα πεδία αυτής της δομής είναι :

- Μια λίστα με τα id των πλάνων που ανήκουν στο cluster.
- Μια λίστα με τα διανύσματα χαρακτηριστικών των προαναφερθέντων πλάνων.
- Μια λίστα με τις 10 δημοφιλέστερες ετικέτες του cluster. Αυτές είναι τα ονόματα των 10 χαρακτηριστικών, τα οποία έχουμε κρατήσει όπως είχαμε πει από τη διαδικασία εξαγωγής χαρακτηριστικών και δημιουργίας διανυσμάτων, τα οποία εμφανίζονται τις περισσότερες φορές στα διανύσματα του cluster. Δηλαδή τα 10 πιο κοινά χαρακτηριστικά ανάμεσα στα πλάνα που ομαδοποιήθηκαν στο cluster αυτό. Αυτές οι ετικέτες βρίσκονται για κάθε cluster από μια συνάρτηση που υλοποιήσαμε με όνομα «top10».

Σε αυτή τη δομή δεδομένων βασίζεται η βάση δεδομένων που δημιουργήθηκε για την ιστοσελίδα παρουσίασης των αποτελεσμάτων.

Τα δεδομένα που περνάμε στη βάση δεδομένων είναι τα παρακάτω:

- Τα clusters που προέκυψαν από τη συσταδοποίηση. Για το καθένα περνάμε το id του, τα πλάνα που περιέχει και τις 10 δημοφιλέστερες ετικέτες του.
- Τα πλάνα της ταινίας. Για το καθένα περνάμε το id του τη χρονική στιγμή αρχής και τέλους του. Επίσης, τα id των 5 πλάνων τα οποία είναι τα πιο κοντινά σε αυτό, κατά τον δείκτη της Ομοιότητας Συνημίτονου, δηλαδή τα 5 πλάνα με το μικρότερο μέτρο Ομοιότητας Συνημίτονου για το συγκεκριμένο πλάνο. Τα 5 αυτά πλάνα τα βρίσκουμε με την συνάρτηση «min5» που υλοποιήσαμε. Τέλος, για το κάθε πλάνο περνάμε στη βάση το όνομα μιας εικόνας. Αυτή η εικόνα είναι ένα χαρακτηριστικό screenshot από την ταινία παρμένο τη χρονική στιγμή που αντιστοιχεί στην έναρξη του πλάνου, και ως εκ τούτου διαφορετικό για κάθε πλάνο.

Για να πάρουμε τα screenshots που αναφέρθηκαν προηγουμένως, χρησιμοποιήθηκε το πρόγραμμα FFmpeg. Το FFmpeg είναι ένας γρήγορος μετατροπέας βίντεο και ήχου, και υποστηρίζει όλα τα δημοφιλή formats, καθώς και ζωντανή μετάδοση. Επίσης μπορεί να κάνει μετατροπές ανάμεσα σε αυθαίρετους ρυθμούς

δειγματοληψίας και να αλλάξει διαστάσεις ενός βίντεο απευθείας, με χρήση ενός πολυφασικού φίλτρου υψηλής ποιότητας.

Στην εργασία αυτή βέβαια το ζήτημα για το οποίο το FFmpeg χρησιμοποιήθηκε ήταν απλό σε σχέση με τα παραπάνω. Κανονικά το πρόγραμμα τρέχει από το command window των Windows, αλλά εμείς το χρησιμοποιήσαμε μέσω του Eclipse με τον ακόλουθο κώδικα:

```
String src = "C:/ffmpeg/BangkokDangerous.mp4";
String folderpth = "C:/ffmpeg";

for (int i = 0;i<ls.size();i++){
    Date d  = df.parse(ls.get(i));
    Calendar cal = Calendar.getInstance();
    cal.setTime(d);
    cal.add(Calendar.SECOND, 25);
    String time = df.format(cal.getTime());

    String cmd="C:/ffmpeg/bin/ffmpeg -ss " + time +" -i "+
            src  +"  -y  -f  image2  -vcodec  mjpeg  -vframes  1
            "+folderpth+"/"+ String.format("%04d", i) +".jpeg";


    Process p = Runtime.getRuntime().exec(cmd,null,new
            File("C:/ffmpeg/bin"));
```

Αυτό το τμήμα κώδικα εκτελεί εντολές του CMD μέσω Java, δημιουργώντας διεργασίες Process με την τελευταία εντολή. Αυτές οι εντολές είναι διαδοχικές κλήσεις του FFmpeg, μια για κάθε πλάνο. Οι παράμετροι κλήσης του FFmpeg είναι ίδιες για όλα τα πλάνα:

- Η παράμετρος “-i” δηλώνει ότι η πηγή του βίντεο είναι ο όρος που ακολουθεί (src).
- Η παράμετρος “-y” διαγράφει προηγούμενα αρχεία εξόδου χωρίς επιβεβαίωση.
- Η παράμετρος “-f image2” δηλώνει ότι τα outputs θα είναι εικόνες.
- Η παράμετρος “-vcodec mjpeg” αφορά την κωδικοποίηση του βίντεο.
- Η παράμετρος “-vframes 1” δηλώνει ότι το output θα είναι ένα frame του βίντεο.

Οι μόνες παράμετροι που αλλάζουν σε κάθε κλήση είναι το όνομα του αρχείου εξόδου, δηλαδή του screenshot, το οποίο έχει το όνομα του id του κάθε πλάνου, και η παράμετρος “-ss” που δηλώνει τη χρονική στιγμή που θα ληφθεί το screenshot. Οι χρονικές στιγμές για όλα τα πλάνα βρίσκονται σε μια λίστα, η οποία είχε κρατηθεί προηγουμένως, κατά της εξαγωγή χαρακτηριστικών των πλάνων. Η μόνη διαφορά είναι ότι για κάθε πλάνο, στη χρονική στιγμή που δηλώνεται στη λίστα προσθέτουμε ένα offset 25 δευτερολέπτων, γιατί το συγκεκριμένο βίντεο της ταινίας που είχαμε διαθέσιμο διαφέρει κατά τόσο από αυτό στο οποίο βασίστηκε το RDF αρχείο που είχαμε ως δεδομένο.

4

Γραφική απεικόνιση αποτελεσμάτων

Για να παρουσιαστούν τα δεδομένα της εφαρμογής στον χρήστη, δημιουργήθηκε μια web εφαρμογή, η οποία αντλεί πληροφορίες από τη βάση δεδομένων που αναφέραμε στο προηγούμενο κεφάλαιο. Η web εφαρμογή αναπτύχθηκε με HTML, CSS και Javascript, ενώ για τη δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε το Neo4j.

Η Neo4j είναι μια βάση δεδομένων για γράφους, ανοιχτού λογισμικού, υλοποιημένη σε java. Αναπτύχθηκε από την Neo Technology Inc., και το version 1.0 κυκλοφόρησε το Φεβρουάριο του 2010. Η Neo4j απηθηκεύει τα δεδομένα σε γράφους αντί για πίνακες. Είναι η δημοφιλέστερη graph database.

Στη Neo4j, τα πάντα αποθηκεύονται με μορφή ακμής, κόμβου ή γνωρίσματος (attribute). Κάθε κόμβος και ακμή μπορεί να έχει απεριόριστο αριθμό από γνωρίσματα. Και οι κόμβοι και οι ακμές μπορούν να χαρακτηριστούν με ετικέτες. Οι ετικέτες είναι χρήσιμες γιατί επιτρέπουν τον περιορισμό της περιοχής αναζήτησης σε συγκεκριμένα labels.

Η Cypher είναι μια δηλωτική γλώσσα ερωτημάτων για γράφους που χρησιμοποιείται από τη βάση Neo4j. Επιτρέπει να γίνονται αποδοτικά τα ερωτήματα στη βάση και η ενημέρωση των γράφων. Πολύπλοκα queries μπορούν να εκφραστούν εύκολα μέσω της Cypher.

Το REpresentational State Transfer (REST) είναι ένα στυλ αρχιτεκτονικής λογισμικού που αποτελείται από κατευθυντήριες γραμμές και πρακτικές για τη δημιουργία επεκτάσιμων υπηρεσιών διαδικτύου. Το REST είναι ένα σύνολο περιορισμών που εφαρμόζεται στη σχεδίαση των μερών σε ένα διαμοιρασμένο σύστημα υπερμέσων (υπερμέσα: graphics, audio, video, plain text and hyperlinks) που μπορεί να οδηγήσει σε μια πιο αποδοτική και εύκολα συντηρήσιμη αρχιτεκτονική.

Το REST είναι πια ευρέως αποδεκτό σαν μια απλούστερη εναλλακτική αντί του SOAP.

Τα συστήματα RESTful επικοινωνούν συνήθως μέσω του πρωτοκόλλου HTTP με τα ίδια HTTP verbs (GET, POST, PUT, DELETE...) που χρησιμοποιούν και οι web browsers για να λάβουν σελίδες του διαδικτύου και να στείλουν δεδομένα σε απομακρυσμένους servers.

Το αρχιτεκτονικό στυλ REST αναπτύχθηκε από το W3C Technical ARchitecture Group παράλληλα με το HTTΠ 1.1. Ο παγκόσμιος ιστός αναπαριστά τη μεγαλύτερη υλοποίηση ενός συστήματος που συμμορφώνεται με το REST.

Εφαρμογή σε υπηρεσίες ιστού (web)

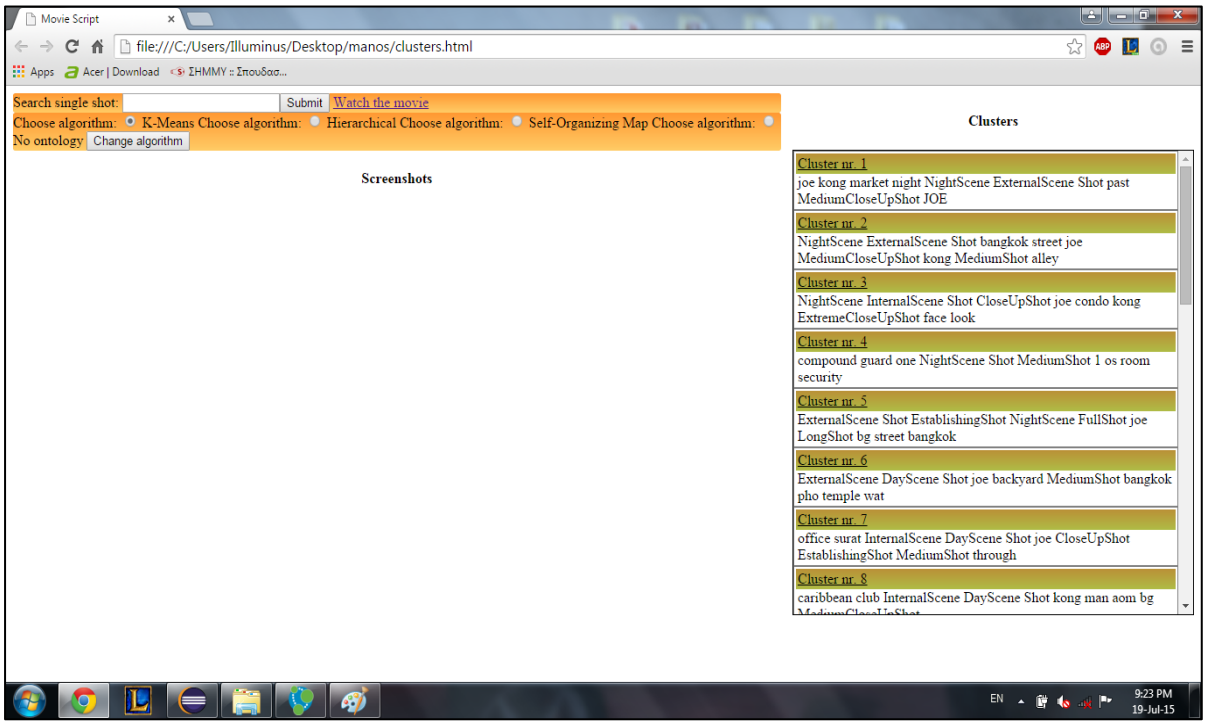
Τα APIs υπηρεσιών ιστού που συμμορφώνονται προς τους περιορισμούς της αρχιτεκτονικής REST ονομάζονται RESTful APIs. Τα RESTful APIs που βασίζονται στο πρωτόκολλο HTTP προσδιορίζονται από τις ακόλουθες πτυχές:

- base URI, όπως `http://example.com/resources/`
- ένα τύπο Internet media για τα δεδομένα. Ο τύπος είναι συνήθως JSON, αλλά μπορεί να είναι οποιοσδήποτε άλλος τύπος internet media (πχ XML, Atom, microformats, images, κλπ.)
- μεθόδους HTTP (πχ GET, PUT, POST, ή DELETE)
- συνδέσμους υπερκειμένου που αναφέρουν κατάσταση
- συνδέσμους υπερκειμένου προς σχετικούς πόρους

Η βάση neo4j δέχεται και επιστρέφει με το REST API της δεδομένα σε τύπο JSON.

Στη συνέχεια παρουσιάζονται οι διάφορες όψεις της ιστοσελίδας, όπως τις βλέπει κάποιος χρήστης.

Η αρχική όψη της web εφαρμογής φαίνεται στην παρακάτω εικόνα.

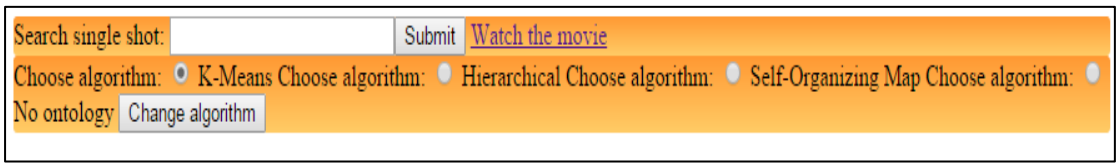


Εικόνα 27. Όψη (1) της web εφαρμογής.

Όπως βλέπουμε, το πάνω πορτοκαλί τμήμα περιέχει:

- Μια μπάρα αναζήτησης, όπου ο χρήστης μπορεί να ψάξει ένα συγκεκριμένο πλάνο με το id του.

- Ένας σύνδεσμος που ανοίγει ένα νέο παράθυρο στο οποίο ο χρήστης μπορεί να παρακολουθήσει την ταινία.
- Δυνατότητα επιλογής του αλγορίθμου συσταδοποίησης, τα αποτελέσματα του οποίου θα παρουσιάσει η σελίδα. Οι επιλογές είναι, όπως αναλύσαμε και στο προηγούμενο κεφάλαιο, οι αλγόριθμοι K-Means, SOM και Hierarcical. Υπάρχει και μια τέταρτη επιλογή, που δείχνει τα αποτελέσματα της συσταδοποίησης στην περίπτωση που δεν είχε ληφθεί υπόψιν η οντολογία κατά τη δημιουργία των διανυσμάτων, κάτι που θα παρουσιαστεί στο επόμενο κεφάλαιο.



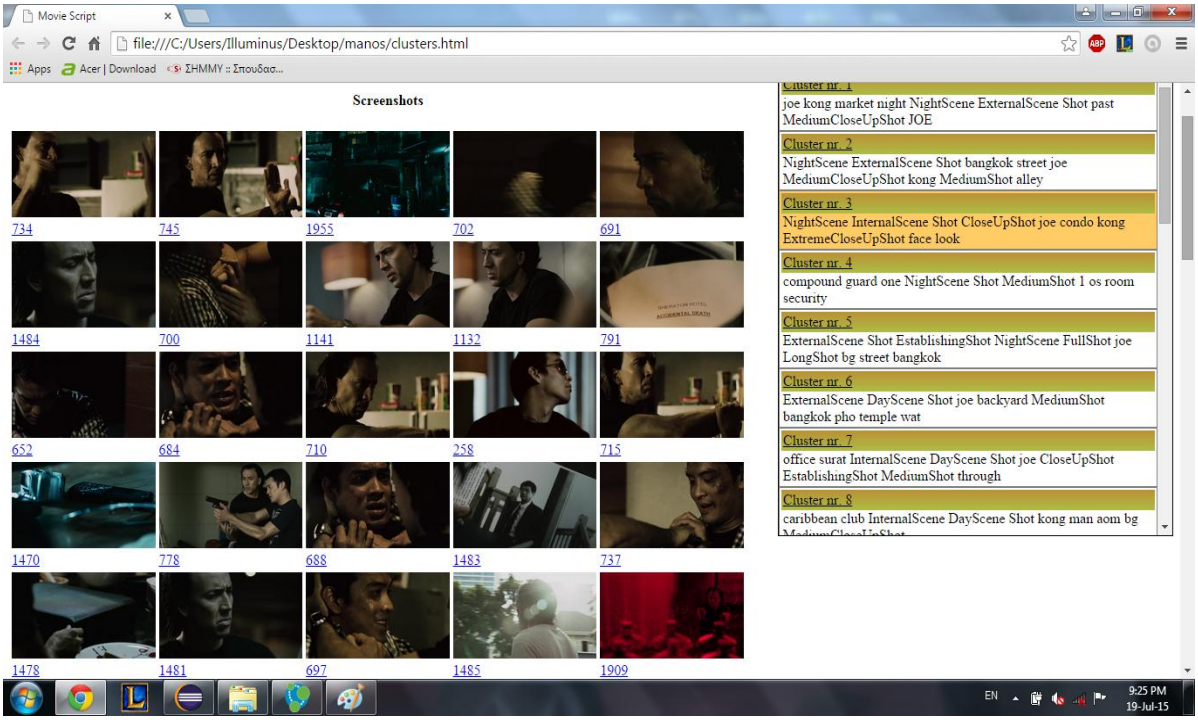
Εικόνα 28. Μπάρα επιλογών ιστοσελίδας

Στα δεξιά υπάρχει η λίστα με όλα τα clusters. Η λίστα αυτή προφανώς είναι διαφορετική ανάλογα με τον αλγόριθμο συσταδοποίησης που επιλέγεται. Η default επιλογή είναι ο αλγόριθμος K-Means, καθώς έδωσε τα πιο ικανοποιητικά αποτελέσματα όπως αναλύσαμε στο προηγούμενο κεφάλαιο.

Εικόνα 29. Λίστα των clusters, με επιλογή K-Means.

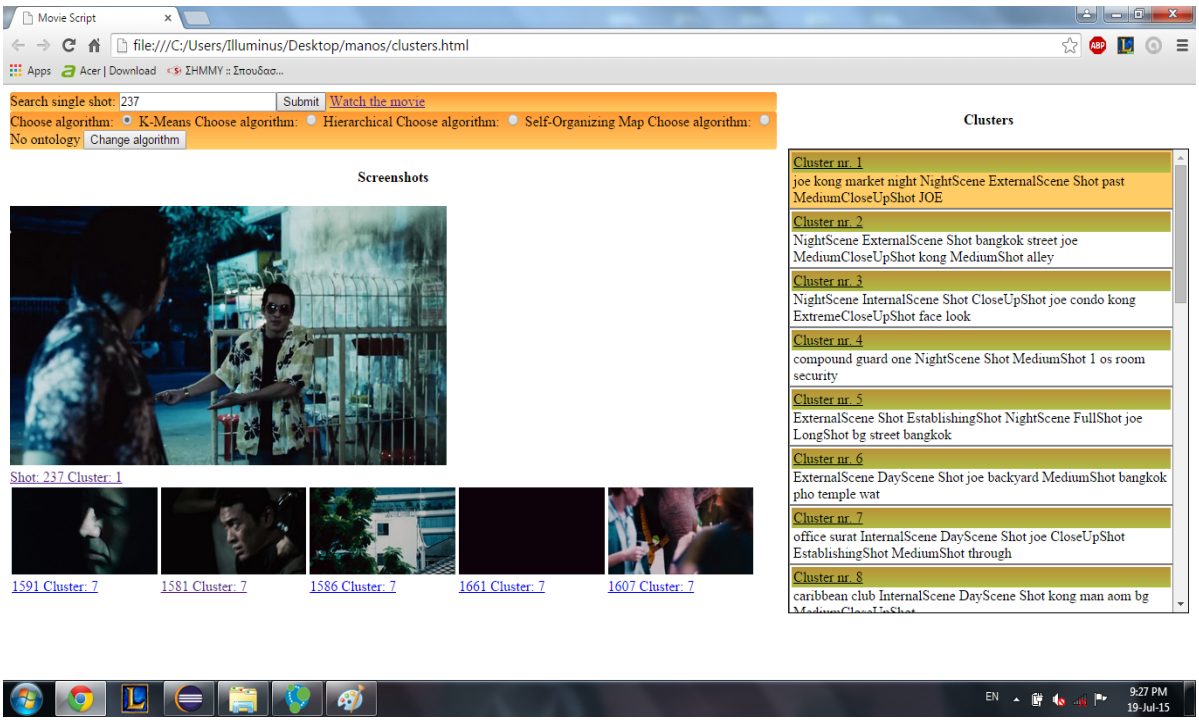
Στη λίστα αυτή, εμφανίζονται τα clusters με το id τους, και κάτω από το καθένα οι χαρακτηριστικές ετικέτες του.

Όταν ο χρήστης επιλέξει κάποιο cluster, εμφανίζονται όλα τα πλάνα που ανήκουν σε αυτό, με το id και το χαρακτηριστικό screenshot τους.



Εικόνα 30. 'Οψη (2) της web εφαρμογής. Επιλογή cluster.

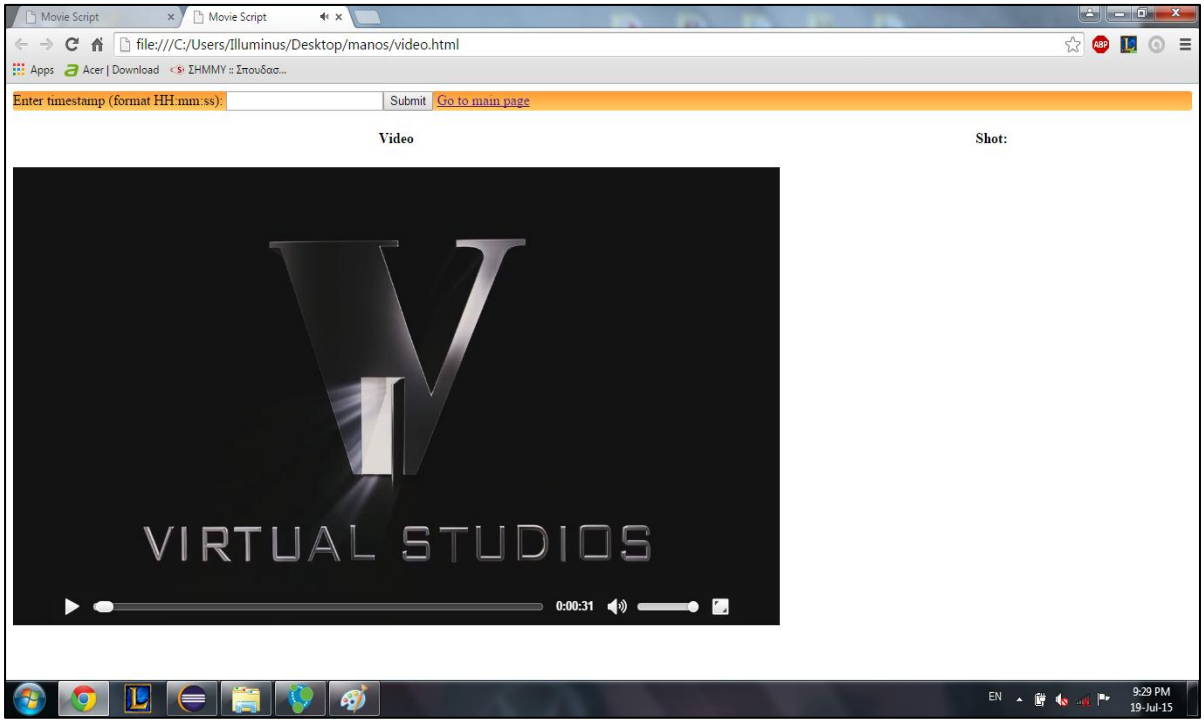
Στην περίπτωση που ο χρήστης αναζητήσει ένα συγκεκριμένο πλάνο με το id του από την μπάρα αναζήτησης, βλέπει το εξής.



Εικόνα 31. 'Οψη (3) της web εφαρμογής. Αναζήτηση πλάνου.

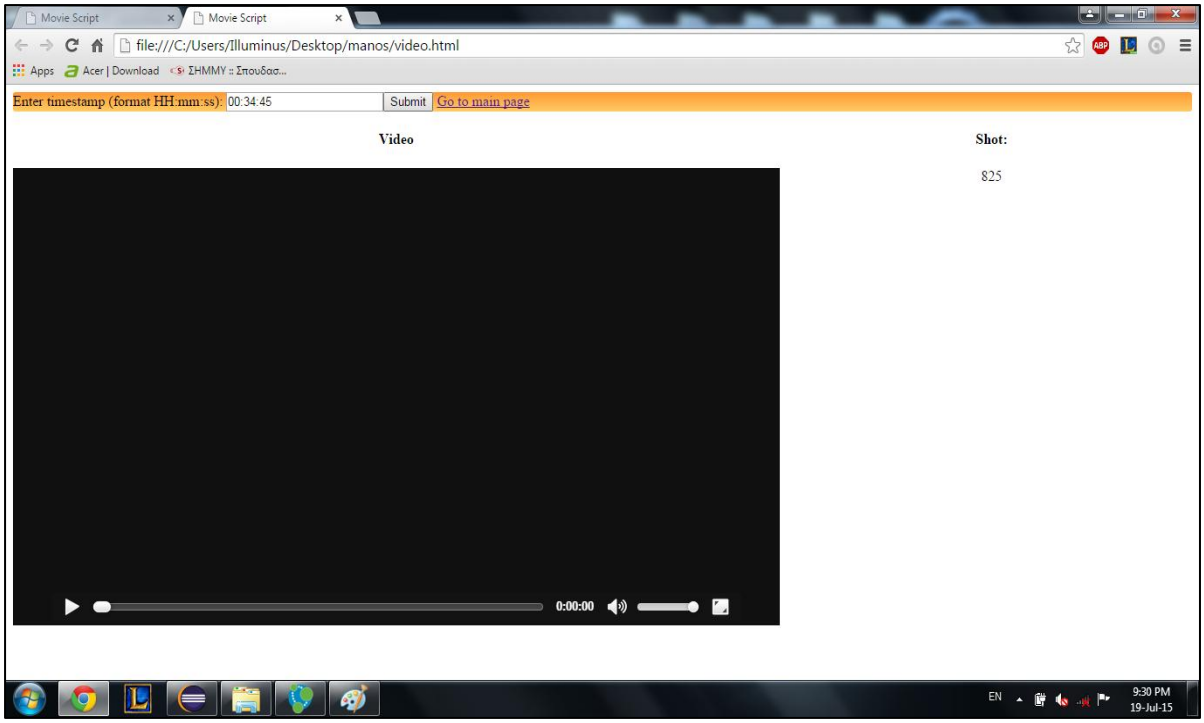
Όπως βλέπουμε, για το πλάνο εμφανίζεται το χαρακτηριστικό του screenshot, και από κάτω το id του πλάνου, και το id του cluster στον οποίο αυτό ανήκει. Τέλος, από κάτω εμφανίζονται παρομοίως, αλλά με μικρότερα thumbnails τα 5 κοντινότερα πλάνα βάσει του δείκτη Ομοιότητας Συνημίτονου.

Ωστόσο, είναι δύσκολο για τον χρήστη να γνωρίζει το id του κάθε πλάνου για το οποίο θέλει να αναζητήσει πληροφορίες. Για να αντιμετωπιστεί αυτό το πρόβλημα, ο χρήστης μπορεί να επιλέξει το σύνδεσμο "Watch the movie", που ανοίγει την ακόλουθη νέα καρτέλα.



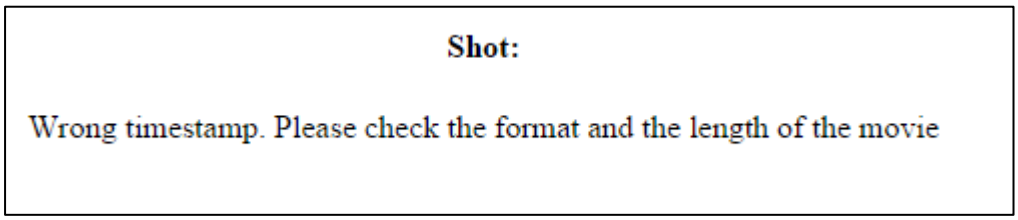
Εικόνα 32. 'Οψη (4) της web εφαρμογής. Καρτέλα ταινίας.

Σε αυτή την καρτέλα, ο χρήστης μπορεί να παρακολουθήσει ολόκληρη την ταινία σε ανάλυση 720p, με δυνατότητα πλήρους οθόνης. Αυτό τον διευκολύνει να εντοπίσει κάποιο συγκεκριμένο πλάνο. Στη συνέχεια μπορεί να χρησιμοποιήσει μια νέα μπάρα αναζήτησης. Σε αυτήν, μπορεί να εισαγάγει τη χρονική στιγμή στην οποία εντόπισε το πλάνο που τον ενδιαφέρει, και να δει το id του πλάνου αυτού.



Εικόνα 33. 'Οψη (5) της web εφαρμογής. Αναζήτηση id πλάνου.

Για την αναζήτηση αυτή ο χρήστης πρέπει να εισαγάγει τη χρονική στιγμή με συγκεκριμένο format, το HH:mm:ss (όπου HH=Ώρες, mm=Λεπτά, ss=Δευτερόλεπτα), κάτι που δηλώνεται καθαρά στην μπάρα αναζήτησης. Σε περίπτωση που ο χρήστης εισάγει κάποιο invalid format, η μια χρονική στιγμή μεγαλύτερη από το μήκος της ταινίας, εμφανίζεται ένα μήνυμα λάθους.



Εικόνα 34. Μήνυμα λάθους στην αναζήτηση πλάνου με timestamp.

Αυτή ήταν η web εφαρμογή που επιτρέπει στο χρήστη να δει όλα τα αποτελέσματα της εφαρμογής με ευδιάκριτο, εύχρηστο και απλό τρόπο.

5

Σύγκριση με αποτελέσματα αλγορίθμου που δεν αξιοποιεί την οντολογία

Τα αποτελέσματα της εφαρμογής που παρουσιάσαμε στο προηγούμενο κεφάλαιο είναι ικανοποιητικά, αλλά δεν απαντούν από μόνα τους στο αρχικό ερώτημα που τέθηκε σε αυτή την εργασία. Στόχος ήταν να εξεταστεί αν η χρήση των οντολογικών πληροφοριών των δεδομένων οδηγεί σε καλύτερα αποτελέσματα, απ' ότι αν τα αγνοήσουμε.

Δυστυχώς όπως είπαμε και σε προηγούμενο σημείο της εργασίας, δεν υπάρχει κάποιο σύνολο επαλήθευσης για τη συσταδοποίηση, δηλαδή ένα σύνολο δεδομένων μαζί με τις εξακριβωμένα σωστές ομαδοποιήσεις τους, ώστε να μπορέσουμε να ελέγξουμε την ορθότητα των δικών μας αποτελεσμάτων. Επομένως αυτό που μπορούμε να κάνουμε είναι να εξετάσουμε τα αποτελέσματα, όπως εμφανίζονται στο περιβάλλον της web εφαρμογής που δημιουργήσαμε. Γι αυτό το σκοπό, υλοποιήθηκε η ίδια εφαρμογή, με τι διαφορά ότι κατά την εξαγωγή χαρακτηριστικών και τη δημιουργία των διανυσμάτων, οι πληροφορίες της οντολογία δεν λήφθηκαν υπόψιν. Με άλλα λόγια, τα διανύσματα που περιγράφουν τα πλάνα βασίζονται μόνο στην κειμενική πληροφορία των δεδομένων. Στην συσταδοποίηση χρησιμοποιήθηκε, για ομοιομορφία στα αποτελέσματα, αλγόριθμος K-means με τιμή k=25.

Καταρχάς, μπορούμε να εξετάσουμε τις δημοφιλείς ετικέτες όλων των clusters των δύο περιπτώσεων, με και χωρίς οντολογικές πληροφορίες, που ακολουθούν. Παρατηρούμε ότι όταν χρησιμοποιείται η οντολογία, οι όροι/ετικέτες που προέρχονται από αυτήν εμφανίζονται στις σημαντικότερες ετικέτες σχεδόν κάθε cluster. Δηλαδή, σαν χαρακτηριστικά είναι καθοριστικά για την ομαδοποίηση των πλάνων, και μια από τα πιο συνηθισμένες ιδιότητες που μοιράζονται τα πλάνα της κάθε ομάδας είναι οι οντολογικές κλάσεις που ανήκουν από κοινού.

Αντίθετα, όταν δεν χρησιμοποιείται η οντολογία, οι δημοφιλέστερες ετικέτες αφορούν μόνο τους διαλόγους και τις περιγραφές των πλάνων, ωστόσο δεν μπορούν να διακρίνουν κάποια σχέση πέρα από αυτή.

Στα 2 παρακάτω πλαίσια εμφανίζονται κατά σειρά οι δημοφιλέστερες ετικέτες των clusters, για χρήση ή όχι της οντολογίας αντίστοιχα.

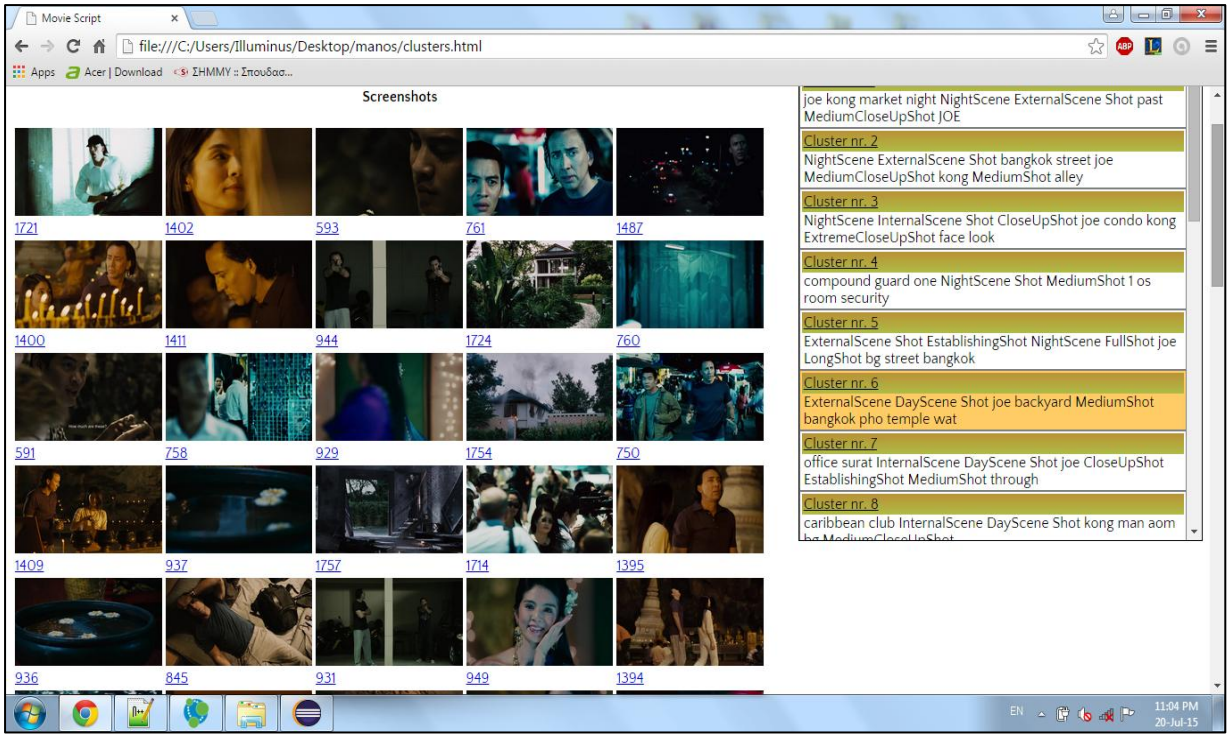
Cluster 1: "joe","kong","market","night","NightScene","ExternalScene","Shot","past",
"MediumCloseUpShot","JOE"
Cluster 2: "NightScene","ExternalScene","Shot","bangkok","street","joe","MediumCloseUpShot",
"kong","MediumShot","alley"
Cluster 3: "NightScene","InternalScene","Shot","CloseUpShot","joe","condo","kong",
"ExtremeCloseUpShot","face","look"
Cluster 4: "compound","guard","one","NightScene","Shot","MediumShot","1","os","room",
"security"
Cluster 5: "ExternalScene","Shot","EstablishingShot","NightScene","FullShot","joe","LongShot",
"bg","street","bangkok"
Cluster 6: "ExternalScene","DayScene","Shot","joe","backyard","MediumShot","bangkok","pho",
"temple","wat"
Cluster 7: "office","surat","InternalScene","DayScene","Shot","joe","CloseUpShot",
"EstablishingShot","MediumShot","through"
Cluster 8: "caribbean","club","InternalScene","DayScene","Shot","kong","man","aom","bg",
"MediumCloseUpShot"
Cluster 9: "NightScene","InternalScene","Shot","joe","condo","kong","JOE","KONG",
"CloseUpShot","MediumCloseUpShot"
Cluster 10: "bell","cathedral","tower","NightScene","ExternalScene","Shot","joe","CloseUpShot",
"scope","gun"
Cluster 11: "compound","NightScene","ExternalScene","Shot","joe","CloseUpShot",
"MediumShot","car","LowAngleShot","gun"
Cluster 12: "bangkok","NightScene","ExternalScene","Shot","patpong","kong","KONG","man",
"larry","CloseUpShot"
Cluster 13: "pharmacy","NightScene","InternalScene","Shot","joe","fon","CloseUpShot",
"ExtremeCloseUpShot","past","MediumCloseUpShot"
Cluster 14: "NightScene","InternalScene","Shot","joe","warehouse","MediumShot",
"EstablishingShot","guard","to","MediumCloseUpShot"
Cluster 15: "NightScene","Shot","club","InternalScene","caribbean","kong","aom","room",
"MediumCloseUpShot","MediumShot"
Cluster 16: "ExternalScene","Shot","kong","joe","KONG","DayScene","JOE","condo","do","i"
Cluster 17: "NightScene","ExternalScene","Shot","joe","CloseUpShot","MediumCloseUpShot",
"bangkok","marketplace","open","fon"
Cluster 18: "NightScene","ExternalScene","Shot","joe","MediumShot","fon","to","house",
"park","move"
Cluster 19: "InternalScene","DayScene","Shot","joe","condo","floor","second","bathroom",
"thug","CloseUpShot"
Cluster 20: "bang","pha","river","ExternalScene","DayScene","Shot","joe","boat","target","white"
Cluster 21: "room","NightScene","InternalScene","Shot","hotel","joe","anton","CloseUpShot",
"past","MediumShot"
Cluster 22: "restaurant","NightScene","ExternalScene","Shot","joe","fon","past","CloseUpShot",
"MediumCloseUpShot","JOE"
Cluster 23: "NightScene","InternalScene","Shot","house","fon","CloseUpShot","joe",
"grandmother","to","room"
Cluster 24: "condo","joe","NightScene","ExternalScene","Shot","kong","MediumCloseUpShot",
"past","JOE","car"
Cluster 25: "NightScene","InternalScene","Shot","surat","CloseUpShot","aran","office",
"restaraunt","SURAT","car"

Cluster 1: "joe","kong","market","night","past","?","money","eating","keep","some"
Cluster 2: "bangkok","joe","street","fon","marketplace","open","pho","temple","wat","car"
Cluster 3: "condo","joe","knife","kong","throat","against","with","hand","press","blade"
Cluster 4: "compound","guard","security","room","2","monitor","one","os","they","1"
Cluster 5: "to","joe","it","back","as","lt","rt","tilt","bg","move"
Cluster 6: "kong","alley","bangkok","thug","street","bg","run","-lsb-","-rsb-","look"
Cluster 7: "warehouse","joe","guard","as","bottle","move","bg","curtain","fg","plastic"
Cluster 8: "club","kong","to","caribbean","aom","back","bg","room","briefcase","face"
Cluster 9: "joe","kong","condo","past","?","face","to","look","man","door"
Cluster 10: "bell","cathedral","tower","joe","scope","gun","through","pov","past","look"
Cluster 11: "compound","car","guard","surat","from","move","shooter","aran","as","outside"
Cluster 12: "bangkok","patpong","kong","man","larry","go","joe","tall","tourist","past"
Cluster 13: "pharmacy","joe","fon","past","look","coworker","smile","face","to","car"
Cluster 14: "joe","man","bold","mercedes","bg","fg","building","lsb","rsb","look"
Cluster 15: "club","caribbean","kong","aom","room","girl","private","past","smile","back"
Cluster 16: "joe","do","kong","condo","not","i","it","get","past","backyard
Cluster 17: "park","joe","fon","to","fg","move","look","mugger","os","as"
Cluster 18: "compound","joe","gun","hallway","lt","face","fg","past","point","room"
Cluster 19: "joe","surat","office","pov","through","scope","cross-hair","rifle","eye","window"
Cluster 20: "bang","pha","river","ExternalScene","DayScene","Shot","joe","boat","target","white"
Cluster 21: "hotel","room","joe","anton","past","hand","to","arm","gloved","door"
Cluster 22: "restaurant","fon","joe","past","bowl","it","laugh","smile","dark","eat"
Cluster 23: "house","fon","joe","grandmother","to","room","court","interrogation","hand",
"window"
Cluster 24: "condo","joe","floor","second","thug","to","look","bathroom","face","lit"
Cluster 25: ""surat","office","aran","car","past","restaraunt","to","kong","sniper","bg"

Επίσης κοιτάζοντας στην ιστοσελίδα τα πλάνα που περιέχονται σε κάθε cluster, με τη βοήθεια και των id και των χαρακτηριστικών screenshots των πλάνων, μπορούμε να παρατηρήσουμε καλύτερα αποτελέσματα όταν χρησιμοποιήθηκαν οι οντολογικές πληροφορίες. Αρχικά υπάρχει πιο ομοιόμορφη κατανομή των πλάνων στα clusters. Επιπλέον, φαίνεται καθαρά η επίδραση των οντολογικών πληροφοριών στη συσταδοποίηση. Δηλαδή, τα πλάνα του cluster κατά συντριπτική πλειοψηφία μοιράζονται όντως τα οντολογικά χαρακτηριστικά που εμφανίζονται στις ετικέτες του.

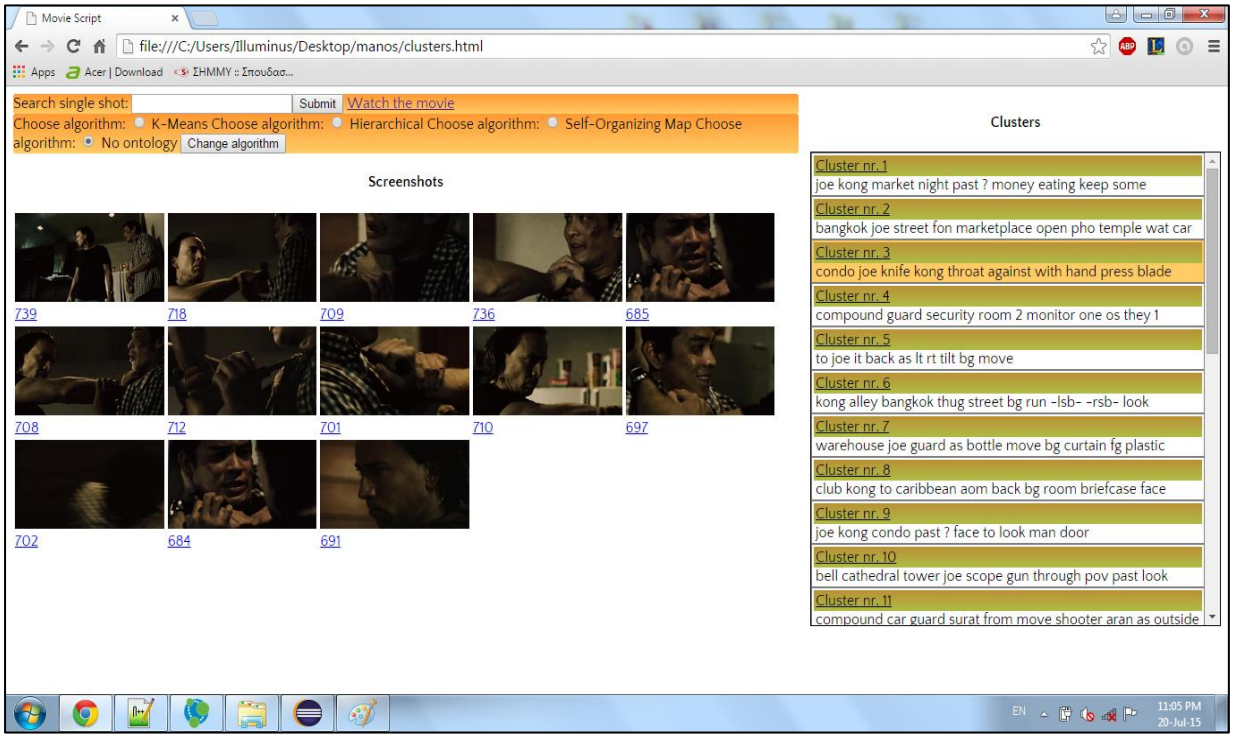
Αντίθετα, στα αντίστοιχα αποτελέσματα της μη-χρήσης οντολογίας, παρατηρούμε ότι συχνά τα πλάνα ενός cluster περιέχουν μόνο συνεχόμενα πλάνα. Αυτό εξηγείται καθώς αυτά τα συνεχόμενα πλάνα έχουν de facto κάποιες κοινές κειμενικές πληροφορίες, την περιγραφή της σκηνής στην οποία όλα ανήκουν. Πέρα από αυτά όμως δεν γίνεται κάποια ομαδοποίηση με πλάνα από άλλα σημεία της ταινίας καθώς οι κειμενικές πληροφορίες δεν δίνουν περαιτέρω κοινά χαρακτηριστικά.

Ένα τέτοιο παράδειγμα δίνεται παρακάτω. Οι εικόνες είναι παρμένες από την web εφαρμογή της εργασίας, και δείχνουν τα σημεία που τονίσαμε παραπάνω.



Εικόνα 35. Παράδειγμα αποτελέσματος με χρήση οντολογίας.

Βλέπουμε λοιπόν, ότι για τον cluster 6, σχεδόν όλα τα πλάνα που φαίνονται στην παραπάνω εικόνα όντως φαίνονται να είναι πλάνα εξωτερικά, τραβηγμένα μέρα και από μεσαία απόσταση, όπως δηλώνουν οι ετικέτες του cluster ExternalScene, DayScene και MediumShot αντίστοιχα. Αυτό γίνεται ενώ πολλά πλάνα αυτά δεν είναι διαδοχικά ή κοντινά, αλλά διάσπαρτα μέσα στην ταινία.



Εικόνα 36. Παράδειγμα αποτελέσματος χωρίς χρήση οντολογίας.

Στην δεύτερη εικόνα αντίθετα, όπου δεν έχει χρησιμοποιηθεί καμία οντολογική πληροφορία, βλέπουμε ότι τα πλάνα που ανήκουν στον cluster 3, είναι όλα κοντινά πλάνα της ίδιας σκηνής, και για αυτό μοιράζονται τις ετικέτες του, οι οποίες όπως μπορεί να υποτεθεί (και όντως να διαπιστωθεί με μια πρόχειρη έρευνα στα RDF δεδομένα), είναι λέξεις που απλά περιέχονται στην περιγραφή της σκηνής, στην οποία όλα τα πλάνα αυτά ανήκουν.

6

Επίλογος

Για να συνοψίσουμε την εργασία αυτή, αποδείχθηκε ότι η αποτελεσματικότητα της ομαδοποίησης ενός συνόλου ημιδομημένων δεδομένων βελτιώθηκε αισθητά, με τη χρήση ακόμα και μιας απλής οντολογίας η οποία περιγράφει τα δεδομένα αυτά. Μπορεί να φανταστεί κάποιος πόσο μεγαλύτερη θα είναι η βελτίωση αυτή για δεδομένα που είναι πολύ μεγαλύτερα σε όγκο, και για οντολογίες που περιγράφουν πολύ περισσότερες οντότητες, ακόμα πιο λεπτομερώς.

Αν αναλογιστεί κανείς ότι ο Σημασιολογικός Ιστός είναι το μέλλον στις τεχνολογίες γνώσης, γίνεται εύκολα κατανοητό ότι η αξιοποίηση όλων αυτών των σημασιολογικών πληροφοριών θα έχει αμέτρητες εφαρμογές σε πολλούς και διαφορετικούς τομείς της επιστήμης. Το μόνο που χρειάζεται είναι να κατασκευασθούν οι οντολογίες που θα περιγράφουν δεδομένα κάθε είδους, από οντότητες στο διαδίκτυο, μέχρι αντικείμενα και πρόσωπα στον πραγματικό κόσμο. Μετά από αυτό οι δυνατότητες είναι απεριόριστες.

Η παρούσα εργασία μπορεί εύκολα να επεκταθεί ή να χρησιμοποιηθεί ως βοήθημα για αυτό τον σκοπό, με απλές μετατροπές του περιεχομένου της ώστε να υποστηρίζει όλες τις προαναφερθείσες οντολογίες.

Βιβλιογραφία

[1] Simon Haykin, *Neural Networks and Learning Machines, Third Edition* (2010).

[2] G. Fung, *A Comprehensive Overview of Basic Clustering Algorithms* (2001).

[3] D.L.McGuinness, D.Nardi, P.F.Patel-Schneider, *The Description Logic Handbook: Theory, implementation, and applications*.

[4] Resource Description Framework (RDF) Model and Syntax Specification (1999). Διαθέσιμο στο δικτυακό τόπο: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[5] C.Rogers, L.Laura, *Sams Teach Yourself Java 6 in 21 Days, Fifth Edition* (2007).

[6] Apache Jena Documentation. Διαθέσιμο στο δικτυακό τόπο: https://jena.apache.org/documentation/index.html

[7] C.D.Manning, M.Surdeanu, J.Bauer, J.Finkel, S.J.Bethard, D.McClosky, *The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60* (2014).

[8] D.T.Pham, S.S.Dimov, C.D.Nguyen, *Selection of K in K-means Clustering* (2004). Διαθέσιμο στο δικτυακό τόπο: http://www.ee.columbia.edu/~dpwe/papers/PhamDN05-kmeans.pdf

[9] MATLAB Documentation (2015). Διαθέσιμο στο δικτυακό τόπο: http://www.mathworks.com/help/matlab/

[10] FFmpeg Documentation. Διαθέσιμο στο δικτυακό τόπο: http://ffmpeg.org/documentation.html

[11] The Neo4j Manual v2.2.3. Διαθέσιμο στο δικτυακό τόπο: http://neo4j.com/docs/2.2.3/