Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης

Διπλωματικη Εργασια

Αναγνώριση Συμβάντων σε Κοινωνικά Δίκτυα, με χρήση τεχνικών Επεξεργασίας Φυσικής Γλώσσας και Μηχανικής Μάθησης

Ιωαννης Γ. Παπαμιχαηλ

Επιβλέπουσα: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Αθήνα, Μάρτιος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Αναγνώριση Συμβάντων σε Κοινωνικά Δίκτυα, με χρήση τεχνικών Επεξεργασίας Φυσικής Γλώσσας και Μηχανικής Μάθησης

Διπλωματικη Εργασια

Ιωάννης Γ. Παπαμιχαήλ

Επιβλέπουσα: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Μαρτίου 2019.

.....	.....	.....
Θεοδώρα Βαρβαρίγου	Εμμανουήλ Βαρβαρίγος	Συμεών Παπαβασιλείου
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

.....
Ιωάννης Γ. Παπαμιχαήλ
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ένα σύστημα αναγνώρισης συμβάντων, σε κοινωνικά δίκτυα, αποτελείται από δύο κύρια στάδια. Το πρώτο στάδιο, αφορά την προεπεξεργασία των συλλεχθέντων δεδομένων. Η προεπεξεργασία επιτυγχάνεται με τη χρήση διαφόρων τεχνικών επεξεργασίας κειμένου, φυσικής γλώσσας και μεταδεδομένων. Στο επόμενο στάδιο, επιλέγεται ο αλγόριθμος που θα συντελέσει στην ανάλυση συστάδων των δεδομένων και ορίζεται η συνάρτηση ομοιότητας, στην οποία θα βασιστεί ο αλγόριθμος αυτός, για τον υπολογισμό των αποστάσεων μεταξύ των δεδομένων, στο χώρο αναπαράστασής τους. Στα συστήματα αναγνώρισης συμβάντων που υπάρχουν σήμερα, η συλλογή δεδομένων, για την εκπαίδευση και αξιολόγησή τους, γίνεται με τη χρήση των API, τα οποία παρέχονται από τις διαδικτυακές εφαρμογές των κοινωνικών δικτύων. Μέχρι στιγμής, έχουν προταθεί αρκετές διαφορετικές προσεγγίσεις όσον αφορά τους αλγορίθμους που επιλέγονται για την ομαδοποίηση των δεδομένων. Εντούτοις, κρίνεται επιτακτική ανάγκη η υλοποίηση ενός συστήματος, το οποίο θα μπορεί να εφαρμοστεί σε διαφορετικούς τύπους συνόλων δεδομένων, διατηρώντας υψηλή ακρίβεια για κάθε έναν από αυτούς.

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος αναγνώρισης συμβάντων, βασισμένο σε δεδομένα κοινωνικών δικτύων, το οποίο, αφενός, θα επιτρέπει μια σχετική ευελιξία στην χρήση διαφορετικών συνόλων δεδομένων (Κοινωνικών Δικτύων), και, αφετέρου, θα επιτυγχάνει υψηλή ακρίβεια στην ομαδοποίησή τους σε συστάδες.

Λέξεις Κλειδιά

σύστημα αναγνώρισης συμβάντων, σύστημα αναγνώρισης συμβάντων βασισμένο σε κοινωνικά δίκτυα, ομαδοποίηση δεδομένων, ανάλυση συστάδων, επεξεργασία φυσικής γλώσσας, ομοιότητα αναρτήσεων κοινωνικών δικτύων

Abstract

An event detection system based on social media, consists of two main steps. The first step concerns the pre-processing of data collections. Preprocessing is accomplished using various methods of processing text and metadata. In the next step, we select the clustering algorithm which will help on analyzing the data and we define the similarity function on which this algorithm will be based to calculate the distances between the data in their representation space. In existing event detection systems, data collection, used to train and evaluate the system, is achieved through the APIs provided by social networking applications. So far, several different approaches have been proposed concerning the selected algorithm used for clustering the data. However, it is imperative to implement a system that can be applied to different types of data sets, maintaining high accuracy for each of them.

This diploma thesis aims to develop a social network based event detection system that allows flexibility in the use of different data collections (Social Networks) and achieves high precision in cluster analysis.

Keywords

event detection system, event detection on social media, clustering, cluster analysis, natural language processing, posts similarity

στους γονείς μου

Ευχαριστίες

Θα ήθελα, καταρχάς, να ευχαριστήσω την καθηγήτρια κ. Βαρβαρίγου, για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο της. Επίσης, ευχαριστώ ιδιαίτερα τον Δρ. Γιώργο Παλαιοκρασσά, για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου, για την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Η εποχή που διανύουμε, γνωστή και ως ψηφιακή εποχή, χαρακτηρίζεται από την ευκολία μετάδοσης πληροφοριών. Ένας από τους πιο καθοριστικούς παράγοντες που συντέλεσε στην πραγματοποίηση της ψηφιακής επανάστασης είναι ο Παγκόσμιος Ιστός. Ο Παγκόσμιος Ιστός αποτελεί έναν ανθρωποκεντρικό χώρο διακίνησης τεράστιου όγκου πληροφοριών, με τα Κοινωνικά Δίκτυα να καταλαμβάνουν ένα πολύ σημαντικό τμήμα του, με ιδιαίτερη δυναμική.

Τα Κοινωνικά Δίκτυα, των οποίων η καθιέρωση αποτελεί πραγματικότητα και είναι το κυρίαρχο μέσο ανθρώπινης επικοινωνίας, συμβάλλουν στην πραγματοποίηση ενός συνόλου αλληλεπιδράσεων και διαπροσωπικών σχέσεων και επιτρέπουν τη διεπαφή ανάμεσα στους χρήστες. Η μετάδοση της πληροφορίας γίνεται είτε μέσω γραπτού κειμένου, είτε μέσω οπτικοακουστικού περιεχομένου.

Μία πολύ συχνή πρακτική, που φαίνεται να έχει ολοένα και μεγαλύτερη απήχηση στη σύγχρονη κοινωνία, συγκριτικά με προγενέστερες, είναι η χρήση σχολίων που θίγουν τα κοινωνικοπολιτικά δρώμενα καθώς και η παρουσίαση γεγονότων εξ ολοκλήρου από προσωπική, υποκειμενική σκοπιά, υπό τη μορφή προσωπικής εμπειρίας. Το παραπάνω φαινόμενο, σε συνδυασμό με την απήχηση των μέσων αυτών στους χρήστες, τα καθιστά ιδανικό τρόπο αμφίδρομης ενημέρωσης. Επομένως, το ενδιαφέρον των Μέσων Μαζικής Ενημέρωσης έχει στραφεί στα Κοινωνικά Δίκτυα ως μία επιπλέον πηγή πληροφοριών.

Προκειμένου να επιτευχθεί η επεξεργασία και κατά συνέπεια η χρήση της διατιθέμενης πληροφορίας, δεδομένου του μεγάλου όγκου της, κρίνεται απαραίτητη η χρήση υπολογιστικών μηχανημάτων. Ωστόσο, η παράλειψη του ανθρώπινου παράγοντα στην διαδικασία της επεξεργασίας δεδομένων αποτελεί αναγκαία αλλά μη επαρκή συνθήκη. Αυτο συμβαίνει γιατί, παρά την διαθέσιμη υπολογιστική ισχύ, εμφανίζονται τεχνικές δυσκολίες, οι οποίες είναι αναγκαίο να αντιμετωπιστούν. Συνοπτικά, η συνεχής αύξηση της ροής των δεδομένων από τους χρήστες (Big Data) και ο ανθρωποκεντρικός τους χαρακτήρας, αποτελούν μερικά από τα τεχνικά εμπόδια που επιβάλλεται να υπερνικήσουμε, ώστε να καταστεί εφικτή και αποδοτική η επεξεργασία. Το ίδιο ισχύει και σχετικά με την αξιολόγηση και την σταχυολόγηση της ανεξέλεγκτης πληροφορίας (Noise), καθώς επίσης και με την έλλειψη ενιαίας δομής που παρέχουν τα Κοινωνικά Δίκτυα στον τρόπο αποθήκευσης και παρουσίασης των δεδομένων.

Ένας τρόπος αξιοποίησης της τελικής πληροφορίας είναι η Αναγνώριση Συμβάντων (Event

Detection). Ως συμβάν ορίζεται μία ακολουθία αναρτήσεων με κοινά χαρακτηριστικά, οι οποίες ξεπερνούν κάποιο επιθυμητό όριο ομοιότητας. Η ομαδοποίηση των αναρτήσεων αυτών μπορεί να υλοποιηθεί με ποικίλους τρόπους. Για την μεγιστοποίηση της απόδοσης ενός συστήματος αναγνώρισης συμβάντων παρατηρούμε ότι στις περισσότερες περιπτώσεις είναι αναγκαία η "κατανόηση" του περιεχομένου από τις υπολογιστικές μηχανές. Η κατανόηση, ακολούθως, επιτυγχάνεται με χρήση τεχνικών επεξεργασίας φυσικής γλώσσας (Natural Language Processing) και Μηχανικής Μάθησης (Machine Learning).

1.1 Αντικείμενο της διπλωματικής

Τα τελευταία χρόνια, παρατηρείται αύξηση του ενδιαφέροντος των ερευνητών προς το πεδίο της Αναγνώρισης Συμβάντων, δεδομένου του ότι η εκμετάλλευσή του προσφέρει πληθώρα οφελών. Η αυτοματοποίηση της διαδικασίας εύρεσης ειδήσεων και η ταχύτερη ενημέρωση του κοινού είναι κάποια από αυτά. Για την Αναγνώριση Συμβάντων απαιτείται η εφαρμογή τεχνικών "Ανάλυσης Δεδομένων". Η επικρατέστερη μέθοδος Ανάλυσης Δεδομένων, που χρησιμοποιείται, ονομάζεται "Μηχανική Μάθηση" (Machine Learning). Μέχρι στιγμής έχουν προταθεί αρκετές διαφορετικές προσεγγίσεις από την επιστημονική κοινότητα, όσον αφορά τον τρόπο πραγμάτωσης ενός τέτοιου συστήματος. Κοινή παραδοχή αποτελεί ο διαχωρισμός του προβλήματος σε δύο βασικές κατηγορίες. Για τη διεξαγωγή και επαλήθευση των αποτελεσμάτων κάθε έρευνας κρίνεται απαραίτητη η συλλογή, η κατηγοριοποίηση -από τον άνθρωπο- και η αποθήκευση των δεδομένων σε "σύνολα δεδομένων" (Datasets). Στη συνέχεια, τα σύνολα δεδομένων διαχωρίζονται σε εκείνα που θα χρησιμοποιηθούν κατά την διαδικασία εκπαίδευσης του μοντέλου ανάλυσης, στο οποίο βασίζεται το σύστημα (Train Data) και σε εκείνα που θα χρησιμοποιηθούν για την αξιολόγηση της απόδοσης του εκάστοτε συστήματος (Test Data).

1. Η πρώτη κατηγορία κατατάσσει το πρόβλημα ως πρόβλημα κατηγοριοποίησης (Classification Problem). Κατά την διαδικασία της κατηγοριοποίησης, το υποσύνολο των υπό δοκιμή δημοσιεύσεων των χρηστών αντιστοιχίζεται σε προκαθορισμένες κατηγορίες, οι οποίες έχουν προκύψει από το υποσύνολο των υπό εκπαίδευση δεδομένων.

2. Η δεύτερη κατηγορία κατατάσσει το πρόβλημα ως "πρόβλημα ομαδοποίησης" (Clustering Problem). Σε αυτή την περίπτωση, τα υπό δοκιμή δεδομένα ομαδοποιούνται με τέτοιο τρόπο ώστε αυτά που ανήκουν στην ίδια ομάδα (Cluster), να ξεπερνούν κάποιο επιθυμητό βαθμό ομοιότητας. Ο βαθμός ομοιότητας καθορίζεται κατά την διαδικασία εκπαίδευσης του μοντέλου και η επιλογή της συνάρτησης ομοιότητας έγκειται στην ευχέρεια του εκάστοτε ερευνητή.

Οι δύο παραπάνω κατηγορίες εξυπηρετούν διαφορετικούς σκοπούς. Η μεταξύ τους διαφορά συνοψίζεται στο γεγονός ότι η κατηγοριοποίηση (Classification) διατηρεί σταθερό πλήθος ομάδων, ανεξάρτητα από το μέγεθος της εισόδου δεδομένων, ενώ στην ομαδοποίηση (Clustering) το πλήθος των ομάδων εξαρτάται από το μέγεθος της εισόδου.

Αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη ενός συστήματος αναγνώρισης συμβάντων, το οποίο υπόκειται στη δεύτερη κατηγορία και κάνει χρήση παραμετροποιημένων αλγορίθμων ομαδοποίησης (Clustering Algorithms).

Στόχος της εργασίας αποτελεί η μελέτη, η επιλογή και η δημιουργία των κατάλληλων μέσων, με σκοπό την επίτευξη υψηλής αποδοτικότητας και ακρίβειας του νέου συστήματος. Βασική αρχή, κατά τη διαδικασία υλοποίησης του συστήματος, υπήρξε η δυνατότητα χρήσης του, χωρίς να είναι απαραίτητη η επέμβαση στον πηγαίο κώδικα, καθώς και η δυνατότητα λειτουργίας του με "σύνολα δεδομένων" (Datasets) επιλογής του κάθε χρήστη.

1.2 Οργάνωση του κειμένου

Η παρούσα διπλωματική οργανώνεται σε 7 κεφάλαια.

Το **Κεφάλαιο 1** περιέχει την εισαγωγή και την ανάλυση του αντικειμένου της έρευνας που διεξήχθηκε.

Το **Κεφάλαιο 2** αναλώνεται στην παρουσίαση όλων των θεωρητικών γνώσεων που απαιτούνται για την κατανόηση της μετέπειτα υλοποίησης, καθώς και των εργαλείων που χρησιμοποιήθηκαν.

Το **Κεφάλαιο 3** περιγράφει τη διαδικασία συλλογής και αποθήκευσης των απαραίτητων δεδομένων σε σύνολα δεδομένων, επεξηγώντας, παράλληλα, τον τρόπο λειτουργίας των API.

Το **Κεφάλαιο 4** επεξηγεί τον τρόπο που διαβάζονται τα διαφορετικής μορφής σύνολα δεδομένων από το σύστημα, καθώς και τις συναρτήσεις που τα επεξεργάζονται, για την εξαγωγή των κατάλληλων πληροφοριών που θα χρησιμοποιηθούν από τους αλγορίθμους ανάλυσης συστάδων.

Το **Κεφάλαιο 5** αναλύει τους αλγορίθμους και τις τεχνικές ομαδοποίησης που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος αναγνώρισης συμβάντων.

Το **Κεφάλαιο 6** περιγράφει την πειραματική διάταξη όπου πραγματοποιήθηκαν οι μετρήσεις, παρουσιάζει τα αποτελέσματα και παραθέτει μία σύντομη εκτίμησή τους.

Τέλος, στο **Κεφάλαιο 7** συγκεντρώνονται τα συμπεράσματα της εργασίας και προτείνονται μελλοντικές επεκτάσεις της.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά οι τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη του συστήματος, με σκοπό την προετοιμασία του αναγνώστη της εργασίας για την κατανόηση της μετέπειτα υλοποίησης.

2.1 Επεξεργασία Φυσικής Γλώσσας

Η επεξεργασία φυσικής γλώσσας (NLP) αποτελεί ένα υποπεδίο της επιστήμης των υπολογιστών (Computer Science), της μηχανικής πληροφοριών (Information Engineering) και της τεχνητής νοημοσύνης (Artificial Intelligence), που σχετίζεται με την αλληλεπίδραση μεταξύ των ηλεκτρονικών υπολογιστών και των ανθρώπων. Οι αλληλεπιδράσεις αυτές επιτυγχάνονται με τη χρήση φυσικών γλωσσών. Πιο συγκεκριμένα, το κύριο ενδιαφέρον του υποπεδίου αυτού είναι η επεξεργασία και η ανάλυση μεγάλου όγκου δεδομένων φυσικής γλώσσας. Καθώς οι έξυπνες συσκευές εξαπλώνονται σε ολοένα και περισσότερους τομείς, παρατηρείται ραγδαία αύξηση της απήχησής τους προς το κοινό. Έτσι, οι ψηφιακοί βοηθοί καθιερώνονται ως ένας τρόπος επικοινωνίας και διαχείρισης όλων αυτών των συσκευών από τον άνθρωπο. Στο σημείο αυτό, γίνεται αντιληπτό ότι η αναγνώριση ομιλίας, η κατανόηση και η παραγωγή κειμένου αποτελούν κάποιες από τις προκλήσεις που δημιουργούνται κατά την επεξεργασία φυσικής γλώσσας. Παρακάτω παρουσιάζονται κάποιες από τις τεχνικές που συνέβαλαν στο να ξεπεραστούν αυτές οι δυσκολίες, για τους σκοπούς της παρούσας έρευνας.

2.1.1 Διασύνδεση Οντοτήτων - Entity Linking

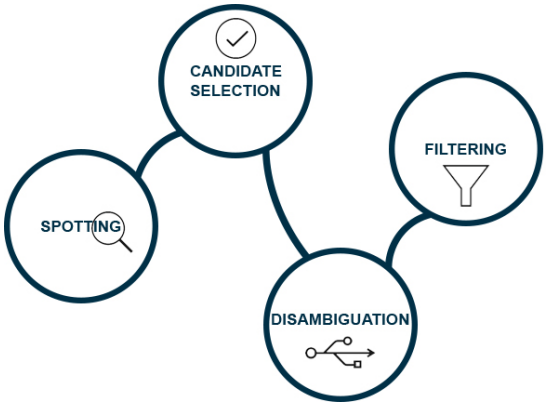
Στην επεξεργασία φυσικής γλώσσας, η διασύνδεση οντοτήτων είναι η διαδικασία αναγνώρισης της ταυτότητας κάθε οντότητας που αναφέρεται στο κείμενο που υπόκειται σε επεξεργασία. Πιο συγκεκριμένα, η χρησιμότητα αυτής της διαδικασίας, καθώς και αυτό που την διαχωρίζει από την διαδικασία "απλής" αναγνώρισης οντοτήτων, είναι το γεγονός ότι καθίσταται εφικτό να αντιληφθεί ένα υπολογιστικό μηχάνημα την ιδιότητα της συγκεκριμένης οντότητας που εξετάζει και να την ξεχωρίσει μέσα σε πλήθος ομοίων οντοτήτων. Για παράδειγμα, στην περίπτωση επεξεργασίας της πρότασης "Gates is the principal founder of Microsoft", το αποτέλεσμα της εξαγωγής οντοτήτων θα ήταν μία λίστα της μορφής ["Gates", "Microsoft"]. Το αποτέλεσμα μοιάζει αποδεκτό, αλλά τι συμβαίνει στην περίπτωση που θέλουμε να γίνει κατανοητό απ΄ το υπολογιστικό μηχάνημα, ότι το "Gates" αναφέρεται συγκεκριμένα στον ιδρυτή της Microsoft και όχι στη λέξη πύλες; Εδώ, λοιπόν, έρχεται να συμπληρώσει αυτό το κενό η διασύνδεση οντοτήτων.

Η λογική της υλοποίησης γίνεται εύκολα κατανοητή. Αρχικά, χρειαζόμαστε μία βάση γνώσης ως μέτρο σύγκρισης. Μία τέτοια θα μπορούσε να είναι αποτέλεσμα ενός μοντέλου μηχανικής μάθησης με εφαρμογή στο σύνολο των δεδομένων που εξετάζουμε ή ενός τμήματος αυτών. Ωστόσο, για τα δεδομένα της δικής μας υλοποίησης μπορούμε να χρησιμοποιήσουμε τα δεδομένα από ηλεκτρονικές εγκυκλοπαίδειες όπως η Wikipedia ή η DBpedia. Έχοντας επιλέξει, λοιπόν, την βάση γνώσης, στην οποία θα βασιστούμε, μπορούμε πλέον να προχωρήσουμε στο επόμενο βήμα, το οποίο είναι η αναγνώριση των σχέσεων ανάμεσα στις οντότητες ενός κειμένου. Το βήμα αυτό αποτελεί την αρχή διαχώρισης οντοτήτων, καθώς είναι ξεκάθαρα αντιληπτό ότι δύο έννοιες μίας πρότασης έχουν πολύ μεγάλη πιθανότητα να σχετίζονται σημασιολογικά. Στο παράδειγμά μας, ο βασικός διαχωρισμός των δύο διαφορετικών οντοτήτων "Gates" οφείλεται στη σύνδεση που φαίνεται να έχει η οντότητα αυτή με την οντότητα "Microsoft". Συγκρίνοντας, λοιπόν, τις πιθανές "ερμηνείες" της οντότητας, σε σημασιολογικό επίπεδο, με το υπόλοιπο κείμενο, έχοντας ως κριτήριό μας την βάση γνώσης που επιλέξαμε, καταλήγουμε στο συμπέρασμα ότι η οντότητα αναφέρεται στον ιδρυτή της Microsoft. Το αποτέλεσμα είναι η δημιουργία ενός συνδέσμου, που αντιστοιχίζει την οντότητα αυτή με το άρθρο της βάσης, στο οποίο κατέληξε η αντιστοιχία. Στο παράδειγμά μας, αυτός ο σύνδεσμος θα ήταν της μορφής: "https://en.wikipedia.org/wiki/Bill_Gates".

Η υλοποίηση ενός τέτοιου μοντέλου είναι δύσκολη και χρονοβόρα. Η εύρεση μίας έτοιμης υλοποίησης αποτέλεσε απαραίτητη προϋπόθεση για την εκπόνηση της εργασίας. Έπειτα από αρκετή έρευνα, καταλήξαμε στη χρήση του εργαλείου "https://www.dbpedia-spotlight.org/", το οποίο βασίζεται στην ηλεκτρονική εγυκλοπαίδεια DBpedia. Αυτό που το έκανε να ξεχωρίσει είναι η ευκολία χρήσης του, η δυνατότητα παραμετροποίησής του και η αποτελεσματικότητά του.

2.1.2 DBpedia Spotlight

Το DBpedia Spotlight [1] είναι ένα εργαλείο αυτοματοποίησης της διαδικασίας υπομνηματισμού ενός συνόλου υπαρχουσών παραπομπών μίας γνωσιακής βάσης σε ένα κείμενο. Το εργαλείο αυτό παρέχει μία λύση διασύνδεσης αδόμητων πηγών πληροφορίας στο διασυνδεδεμένο ανοιχτό σύννεφο δεδομένων (Linked Open Data Cloud) μέσω της DBpedia. Ο τρόπος λειτουργίας του βασίζεται σε μία προσέγγιση τεσσάρων βημάτων.



Σχήμα 2.1: DBpedia Steps

1. Στο πρώτο βήμα, πραγματοποιείται η ταυτοποίηση επιφανειακών μορφών υποσυνόλων κειμένου της αρχικής εισόδου (input), οι οποίες είναι πιθανές παραπομπές.

2. Στο δεύτερο βήμα, επιλέγεται ένα σύνολο από τις επιφανειακές μορφές του πρώτου βήματος μαζί με τις πηγές τους στη DBpedia, οι οποίες αποτελούν πιθανές ερμηνείες τους.

3. Στο τρίτο βήμα, αποφασίζεται ποιες είναι οι πιο πιθανές υποψήφιες πηγές για κάθε επιλεγμένη επιφανειακή μορφή.

4. Στο τέταρτο βήμα, προσαρμόζονται τα υπομνήματα στις συγκεκριμένες απαιτήσεις της εκάστοτε διεργασίας, σύμφωνα με τις παραμέτρους συστήματος που έχουν οριστεί από τον χρήστη.

2.1.3 Λεξική Ενσωμάτωση - Word Embeddings

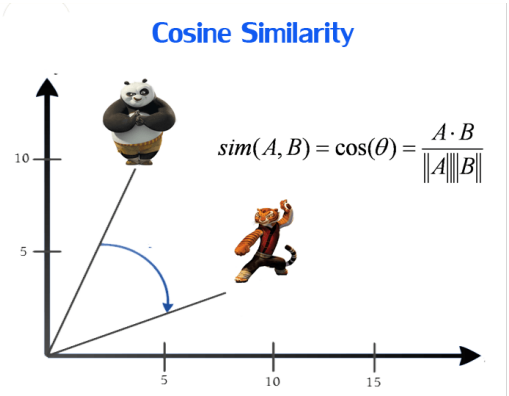
Η λεξική ενσωμάτωση είναι ένας από τους πιο διάσημους τρόπους αναπαράστασης του κειμενικού λεξιλογίου. Αυτός ο τρόπος καθιστά δυνατό τον εντοπισμό σημασιολογικής και συντακτικής ομοιότητας, την απόδοση περιεχομένου μίας λέξης και τη σχέση που μπορεί να έχει με κάποια άλλη λέξη. Ένας πρώτος ορισμός για τη λεξική ενσωμάτωση θα μπορούσε να είναι αυτός της διανυσματικής αναπαράστασης των λέξεων. Ερωτήματα που προκύπτουν, έπειτα από μία πρώτη προσπάθεια ορισμού της έννοιας, είναι το πώς σχηματίζονται οι διανυσματικές αναπαραστάσεις και το πώς μπορεί να αποδοθεί το περιεχόμενο των λέξεων.

Η Word2Vec είναι μία από τις πιο δημοφιλείς τεχνικές εκμάθησης λεξικής ενσωμάτωσης κάνοντας χρήση ρηχών νευρωνικών δικτύων (Shallow Neural Networks).

Ας λάβουμε υπόψιν μας τις δύο ακόλουθες προτάσεις με παρεμφερές νόημα: “Have a good day” και “Have a great day”. Κατασκευάζοντας έναν εξαντλητικό διανυσματικό χώρο λεξιλογίου (έστω V), θα ισχύει V = {Have, a, good, great, day}. Τώρα, ας δημιουργήσουμε ένα κωδικοποιημένο (one-hot encoded) διάνυσμα για κάθε λέξη του διανυσματικού χώρου V. Το μήκος του κωδικοποιημένου διανύσματος θα μπορούσε να είναι ίσο με το μέγεθος του διανυσματικού χώρου $V = 5$. Με τον τρόπο αυτό, θα είχαμε ένα διάνυσμα μηδενικών στοιχείων σε κάθε θέση, εκτός από το στοιχείο που αντιστοιχεί στον δείκτη που αντιπροσωπεύει την συγκεκριμένη λέξη στο λεξικό. Αυτό το στοιχείο θα ήταν μονάδα. Οι παρακάτω κωδικοποιήσεις θα δώσουν μία πιο σαφή εικόνα του παραδείγματος.

Have = [1,0,0,0,0]; a = [0,1,0,0,0]; good = [0,0,1,0,0]; great = [0,0,0,1,0]; day = [0,0,0,0,1];

Σε μία προσπάθεια απεικόνισης αυτών των κωδικοποιήσεων, μπορούμε να σκεφτούμε έναν χώρο πέντε διαστάσεων, όπου κάθε λέξη καταλαμβάνει μία εκ των διαστάσεων αυτών, διατηρώντας ανεξαρτησία από τις υπόλοιπες (δεν υπάρχει προβολή στις υπόλοιπες διαστάσεις). Αυτό έχει ως συνέπεια, οι λέξεις “good” και “great” να απέχουν όσο οι λέξεις “day” και “have”, κάτι που δεν ισχύει. Σκοπός μας είναι λέξεις με παρόμοια σημασία να έχουν σχετικά κοντινές αποστάσεις στο χώρο. Επομένως, το συνημίτονο της γωνίας που σχηματίζεται μεταξύ των διανυσμάτων, που αναπαριστούν τις λέξεις αυτές, τείνει στη μονάδα. Ειδάλλως, το συνημίτονο τείνει στο μηδέν.



Σχήμα 2.2: Cosine Similarity

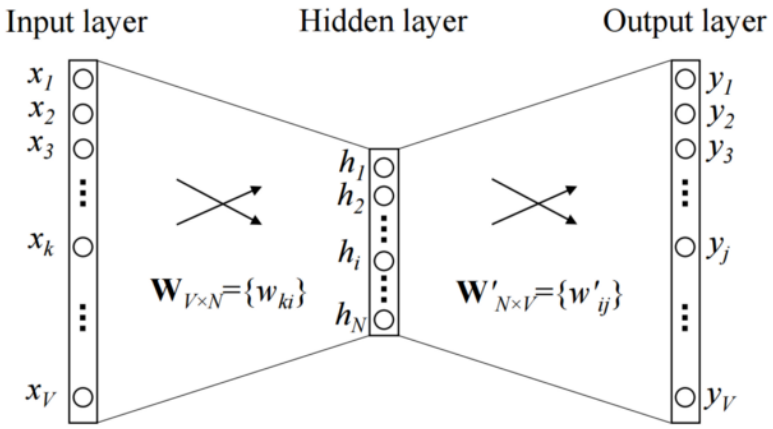
Λογικό επακόλουθο αποτελεί η ιδέα δημιουργίας κατανεμημένων αναπαραστάσεων. Διαισθητικά, εισάγουμε ένα είδος εξάρτησης μίας λέξης από τις υπόλοιπες. Οι λέξεις παρεμφερούς ερμηνείας θα λάβουν μεγαλύτερο τμήμα αυτής της εξάρτησης. Στις κωδικοποιημένες αναπαραστάσεις (one-hot encoding representations), όλες οι λέξεις είναι ανεξάρτητες μεταξύ τους, όπως αναφέραμε προηγουμένως.

Το Word2Vec είναι μία μέθοδος κατασκευής μίας τέτοιας ενσωμάτωσης. Μπορεί να υλοποιηθεί με δύο τρόπους, οι οποίοι σχετίζονται με τη χρήση νευρωνικών δικτύων. Οι δύο τρόποι ονομάζονται: Continuous Skip Gram (CSG) και Continuous Bag of Words (CBOW).

1. **CBOW Model**: Αυτή η μέθοδος παίρνει ως είσοδο το περιεχόμενο κάθε λέξης και προσπαθεί να προβλέψει με ποια λέξη αντιστοιχίζεται σημασιολογικά. Θεωρούμε το εξής παράδειγμα: “Have a great day”.

   Ας υποθέσουμε ότι η είσοδος στο νευρωνικό δίκτυο είναι η λέξη great. Παρατηρούμε ότι γίνεται προσπάθεια να προβλεφθεί μία λέξη-στόχος (day), χρησιμοποιώντας μία μοναδική λέξη εισόδου (great). Πιο συγκεκριμένα, χρησιμοποιούμε την κωδικοποίηση (one-hot encoded) της λέξης εισόδου (great) και παίρνουμε μέτρηση του σφάλματος εξόδου, συγκριτικά με την κωδικοποίηση (one-hot encoded) της λέξης-στόχου (day). Κατά την διαδικασία πρόβλεψης της λέξης-στόχου, ανακαλύπτουμε την διανυσματική αναπαράσταση της.

   Ας εξετάσουμε πιο αναλυτικά την αρχιτεκτονική της μεθόδου.



Σχήμα 2.3: Αρχιτεκτονική μοντέλου CBOW με λέξεις μονοδιάστατου περιεχομένου

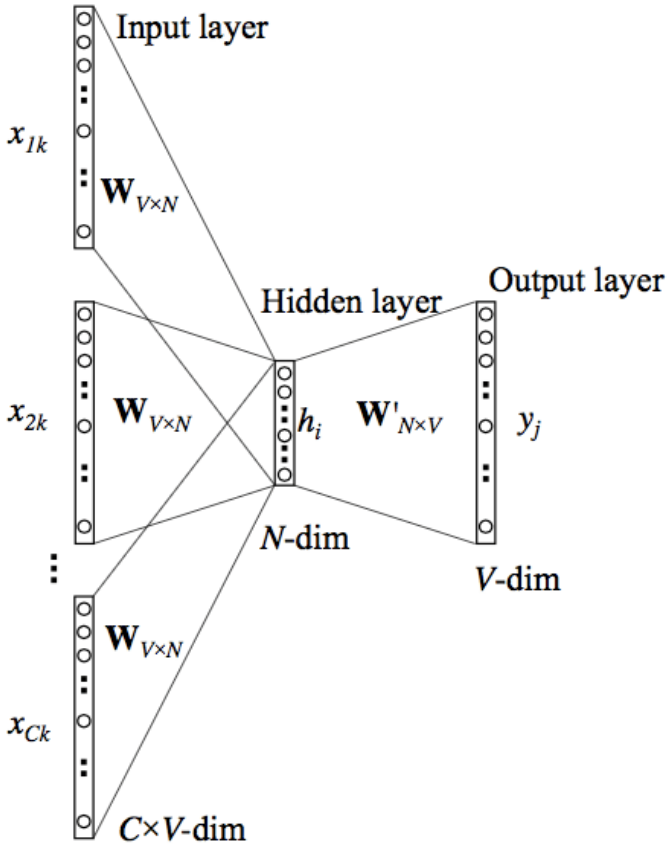
Η είσοδος είναι ένα κωδικοποιημένο (one-hot encoded) διάνυσμα μεγέθους V. Το κρυφό επίπεδο περιέχει N νευρώνες και η έξοδος είναι και αυτή ένα διάνυσμα μήκους V με στοιχεία τα αποτελέσματα της συνάρτησης Softmax.

Ας εξηγήσουμε τα σύμβολα της εικόνας:

- W_{vn} είναι ο πίνακας βαρών που αντιστοιχίζει την είσοδο x με τον $V \times N$ διαστάσεων πίνακα του κρυφού επιπέδου.
- W_{nv} είναι ο πίνακας βαρών που αντιστοιχεί τις εξόδους του κρυφού επιπέδου στον N $\times V$ διαστάσεων πίνακα του τελικού επιπέδου.

Οι νευρώνες του κρυφού επιπέδου απλά αντιγράφουν το άθροισμα των βαρών της εισόδου στο επόμενο επίπεδο. Δεν υπάρχει ενεργοποίηση, όπως στις συναρτήσεις sigmoid, tanh ή ReLU. Τα μόνα που δεν παρουσιάζουν γραμμικότητα είναι οι υπολογισμοί της συνάρτησης softmax, στο επίπεδο εξόδου.

Όμως, το παραπάνω μοντέλο χρησιμοποίησε μοναδικού περιεχομένου λέξη για την πρόβλεψη του στόχου. Έχουμε τη δυνατότητα να χρησιμοποιήσουμε λέξεις πολλαπλού περιεχομένου, για να καταφέρουμε το ίδιο.

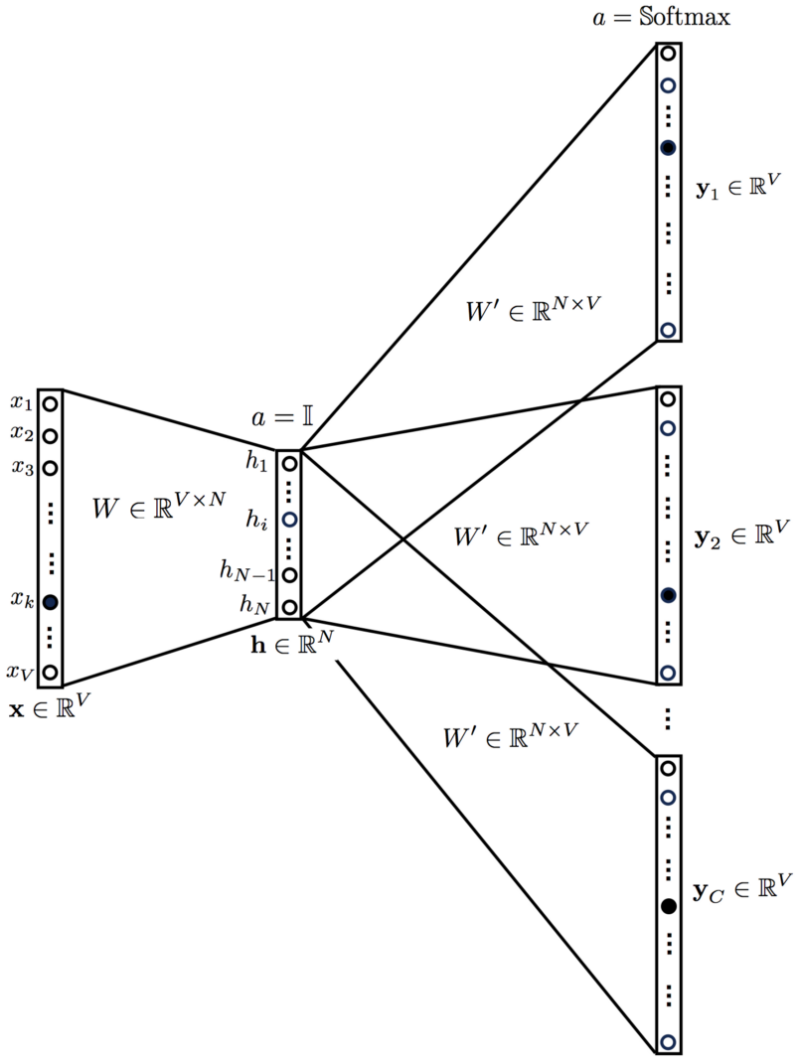


Σχήμα 2.4: Αρχιτεκτονική μοντέλου CBOW με λέξεις πολλαπλού περιεχομένου

Το παραπάνω μοντέλο δέχεται C περιεχομένου λέξεις. Όταν ο W_{vn} χρησιμοποιείται για τον υπολογισμό των εισόδων του κρυφού επιπέδου, παίρνουμε τον μέσο όρο όλων των C περιοχομένων-εισόδων της λέξης.

Έτσι, είδαμε πώς δημιουργούνται οι λεξικές αναπαραστάσεις, χρησιμοποιώντας τις ερμηνείες των λέξεων. Ωστόσο, υπάρχει ένας ακόμη τρόπος, για να καταφέρουμε το ίδιο. Μπορούμε να κάνουμε χρήση της λέξης-στόχου, της οποίας την αναπαράσταση θέλουμε να παραγάγουμε, για να προβλέψουμε το περιεχόμενο. Κατά την διαδικασία αυτή, παράγουμε τις αναπαραστάσεις. Μία άλλη παραλλαγή, η οποία ονομάζεται Skip Gram, κάνει το ίδιο.

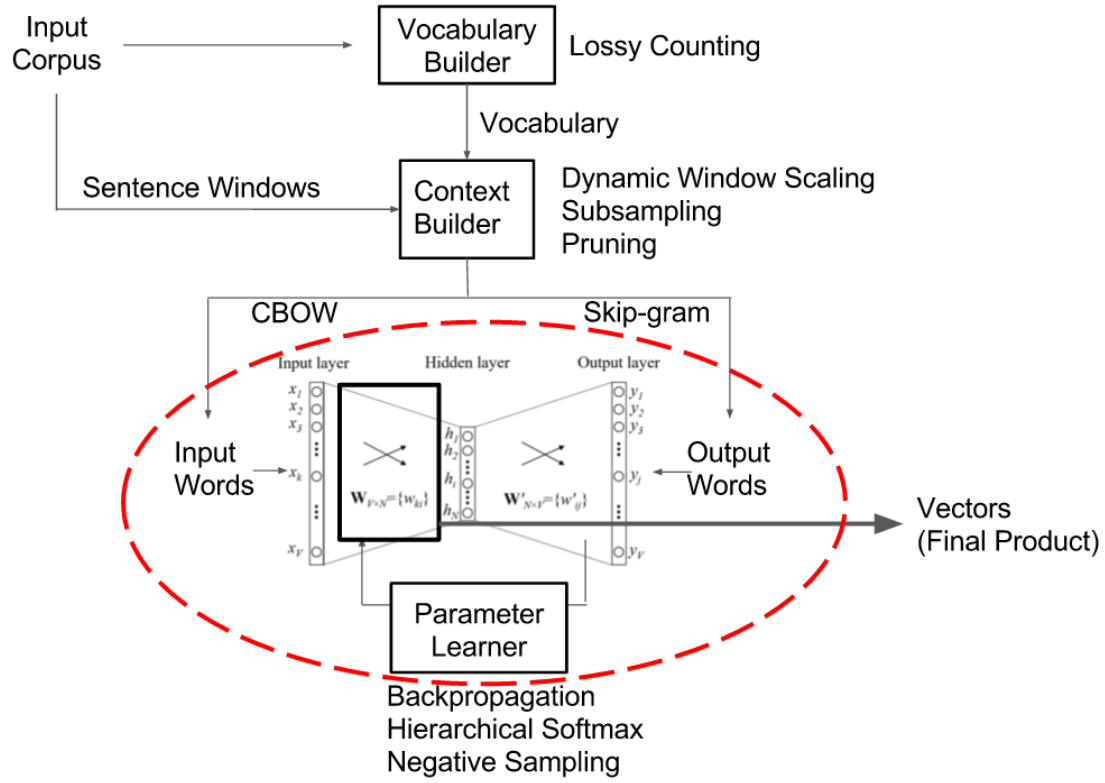
2. **Skip-Gram model**:



Σχήμα 2.5: Αρχιτεκτονική μοντέλου Skip Gram

Το μοντέλο αυτό μοιάζει με το αντίστροφο του πολλαπλού περιεχομένου CBOW μοντέλου. Εκ πρώτης όψεως, αυτό είναι αληθές. Η λειτουργία του βασίζεται στην εισαγωγή της λέξης-στόχου στο δίκτυο. Στη συνέχεια, το μοντέλο εξάγει C πιθανοτικές κατανομές από V πιθανότητες, μία για κάθε λέξη.

Συγκρίνοντας τα μοντέλα, παρατηρούμε ότι και τα δύο έχουν πλεονεκτήματα και μειονεκτήματα. Σύμφωνα με τον ερευνητή Mikolov, το μοντέλο Skip Gram λειτουργεί αποδοτικότερα με μικρό όγκο δεδομένων και αναπαριστά σπάνιες λέξεις με μεγαλύτερη ακρίβεια. Αντιθέτως, το μοντέλο CBOW είναι ταχύτερο και παρέχει πιο ακριβείς αναπαραστάσεις σε λέξεις που επαναλαμβάνονται συχνά. Ωστόσο, και στις δύο περιπτώσεις, το δίκτυο χρησιμοποιεί τον αλγόριθμο Backpropagation για εκμάθηση. Στο επόμενο σχήμα, παρουσιάζεται η συνολική διαδικασία λειτουργίας του word2vec.

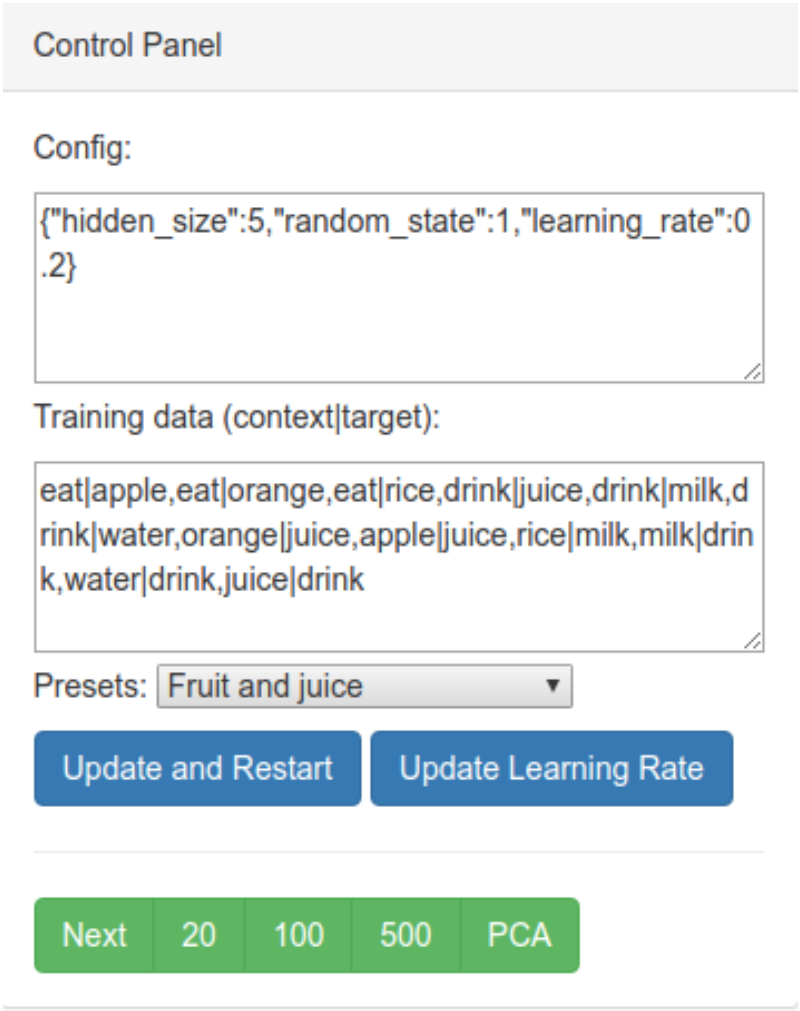


Σχήμα 2.6: Διάγραμμα λειτουργίας του Word2Vec

Τα παραπάνω αποτελούν τις βασικές αρχές της λεξικής ενσωμάτωσης. Για την επίτευξη καλύτερης χρονικής πολυπλοκότητας, μπορούμε να εφαρμόσουμε κάποιες τεχνικές, όπως η Hierarchical Softmax [2].

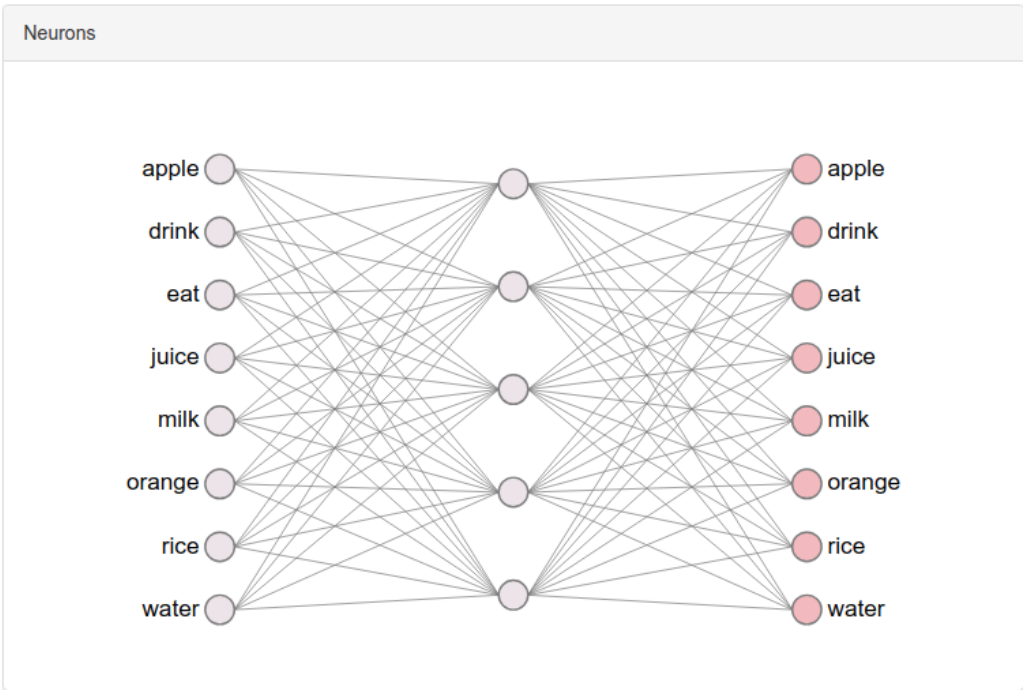
Κάνοντας χρήση του εργαλείου **wevi** [3], το οποίο είναι ένα εργαλείο εικονικής αναπαράστασης της δομής και των αποτελεσμάτων μίας διαδικασίας λεξικής ενσωμάτωσης, γίνεται καλύτερα κατανοητή η λογική στην οποία βασίζεται η διαδικασία αυτή.

Στην πρώτη εικόνα, απεικονίζονται οι ρυθμίσεις και τα δεδομένα εισόδου που χρησιμοποιήθηκαν για την εξαγωγή των πειραματικών αποτελεσμάτων του παραδείγματος.



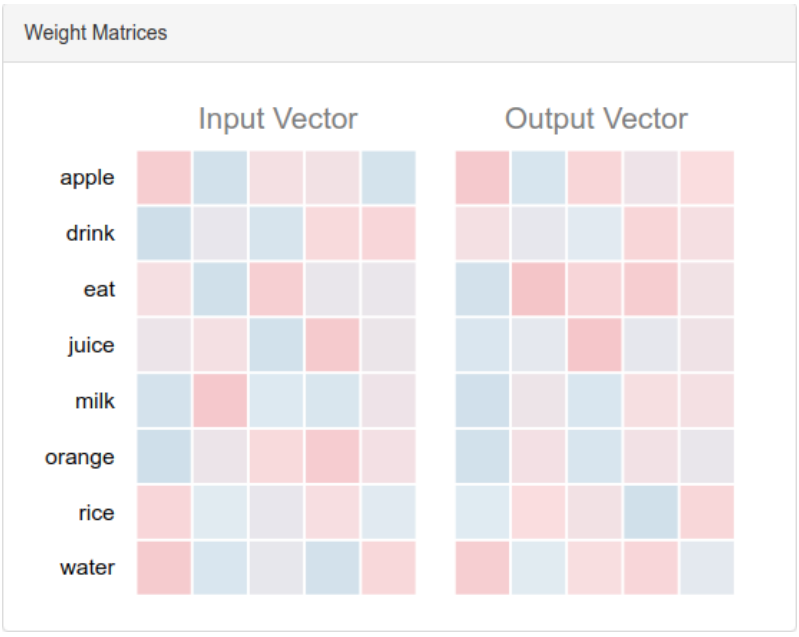
Σχήμα 2.7: Πίνακας ελέγχου του Wevi

Στη δεύτερη εικόνα, προβάλλονται οι νευρωνικές συνάψεις ανάμεσα στα δεδομένα εισόδου.



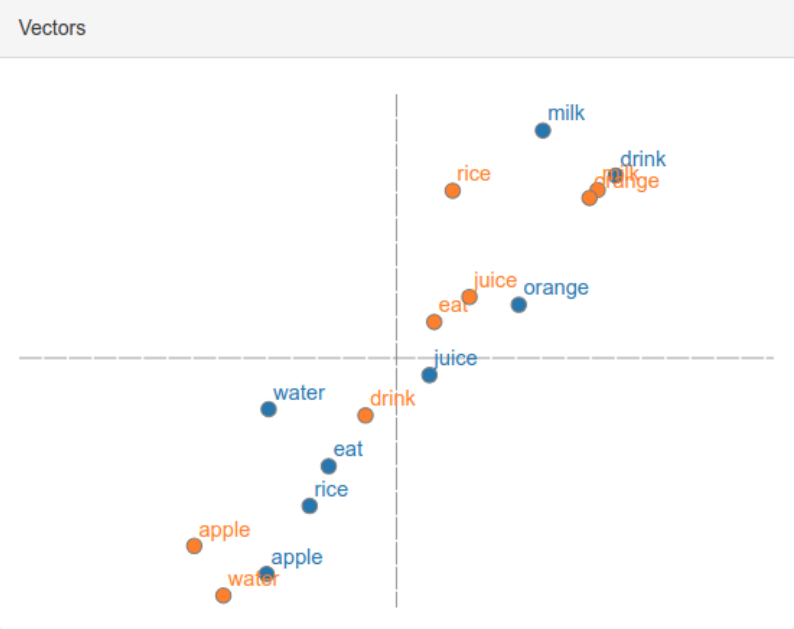
Σχήμα 2.8: Νευρωνικές συνδέσεις

Στην τρίτη εικόνα, εμφανίζονται οι πίνακες εισόδου / εξόδου που περιέχουν τα βάρη για κάθε δεδομένο.



Σχήμα 2.9: Πίνακες Βαρών

Στην τέταρτη εικόνα, αναπαρίστανται οι σχέσεις των δεδομένων στον δυσδιάστατο χώρο.



Σχήμα 2.10: Διανυσματικός χώρος αναπαράστασης των λέξεων

2.1.4 Spacy

Το spacy [4] είναι ένα εργαλείο επεξεργασίας φυσικής γλώσσας, βιομηχανικού επιπέδου, που κάνει χρήση μεθόδων λεξικής ενσωμάτωσης. Πρωτοπορεί σε διεργασίες εξαγωγής πληροφοριών μεγάλης κλίμακας. Είναι γραμμένο εξ ολοκλήρου σε Cython, με κύριο στόχο την αποδοτική διαχείριση της μνήμης RAM. Έρευνες επιβεβαιώνουν ότι το spacy έχει την καλύτερη επίδοση παγκοσμίως.

	ABSOLUTE (MS PER DOC)			RELATIVE (TO SPACY)		
SYSTEM	TOKENIZE	TAG	PARSE	TOKENIZE	TAG	PARSE
spaCy	0.2ms	1ms	19ms	1x	1x	1x
CoreNLP	0.18ms	10ms	49ms	0.9x	10x	2.6x
ZPar	1ms	8ms	850ms	5x	8x	44.7x
NLTK	4ms	443ms	*n/a*	20x	443x	*n/a*

Σχήμα 2.11: Πίνακας επιδόσεων

Το spacy είναι το καλύτερο εργαλείο προετοιμασίας κειμένου για τη χρήση του σε βαθιά μάθηση (Deep Learning). Διασυνδέεται άψογα με διάφορα εργαλεία, όπως το tensorflow, pytorch, scikit-learn, gensim και το υπόλοιπο οικοσύστημα της Python. Με το εργαλείο αυτό, καθίσταται εύκολη η δημιουργία περίπλοκων γλωσσολογικών στατιστικών μοντέλων, που η χρήση τους θα συνέβαλε στην επίλυση διαφόρων προβλημάτων επεξεργασίας φυσικής γλώσσας.

Γνωρίσματα

Το spacy, ως εργαλείο επεξεργασίας φυσικής γλώσσας, επιτυγχάνει μη-καταστρεπτική αναγνώριση λεξικών μονάδων (non-destructive tokenization). Επιπλέον, παρέχει αναγνώριση οντοτήτων και υποστήριξη πολλαπλών γλωσσών. Άλλα γνωρίσματα του spacy είναι, ακόμη, η παροχή προ-εκπαιδευμένων διανυσμάτων λέξεων και η επισήμανση μερών του λόγου (POS tagging). Η συντακτική ανάλυση προτάσεων και ο εύχρηστος κατακερματισμός συμβολοσειρών διακρίνει το spacy για την λειτουργικότητά του. Δεν θα μπορούσε να παραλειφθεί το γεγονός ότι παρέχει τις δυνατότητες εξαγωγής δεδομένων σε numpy μορφής πίνακες, καθώς και της αποδοτικής δυαδικής σειριοποίησης. Τέλος, το μοντέλο του spacy αποδίδει πολύ γρήγορα και με αυστηρά υπολογισμένη ακρίβεια.

Κεφάλαιο 3

Συλλογή Δεδομένων

Στο κεφάλαιο αυτό, αρχικά, γίνεται περιγραφή της διαδικασίας συλλογής των κατάλληλων δεδομένων για την μετέπειτα υλοποίηση. Στη συνέχεια, θα παρουσιαστούν αναλυτικότερα οι τρόποι απόκτησης των δεδομένων μέσω των διαφόρων API, που παρέχουν οι πλατφόρμες κοινωνικής δικτύωσης, καθώς και το πώς επιτυγχάνεται η κατηγοριοποίησή τους. Τέλος, θα παρουσιαστούν τα σύνολα δεδομένων που χρησιμοποιήθηκαν μαζί με τη δομή τους.

3.1 Περιγραφή Διαδικασίας

Το πλήθος των διαθέσιμων δεδομένων στον παγκόσμιο ιστό είναι μεγάλο. Η επιλογή και συλλογή των κατάλληλων δεδομένων εξαρτάται από την έρευνα την οποία θέλουμε να διεξάγουμε. Στην δική μας περίπτωση, το ενδιαφέρον στρέφεται στις πληροφορίες τις οποίες οι χρήστες των κοινωνικών δικτύων επιλέγουν να αναρτήσουν δημόσια. Για την απόκτηση των πληροφοριών αυτών κρίνεται απαραίτητη η χρήση του αντίστοιχου API, το οποίο παρέχεται από το κοινωνικό δίκτυο στο οποίο απευθυνόμαστε. Πριν προχωρήσουμε, όμως, στο βήμα απόκτησης και αποθήκευσης των δεδομένων, πρέπει να καθορίσουμε τον απαιτούμενο όγκο καθαρής πληροφορίας, που θα επιφέρει ως αποτέλεσμα την εξαγωγή ασφαλών στατιστικών συμπερασμάτων. Με τον τρόπο αυτό, η βαρύτητα των συμπερασμάτων αυξάνεται.

Αφού, λοιπόν, έχει καθοριστεί το μέγεθος της απαιτούμενης καθαρής πληροφορίας, για την εξυπηρέτηση στατιστικών σκοπών, εστιάζουμε στην εύρεση θεμάτων γενικού ενδιαφέροντος. Έπειτα, μόλις επιλεχθεί η θεματολογία, κάνοντας χρήση του API, στο οποίο απευθυνόμαστε για τη συλλογή δεδομένων, προσαρμόζουμε τις παρεχόμενες παραμέτρους, με σκοπό τον περιορισμό της εισερχόμενης ροής δεδομένων, στο πλαίσιο της θεματολογίας που έχουμε επιλέξει. Ο χρόνος ανάρτησης και οι γεωγραφικές συντεταγμένες αποτελούν κάποιες από τις παραμέτρους σύνθετης αναζήτησης που μας παρέχουν τα API και ανήκουν στην κατηγορία των μεταδεδομένων. Με την λήξη αυτής της διαδικασίας, τα δεδομένα αποθηκεύονται, πλέον, σε μορφή της επιλογής μας. Η επιλογή που θα κάνουμε επιβάλλεται να λάβει υπόψιν της την μετέπειτα χρήση των δεδομένων, ώστε να είναι η βέλτιστη. Οι μορφές αποθήκευσης δεδομένων "xml" και "json" έχουν καθιερωθεί ως οι πιο διαδεδομένες τα τελευταία χρόνια, με την δεύτερη, μάλιστα, να αποκτά προβάδισμα με την πάροδο του χρόνου, λόγω της μεγαλύτερης

ευχρηστίας της.

Το επόμενο βήμα, μετά την αποθήκευση των δεδομένων, εστιάζει στο φιλτράρισμά τους. Κατα την διαδικασία του φιλτραρίσματος, τα δεδομένα εξετάζονται από ανθρώπους και απορρίπτονται αυτά που αποτελούν θόρυβο (Noise). Ως θόρυβο ορίζουμε τις αναρτήσεις εκείνες, οι οποίες δεν περιέχουν σαφές λεξιλόγιο (π.χ. ακρωνύμια ή λέξεις που δεν περιέχονται σε λεξικά) καθώς και εκείνες που αποτελούν επαναδημοσιεύσεις. Τέλος, οι αναρτήσεις κατηγοριοποιούνται και επισημαίνονται με βάση το συμβάν στο οποίο αντιστοιχούν.

3.2 Χρήση των API

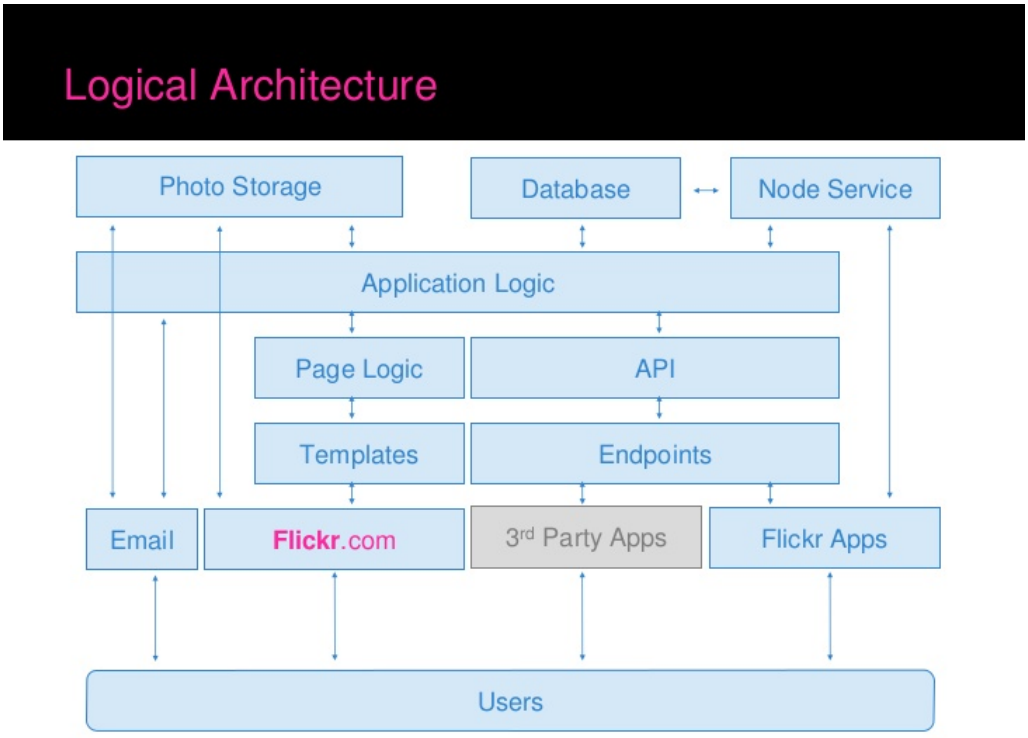
Στην επιστήμη των υπολογιστών, το API (Application Programming Interface) είναι ένα σύνολο ορισμών υπό τη μορφή υπορουτίνας, πρωτοκόλλων επικοινωνίας και εργαλείων ανάπτυξης λογισμικού. Σε γενικές γραμμές, είναι ένα σύνολο σαφώς καθορισμένων μεθόδων επικοινωνίας μεταξύ των διαφόρων τμημάτων της εφαρμογής. Μια καλή υλοποίηση API διευκολύνει την ανάπτυξη προγραμμάτων με το να παρέχει όλα τα απαραίτητα στοιχεία, που χρησιμεύουν στην κατασκευή της εφαρμογής.

Ένα API μπορεί να χρησιμοποιηθεί σε διαδικτυακά συστήματα, σε λειτουργικά συστήματα, σε συστήματα βάσεων δεδομένων, σε υλισμικό υπολογιστών και σε βιβλιοθήκες λογισμικού. Τα ειδικά χαρακτηριστικά ενός API μπορούν να λάβουν πολλές μορφές. Συχνά, περιλαμβάνουν πληροφορίες για τις ρουτίνες, τις δομές δεδομένων, τις κλάσεις αντικειμένων, τις απομακρυσμένες κλήσεις κτλ. Συνήθως, οι οδηγίες χρήσης του API παρέχονται για την διευκόλυνση της χρήσης και υλοποίησής του.

Για την συλλογή των δεδομένων, γίνεται χρήση διαδικτυακών API. Τα διαδικτυακά API είναι οι σαφώς ορισμένες διεπαφές, μέσω των οποίων πραγματοποιούνται αλληλεπιδράσεις μεταξύ διαφόρων συστημάτων. Όταν χρησιμοποιείται στο πλαίσιο της ανάπτυξης διαδικτυακών εφαρμογών, ορίζεται ως ένα σύνολο ειδικών χαρακτηριστικών, όπως τα αιτήματα που ανήκουν στο πρωτόκολλο μεταφοράς υπερσυνδέσμων (HTTP), μαζί με μία ορισμένη δομή απαντήσεων. Οι απαντήσεις αυτές, συνήθως, παρέχονται ως απλά δεδομένα σε “xml” ή “json” μορφή. Ένα παράδειγμα θα μπορούσε να είναι η αλληλεπίδραση μίας ιστοσελίδας ηλεκτρονικού εμπορίου με τα δεδομένα μίας ναυτιλιακής εταιρίας μέσω ενός API.

Στο χώρο των κοινωνικών δικτύων, τα διαδικτυακά API επιτρέπουν την διακίνηση περιεχομένου και δεδομένων μεταξύ εφαρμογών και κοινοτήτων. Με αυτό τον τρόπο, το περιεχόμενο που δημιουργείται σε ένα από τα δύο μέρη μπορεί να υποβληθεί σε δυναμική επεξεργασία μέσω “http” μεθόδων από πολλαπλές διαδικτυακές τοποθεσίες. Για παράδειγμα, το REST API του Twitter επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στα κύρια δεδομένα του Twitter και το Search API τους παρέχει μεθόδους αναζήτησης δεδομένων και αλληλεπίδρασης με αυτά που ανήκουν στις τάσεις [5]. Αντίστοιχα, το Flickr API [6] παρέχει στους προγραμματιστές τη δυνατότητα να αποκτήσουν πρόσβαση στα δεδομένα διακίνησης φωτογραφιών του Flickr. Το Flickr API αποτελείται από ένα σύνολο καλούμενων μεθόδων και μερικά τελικά σημεία “API endpoints”.

Η αρχιτεκτονική ενός τέτοιου API έχει την εξής μορφή:



Σχήμα 3.1: Αρχιτεκτονική του Flickr API

Ενδεικτικά, το Flickr API παρέχει μεθόδους για τις παρακάτω κατηγορίες:

- Δραστηριότητα
- Επαλήθευση στοιχείων χρήστη
- Διαδικτυακά Ιστολόγια
- Φωτογραφικές μηχανές
- Συλλογές
- Επαφές
- Αγαπημένα
- Γκαλερί
- Ομάδες
- Συζητήσεις ομάδων
- Μέλη ομάδων

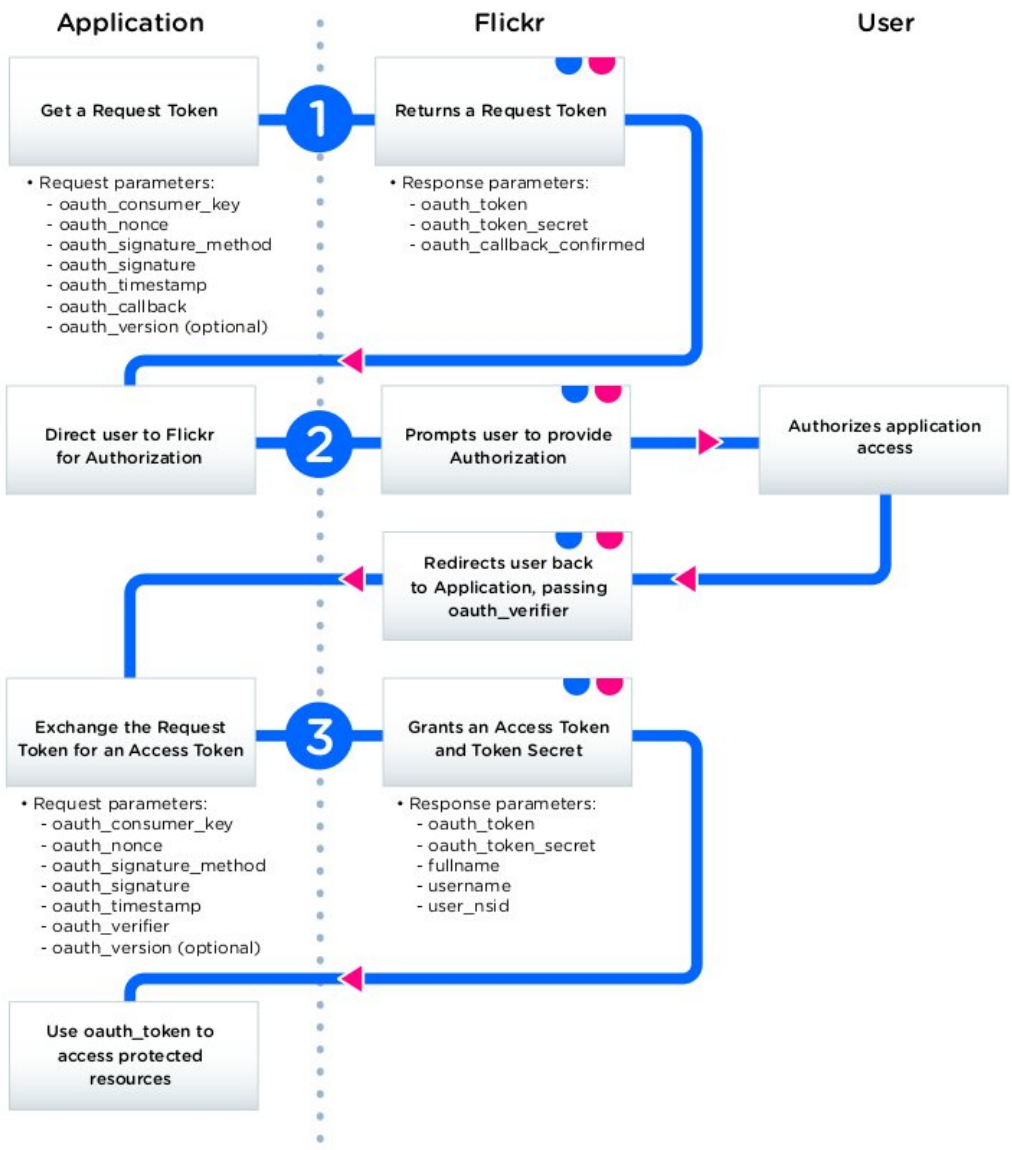
- Ενδιαφέροντα
- Άνθρωποι
- Φωτογραφίες
- Τοποθεσίες
- Προφίλ
- Στατιστικά
- Επισημάνσεις
- Προτιμήσεις

Για παράδειγμα, χρησιμοποιώντας τη μέθοδο flickr.galleries.getPhotos, η οποία δεν απαιτεί επαλήθευση στοιχείων, επέρχεται ως αποτέλεσμα η επιστροφή μιας λίστας φωτογραφιών που ανήκουν σε μία γκαλερί.


```
{
    "photos": {
        "photo": [
            {
                "comment": "best cat picture ever!",
                "_id": "2822546461",
                "_owner": "78398753@N00",
                "_secret": "2dbcdb589f",
                "_server": "1",
                "_farm": "1",
                "_title": "FOO",
                "_ispublic": "1",
                "_isfriend": "0",
                "_isfamily": "0",
                "_is_primary": "1",
                "_has_comment": "1"
            },
            {
                "_id": "2822544806",
                "_owner": "78398753@N00",
                "_secret": "bd93cbe917",
                "_server": "1",
                "_farm": "1",
                "_title": "OOK",
                "_ispublic": "1",
                "_isfriend": "0",
                "_isfamily": "0",
                "_is_primary": "0",
                "_has_comment": "0"
            }
        ],
        "_page": "1",
        "_pages": "1",
        "_perpage": "500",
        "_total": "2"
    }
}
```


Σχήμα 3.2: Λίστα φωτογραφιών σε json μορφή.

Σε περίπτωση που μία μέθοδος απαιτεί επαλήθευση στοιχείων χρήστη, η διαδικασία συνοψίζεται στο παρακάτω διάγραμμα:



Σχήμα 3.3: Μέθοδος επαλήθευσης στοιχείων χρήστη.

Το Flickr API δέχεται τους REST, XML-RPC, SOAP ως αποδεκτούς τρόπους για τη λήψη αιτημάτων και οι τρόποι επιστροφής δεδομένων μπορούν να έχουν τη μορφή: REST, XML-RPC, SOAP, JSON, PHP.

Για παράδειγμα, ένα αίτημα REST μορφής θα είναι:

```
https://api.flickr.com/services/rest/?method=flickr.test.echo&name=value
```

Σχήμα 3.4: Αίτημα σε rest μορφή

Ενώ μία απάντηση σε μορφή json θα είναι:

```
jsonFlickrApi({
        "stat": "ok",
        "blogs": {
                "blog": [
                        {
                                "id"            : "73",
                                "name"          : "Bloxus test",
                                "needspassword" : "0",
                                "url"           : "http://remote.bloxus.com/"
                        },
                        {
                                "id"            : "74",
                                "name"          : "Manila Test",
                                "needspassword" : "1",
                                "url"           : "http://flickrtest1.userland.com/"
                        }
                ]
        }
})
```

Σχήμα 3.5: Απάντηση σε json μορφή

3.3 Δομή των Συνόλων Δεδομένων

Αφού εξηγήσαμε τον τρόπο συλλογής δεδομένων, επόμενο βήμα αποτελεί η τοποθέτησή τους σε σύνολα τα οποία έχουν συγκεκριμένη δομή. Για την εύκολη επεξεργασία των δεδομένων, είναι επιθυμητό η δομή των συνόλων να είναι ίδια για όλα τα σύνολα δεδομένων. Στην παρούσα εργασία, τα σύνολα δεδομένων που χρησιμοποιήσαμε είναι τέσσερα.

Το πρώτο (Zubianga [7]) περιέχει δεδομένα που εξάχθηκαν απ΄ το API του Twitter και αντιστοιχούν σε εννιά διαφορετικά συμβάντα. Το δεύτερο (First Story Detection [8]) περιέχει, επίσης, δεδομένα απ΄ το Twitter, τα οποία αντιστοιχούν σε είκοσι επτά συμβάντα. Η δομή των δύο πρώτων συνόλων δεδομένων περιέχει κάποια κοινά χαρακτηριστικά για κάθε ανάρτηση. Το συμβάν στο οποίο αντιστοιχίζεται η ανάρτηση, το αναγνωριστικό της, το κείμενο και η ημερομηνία είναι τα στοιχεία που επιβάλλεται να υπάρχουν ως απαραίτητες πληροφορίες για τη μετέπειτα επεξεργασία. Ωστόσο, κάποιες από τις αναρτήσεις περιέχουν και επιπλέον πληροφορίες, όπως τη γεωγραφική θέση του χρήστη κατά την διαδικασία μεταφόρτωσης της ανάρτησης.

Τα υπόλοιπα δύο (Social Event Detection 2013, 2014 [9, 10]) περιέχουν δεδομένα που εξάχθηκαν απ΄ το API του Flickr. Η ειδοποιός διαφορά αυτών των συνόλων, σε σχέση με τα προηγούμενα, είναι ότι σε αυτά τα δύο η κύρια πληροφορία της ανάρτησης αποτελεί προϊόν

οπτικοακουστικού περιεχομένου, ενώ στα δύο προηγούμενα ήταν το κείμενο. Η ενιαία δομή συνεχίζει να αποτελεί απαραίτητη προϋπόθεση και για αυτά τα σύνολα δεδομένων. Τα στοιχεία που παρέχονται και θα χρησιμοποιηθούν κατά την αναγνώριση συμβάντων είναι το αναγνωριστικό της κάθε φωτογραφίας, το όνομα χρήστη, η ημερομηνία λήψης και μεταφόρτωσης της φωτογραφίας, ο τίτλος, η περιγραφή και οι επισημάνσεις επιλογής του χρήστη.

Κεφάλαιο 4

Προεπεξεργασία Δεδομένων

Στο κεφάλαιο αυτό, παρουσιάζεται ο τρόπος προσπέλασης, αποθήκευσης στη μνήμη RAM και προεπεξεργασίας των δεδομένων. Σκοπός αυτής της διαδικασίας είναι να προετοιμάσει τα δεδομένα για την χρήση τους από αλγορίθμους ανάλυσης συστάδων.

4.1 Pandas

Στην παρούσα εργασία, η οποία υλοποιήθηκε σε Python, κάναμε χρήση της Pandas [11]. Η Pandas είναι μία δημοφιλής βιβλιοθήκη ανοιχτού κώδικα, η οποία παρέχει υψηλή επίδοση, εύχρηστες δομές δεδομένων και εργαλεία ανάλυσης δεδομένων για την προγραμματιστική γλώσσα Python.

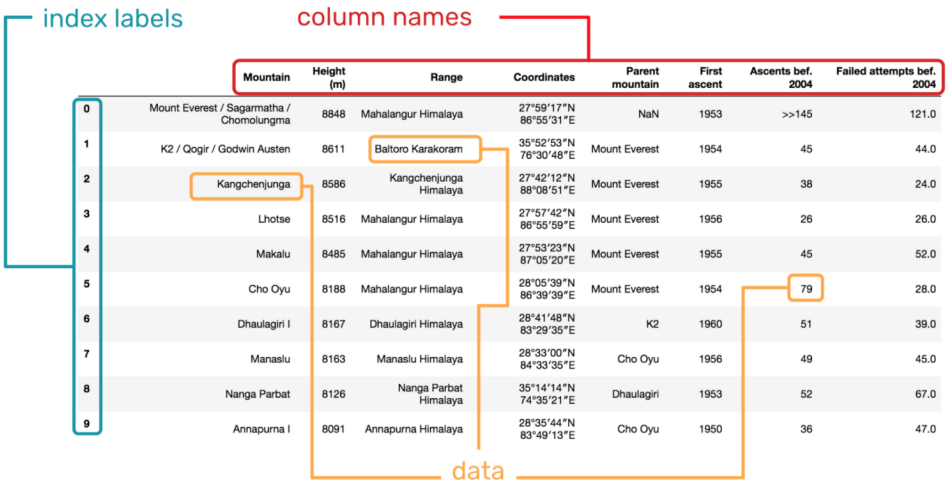
4.1.1 Dataframe

Μία από τις δομές δεδομένων που παρέχει η βιβλιοθήκη Pandas και η οποία χρησιμοποιήθηκε για την αποθήκευση των συνόλων δεδομένων είναι το πλαίσιο δεδομένων (Dataframe). Η δομή αυτή ορίζεται ως μία δυσδιάστατη επισημασμένη δομή δεδομένων, με στήλες που μπορούν να περιέχουν διαφορετικούς τύπους δεδομένων. Γενικά, θα μπορούσαμε να πούμε ότι το πλαίσιο δεδομένων (Pandas Dataframe) αποτελείται από τρία κύρια στοιχεία. Αυτά είναι τα δεδομένα, οι δείκτες και οι στήλες.

Τα σύνολα δεδομένων που θα χρησιμοποιήσουμε περιέχουν τις πληροφορίες που περιγράψαμε στο προηγούμενο κεφάλαιο. Οι πληροφορίες αυτές αποτελούνται από σταθερό πλήθος διαφορετικών τύπων για κάθε ανάρτηση. Στόχος μας είναι η αποθήκευση κάθε ανάρτησης σε μία γραμμή ενός πλαισίου δεδομένων (Dataframe), χρησιμοποιώντας τις επισημασμένες στήλες για την αποθήκευση των δεδομένων που αντιστοιχούν στον τύπο κάθε στήλης.

Τελικά, η δομή μας θα έχει μία μορφή τέτοια, ώστε η κάθε γραμμή να περιέχει όλες τις πληροφορίες που αντιστοιχούν σε μία ανάρτηση και η κάθε στήλη να διαχωρίζει τα είδη των πληροφοριών. Για την προσπέλαση μίας γραμμής, χρησιμοποιούμε έναν δείκτη, ενώ για την προσπέλαση στηλών, χρησιμοποιούμε τις ετικέτες που έχουμε προκαθορίσει.

Παρατίθεται παράδειγμα μίας τέτοια δομής:



	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

Σχήμα 4.1: Dataframe

4.2 Προσπέλαση των Δεδομένων

Κατά την διαδικασία συλλογής των δεδομένων, περιγράψαμε την χρήση των API. Στο σημείο αυτό, είναι σημαντικό να αναφέρουμε ότι τα δεδομένα που επιστρέφονται από την χρήση τέτοιων διεπαφών δεν είναι πάντοτε στην ίδια μορφή. Στην περίπτωσή μας, τα σύνολα δεδομένων που χρησιμοποιήσαμε παρέχονται σε τρεις διαφορετικές μορφές. Για την απλοποίηση του συστήματός μας, δημιουργήσαμε μία βιβλιοθήκη συναρτήσεων, που σκοπό έχουν την μετατροπή κάθε μορφής συνόλου δεδομένων σε μία κοινή. Στόχος αυτής της μετατροπής είναι η δυνατότητα προσπέλασης των δεδομένων, κάνοντας χρήση της συνάρτησης "read_csv" που παρέχει το πλαίσιο δεδομένων (Pandas Dataframe).

Η μορφή συνόλων δεδομένων, στην οποία θα μετατρέψουμε τα αρχικά σύνολα δεδομένων που χρησιμοποιήθηκαν, είναι η μορφή CSV (Comma-Seperated Values). Η CSV είναι μία απλή μορφή αρχείων, που χρησιμοποιείται για την αποθήκευση δεδομένων που βρίσκονται σε μορφή πίνακα, όπως είναι δομημένα τα δεδομένα ενός αρχείου excel ή μιας βάσης δεδομένων. Τα τελικά σύνολα δεδομένων έχουν τη μορφή CSV και διαχωρίζουν τους διαφορετικούς τύπους δεδομένων χρησιμοποιώντας κόμμα (,). Η χρήση της συγκεκριμένης μορφής συνόλων δεδομένων γίνεται ακόμη πιο πρακτική, όταν παραλείπεται το κόμμα από το τέλος κάθε γραμμής.

Τα αρχικά σύνολα δεδομένων παρέχονταν σε τρεις μορφές, όπως αναφέρθηκε πρωτύτερα. Αναλυτικότερα, αυτές οι μορφές είναι οι εξής:

- CSV
- XML
- JSON

4.2.1 Μορφή CSV

Στα σύνολα δεδομένων που είχαν τη μορφή CSV υπήρχε κόμμα (,) στο τέλος κάθε γραμμής. Αυτό δημιουργούσε πρόβλημα κατα τη διαδικασία προσπέλασης και αποθήκευσης των δεδομένων μέσω της συνάρτησης "read_csv", διότι αναμενόταν η παρουσία ενός ακόμη τύπου δεδομένων. Άλλο ένα ζήτημα, που ήταν αναγκαίο να επιλυθεί, ήταν ότι κάποια σύνολα δεδομένων περιείχαν χαρακτήρες που δεν αναγνωρίζονταν απ΄ την κωδικοποίηση "utf-8".

Για την αφαίρεση του κόμματος από το τέλος κάθε γραμμής, δημιουργήσαμε μία συνάρτηση, η οποία διαβάζει γραμμή-γραμμή το αρχείο και αφαρεί τον τελευταίο χαρακτήρα, πριν το ειδικό σύμβολο αλλαγής γραμμής. Όσον αφορά την περίπτωση σφάλματος κωδικοποίησης "utf-8", το πρόβλημα αντιμετωπίζεται με το "άνοιγμα" του αρχείου υπό την κωδικοποίηση "cp1252".

Παράδειγμα συνόλου δεδομένων πριν τη μετατροπή:

```
2011-07-23T19:27:11.000+03:00,
533800,0,0,en,2011-07-23T19:27:37.000+03:00,
23T19:27:45.000+03:00,
,en,2011-07-23T19:27:59.000+03:00,
00,0,31,en,2011-07-23T19:28:29.000+03:00,
2779036600,0,717,en,2011-07-23T19:28:47.000+03:00,
 waste it",                    ,0,0,en,2011-07-23T19:28:49.000+03:00,
T19:29:12.000+03:00,
100,0,0,en,2011-07-23T19:29:14.000+03:00,
,2011-07-23T19:29:31.000+03:00,
7,                          ,0,471,en,2011-07-23T19:29:31.000+03:00,
yc #amywinehouse,94806395089719200,0,2171,en,2011-07-23T19:29:40.000+03:00,
7,94806445396201400,0,471,en,2011-07-23T19:29:52.000+03:00,
T19:30:23.000+03:00,
 #amywinehouse,                          ,0,342,en,2011-07-23T19:30:25.000+03:00,
```

Σχήμα 4.2: FSD πριν τη μετατροπή

Παράδειγμα συνόλου δεδομένων μετά τη μετατροπή:

```
                    ,0,28,en,2011-07-23T19:34:40.000+03:00
ormed at her god-daughter's party earlier this week but has been found dead a
4:55.000+03:00
0,en,2011-07-23T19:35:01.000+03:00
618734500,0,0,en,2011-07-23T19:35:07.000+03:00
23T19:35:07.000+03:00
0,0,en,2011-07-23T19:35:09.000+03:00
Amy winehouse smh,9                    ,0,0,en,2011-07-23T19:35:13.000+03:00
11-07-23T19:35:18.000+03:00
,en,2011-07-23T19:35:19.000+03:00
07-23T19:35:20.000+03:00
,0,0,en,2011-07-23T19:35:23.000+03:00
9:35:26.000+03:00
011-07-23T19:35:32.000+03:00
5:45.000+03:00
11-07-23T19:36:05.000+03:00
T19:36:06.000+03:00
```

Σχήμα 4.3: FSD μετά τη μετατροπή

4.2.2 Μορφή XML

Στα σύνολα δεδομένων που είχαν τη μορφή XML ήταν απαραίτητο να δημιουργηθεί ένα σχεδιάγραμμα (schema) για την επεξήγηση των σχέσεων μεταξύ των κόμβων του XML.

Η αρχική μορφή του συνόλου δεδομένων σε μορφή XML είναι η ακόλουθη:

```
<photo id="[redacted]" photo_url="http://farm1.staticflickr.com/43/[redacted].jpg"
<title>HPIM0041.JPG</title>
<description>Hpim0041.Jpg</description>
<tags>
 <tag>annarbor</tag>
 <tag>arborblogs</tag>
</tags>
<location latitude="42.2753" longitude="-83.7485"></location>
</photo>
<photo id="[redacted]" photo_url="http://farm1.staticflickr.com/43/[redacted]jpg"
<title>HPIM0042.JPG</title>
<description>Hpim0042.Jpg</description>
<tags>
 <tag>annarbor</tag>
 <tag>arborblogs</tag>
</tags>
<location latitude="42.2753" longitude="-83.7485"></location>
</photo>
```

Σχήμα 4.4: Σύνολο Δεδομένων σε XML μορφή

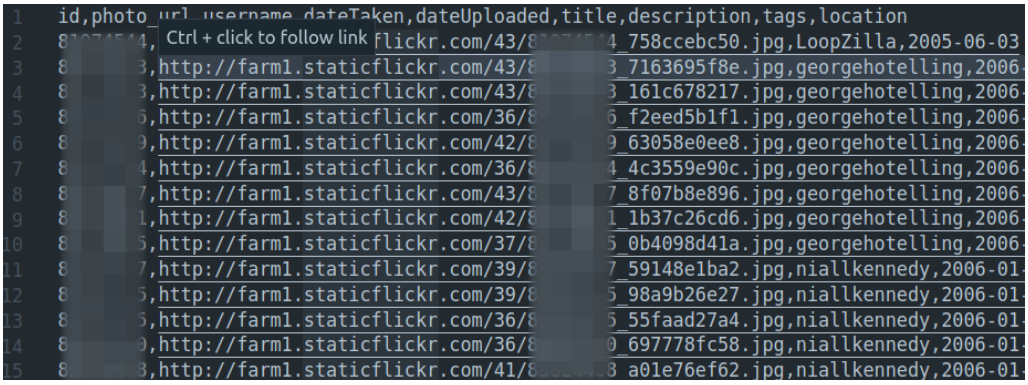
Το αντίστοιχο σχεδιάγραμμα που δημιουργήσαμε για το προηγούμενο σύνολο δεδομένων παρατίθεται παρακάτω σε μορφή JSON:

```
{
    "object": {
        "has_attributes": true,
        "has_nodes": true,
        "attributes": [
            "id",
            "photo_url",
            "username",
            "dateTaken",
            "dateUploaded"
        ],
        "nodes": {
            "title": {
                "has_attributes": false,
                "has_nodes": false
            },
            "description": {
                "has_attributes": false,
                "has_nodes": false
            },
            "tags": {
                "has_attributes": false,
                "has_nodes": true
            },
            "location": {
                "has_attributes": true,
                "has_nodes": false,
                "attributes": [
                    "latitude",
                    "longitude"
                ]
            }
        }
    }
}
```

Σχήμα 4.5: XML Schema

Το εικονιζόμενο σχεδιάγραμμα θα χρησιμοποιηθεί από τη συνάρτηση μετατροπής που δημιουργήσαμε, με σκοπό την προσπέλαση του αρχικού συνόλου δεδομένων και την αλλαγή του σε CSV μορφή.

Η τελική μορφή του συνόλου δεδομένων XML φαίνεται στην ακόλουθη εικόνα:



Σχήμα 4.6: Τελική μορφή σε CSV

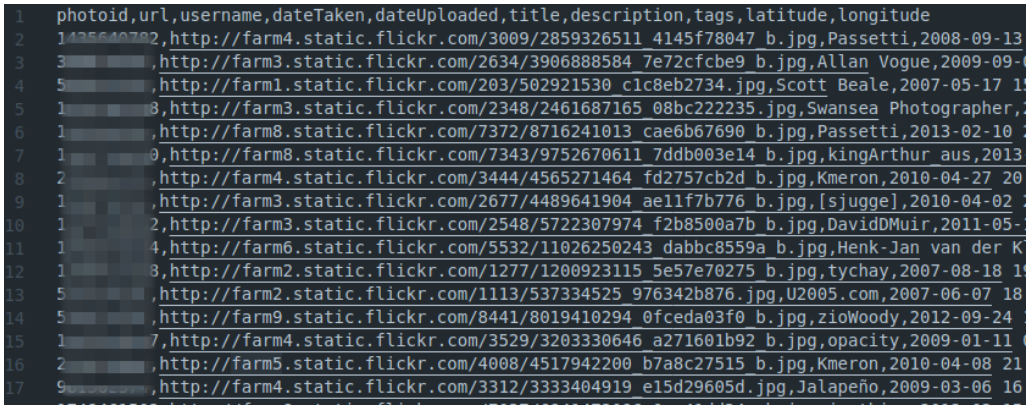
4.2.3 Μορφή JSON

Τα σύνολα δεδομένων, που έφεραν τη μορφή JSON, περιείχαν αναρτήσεις που η καθεμία από αυτές βρισκόταν υπό τη μορφή JSON, η ίδια. Άρα, στην ουσία, το σύνολο δεδομένων σε μορφή JSON ήταν μία συλλογή από JSON. Ο τρόπος μετατροπής του συνόλου δεδομένων JSON σε CSV έγινε σειριακά.

Αρχική μορφή:

Σχήμα 4.7: Σύνολο Δεδομένων σε JSON μορφή

Τελική μορφή:



Σχήμα 4.8: Σύνολο Δεδομένων σε CSV μορφή

Η παραπάνω διαδικασία μετατροπής των τριών μορφών συνόλων δεδομένων σε μορφή CSV έγινε με σκοπό την οικονομία χρόνου ως προς την επαναληπτικότητα των δοκιμών που διεξήχθησαν. Το σύστημα που υλοποιούμε, στην τελική μορφή, μπορεί να αποθηκεύσει σε ένα πλαίσιο δεδομένων (Dataframe) το εκάστοτε σύνολο δεδομένων, να το προεπεξεργάζεται, να το αναλύει και, τέλος, να εξάγει τα αποτελέσματα. Για την βέλτιστη απόδοση του συστήματος, χρειάστηκε να γίνουν πολλές δοκιμές στο στάδιο της ανάλυσης των δεδομένων. Έως το στάδιο της προεπεξεργασίας, διακρίνουμε ότι δεν υπάρχει λόγος επανάληψης των βημάτων αυτών, καθώς οι παρεχόμενες, για προεπεξεργασία και ανάλυση, πληροφορίες παραμένουν αμετάβλητες. Άρα, η αποφυγή επανάληψης της τυποποιημένης διαδικασίας, που ακολουθείται έως το στάδιο της προεπεξεργασίας, εξυπηρετεί το σκοπό εξοικονόμησης χρόνου, κατά τη διάρκεια της έρευνας. Στην τελική μορφή του συστήματος, θα πραγματοποιείται απευθείας η προσπέλαση του συνόλου δεδομένων σε πλαίσιο δεδομένων (Dataframe), με την βοήθεια της συνάρτησης (parseDataset), αποφεύγοντας το ενδιάμεσο στάδιο της αποθήκευσης του συνόλου δεδομένων σε μορφή CSV.

4.3 Στάδιο Προεπεξεργασίας

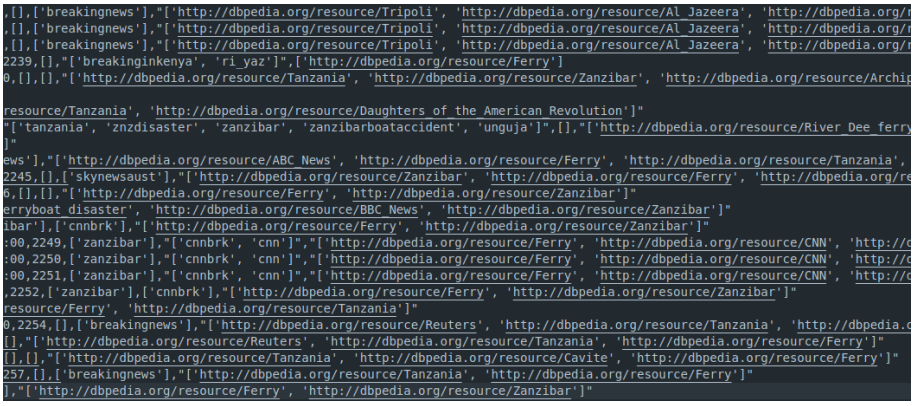
Αρχικά, προσθέτουμε αναγνωριστικό (id) σε κάθε ανάρτηση, δημιουργώντας μία νέα στήλη στο πλαίσιο δεδομένων (Dataframe). Είναι απαραίτητο κάθε αναγνωριστικό να είναι μοναδικό. Στη συνέχεια, ακολουθεί το στάδιο έυρεσης των hashtags. Το hashtag είναι ένας τύπος μεταδεδομένων, που χρησιμοποιείται στα κοινωνικά δίκτυα, όπως το Twitter και άλλα ιστολόγια. Η χρήση του επιτρέπει στους χρήστες την εφαρμογή δυναμικών επισημάνσεων, που καθιστούν δυνατή την εύκολη εύρεση αναρτήσεων συγκεκριμένης θεματολογίας και περιεχομένου. Η δημιουργία των hashtags προκύπτει από την τοποθέτηση του χαρακτήρα της δίεσης (♯) στην αρχή μίας λέξης ή φράσης, της οποίας τα κενά αντικαθίστανται από το χαρακτήρα της κάτω παύλας (_). Λαμβάνουμε υπόψιν μας μόνο το τμήμα της ανάρτησης που περιέχει κείμενο. Κατόπιν, τμηματοποιούμε το κείμενο με βάση τα κενά που υπάρχουν μεταξύ των λέξεων και, έτσι, δημιουργείται μία λίστα από λέξεις. Έπειτα, ελέγχουμε ποια στοιχεία της

λίστας ξεκινούν με το χαρακτήρα της δίεσης (♯) και αυτά που ικανοποιούν αυτή τη συνθήκη τα προσθέτουμε σε μία νέα λίστα. Η λίστα αυτή, με τη σειρά της, καταλαμβάνει μία νέα στήλη στο πλαίσιο δεδομένων (Dataframe). Με τον ίδιο τρόπο, δημιουργείται η λίστα με τις αναφορές σε χρήστες του κοινωνικού δικτύου, η οποία προστίθεται και αυτή ως μία νέα στήλη. Η διαφορά εντοπισμού των αναφορών σε σχέση με τα hashtags έγκειται στο ότι οι αναφορές ξεκινούν με το χαρακτήρα (at).

Αφού ολοκληρωθούν οι παραπάνω διαδικασίες, ξεκινά το στάδιο της διασύνδεσης οντοτήτων. Σε αυτό το στάδιο, κάνουμε χρήση του εργαλείου DBpedia Spotlight. Το εργαλείο αυτό παρέχει τη δυνατότητα απομακρυσμένης και τοπικής χρήσης. Επιλέχθηκε η τοπική χρήση, γιατί παρουσιάζει πλεονέκτημα έναντι της απομακρυσμένης. Η απομακρυσμένη χρήση, παρόλο που δεν απαιτεί κάποια προεργασία, παρουσιάζει περιορισμό εύρους ζώνης. Αυτό σημαίνει ότι υπάρχει ανώτατο όριο στα αιτήματα που μπορεί να δεχτεί ανα συγκεκριμένα χρονικά διαστήματα. Επίσης, ενώ οι χρόνοι αποστολής και λήψης των αιτημάτων σε μεμονωμένες περιπτώσεις είναι μικροί, όταν πρόκειται για αποστολή και λήψη μεγάλου όγκου αιτημάτων, οι χρόνοι προστίθενται και δημιουργούν σημαντική καθυστέρηση. Από την άλλη πλευρά, η τοπική χρήση εξαλείφει τον περιορισμό εύρους ζώνης και οι χρόνοι αποστολής και λήψης των αιτημάτων μειώνονται εκθετικά. Ωστόσο, η τοπική χρήση προϋποθέτει την εγκατάσταση του εργαλείου στο υπολογιστικό μηχάνημα που θα αναπτυχθεί η υλοποίηση, κάτι το οποίο δεν αποτελεί εύκολη διαδικασία. Οι λόγοι αυτοί δικαιολογούν την επιλογή της τοπικής χρήσης.

Αφού εγκαταστήσουμε το εργαλείο στο υπολογιστικό μηχάνημα που κατασκευάζεται το σύστημα, το θέτουμε σε λειτουργία προσομοίωσης τοπικού εξυπηρετητή. Μετά την ενεργοποίηση, στέλνουμε σειριακά τα κείμενα κάθε ανάρτησης, ως αιτήματα για επεξεργασία. Τα αιτήματα που αποστέλλονται περιλαμβάνουν παραμέτρους, δύο από τις οποίες αξίζει να αναφερθούν. Η πρώτη παράμετρος ορίζει τη μορφή που θέλουμε να μας επιστρέψει την απάντηση ο εξυπηρετητής (JSON). Η δεύτερη καθορίζει την πιθανοτική ακρίβεια του αποτελέσματος. Από κάθε απάντηση εξάγουμε τις διασυνδεδεμένες οντότητες και τις αποθηκεύουμε σε μία λίστα. Η λίστα αυτή αποτελεί μία ακόμη στήλη στο πλαίσιο δεδομένων (Dataframe).

Το πλαίσιο δεδομένων (Dataframe) σε αυτό το στάδιο θα έχει την εξής μορφή:



Σχήμα 4.9: Προεπεξεργασμένο Dataframe

Κεφάλαιο 5

Ανάλυση και Ομαδοποίηση των Δεδομένων

Αφού εξηγήθηκε, στο προηγούμενο κεφάλαιο, επαρκώς και όσο εκτενέστερα ήταν δυνατόν η προεπεξεργασία των δεδομένων, είμαστε σε θέση, στο παρόν κεφάλαιο, να ξεκινήσουμε την περιγραφή της ανάλυσης των δεδομένων, έτσι ώστε να γίνει αντιληπτός ο τρόπος πραγματοποίησης της ομαδοποίησης τους.

5.1 Ανάλυση Συστάδων - Ομαδοποίηση

Η ανάλυση συστάδων είναι η διαδικασία ομαδοποίησης ενός συνόλου δεδομένων, με τέτοιο τρόπο, ώστε τα στοιχεία της ίδιας ομάδας να εμφανίζουν περισσότερα κοινά χαρακτηριστικά μεταξύ τους, παρά με αυτά άλλων ομάδων. Η ανάλυση συστάδων δεν είναι ένας συγκεκριμένος αλγόριθμος, αλλά αποτελεί μία κατηγορία προβλημάτων προς επίλυση. Μπορεί να επιτευχθεί μέσω ποικίλων αλγορίθμων, οι οποίοι διαφέρουν σημαντικά μεταξύ τους, όσον αφορά την κατανόηση των συστατικών μιας συστάδας και τον τρόπο ομαδοποίησης.

Προκειμένου να οριστεί σαφώς ο ρόλος της συστάδας στη διαδικασία της ομαδοποίησης, παραθέτουμε κάποιες εννοιολογικές προσεγγίσεις του όρου. Αρχικά, μία πρώτη απόδοση της έννοιας είναι η ομάδα που περιλαμβάνει στοιχεία με μικρή "απόσταση" μεταξύ τους. Δεύτερη προσέγγιση της συστάδας είναι η πυκνή περιοχή του χώρου των δεδομένων και, τέλος, η τρίτη προσέγγιση αποδίδει τον όρο ως συγκεκριμένες στατιστικές κατανομές. Επομένως, η ανάλυση συστάδων μπορεί να διατυπωθεί ως ένα πρόβλημα βελτιστοποίησης πολλαπλών κριτηρίων. Η επιλογή του κατάλληλου αλγορίθμου ομαδοποίησης και των παραμέτρων (όπως η δημιουργία μίας συνάρτησης υπολογισμού απόστασης, η χρήση κατωφλίου πυκνότητας και το αναμενόμενο πλήθος συστάδων) εξαρτάται απ΄ το εκάστοτε σύνολο δεδομένων και το σκοπό χρήσης των εξαχθέντων αποτελεσμάτων.

Με βάση τα προαναφερθέντα, η ανάλυση συστάδων δεν αποτελεί μία αυτοματοποιημένη διαδικασία, αλλά μία επαναλαμβανόμενη διαδικασία γνωστικής ανακάλυψης ή μία διαδραστική διαδικασία βελτιστοποίησης πολλαπλών κριτηρίων, κατά την οποία γίνονται δοκιμές και προκύπτουν σφάλματα. Είναι, συχνά, απαραίτητη η συνεχής τροποποίηση της διαδικασίας

προεπεξεργασίας των δεδομένων και των παραμέτρων του μοντέλου, μέχρι την εξαγωγή των επιθυμητών αποτελεσμάτων (Unsupervised Learning).

5.1.1 Τύποι Ανάλυσης Συστάδων

Η ανάλυση συστάδων μπορεί να διαιρεθεί σε δύο κατηγορίες. Απ΄ τη μία πλευρά, υπάρχει η αυστηρή ανάλυση συστάδων (Hard Clustering), κατά την οποία κάθε στοιχείο μπορεί να ανήκει αποκλειστικά σε μία συστάδα. Απ΄ την άλλη πλευρά, υπάρχει η πιθανοτική ανάλυση συστάδων (Soft Clustering), στην οποία κάθε στοιχείο εμφανίζει ξεχωριστή πιθανότητα να ανήκει σε κάθε συστάδα.

5.1.2 Τύποι Αλγορίθμων Ανάλυσης Συστάδων

Η διαδικασία ανάλυσης συστάδων είναι υποκειμενική. Αυτό σημαίνει ότι τα μέσα που μπορούν να χρησιμοποιηθούν για την επίτευξη αυτού του σκοπού είναι πολλά. Κάθε μεθοδολογία βασίζεται σε ένα διαφορετικό σύνολο κανόνων, για να οριστεί ο τρόπος που διακρίνεται η ομοιότητα μεταξύ των δεδομένων. Στην πραγματικότητα, υπάρχουν περισσότεροι από εκατό γνωστοί αλγόριθμοι ανάλυσης συστάδων, αλλά λίγοι είναι αυτοί που χρησιμοποιούνται ευρέως. Παρακάτω, θα εξεταστούν αναλυτικότερα οι τύποι των αλγορίθμων.

- **Μοντέλα Συνδεσιμότητας:**

  Όπως φαίνεται και απ΄ την ονομασία τους, τα μοντέλα αυτά βασίζονται στη λογική ότι η μικρότερη απόσταση μεταξύ των αναπαραστάσεων των δεδομένων στο χώρο συνεπάγεται μεγαλύτερη ομοιότητα των δεδομένων αυτών, συγκριτικά με αυτά που απέχουν περισσότερο μεταξύ τους. Αυτά τα μοντέλα ακολουθούν δύο προσεγγίσεις. Στην πρώτη προσέγγιση, ξεκινούν με την κατηγοριοποίηση όλων των στοιχείων σε ξεχωριστές συστάδες και συνεχίζουν με τη συσσωμάτωσή τους, όσο η απόσταση ελαττώνεται. Στη δεύτερη προσέγγιση, όλα τα στοιχεία κατηγοριοποιούνται σε μία συστάδα και ακολούθως τμηματοποιούνται, όσο η απόσταση αυξάνεται. Παράλληλα, η επιλογή της συνάρτησης απόστασης έγκειται στην προαίρεση του κάθε ερευνητή. Στο συγκεκριμένο είδος μοντέλων, είναι εύκολο να γίνει αντιληπτή η υποβόσκουσα λογική τους, αλλά υπολείπεται στη διαχείριση των δεδομένων, όταν παρουσιάζεται κλιμάκωση. Ένας γνωστός αλγόριθμος ανάλυσης συστάδων, που ανήκει σε αυτή την κατηγορία, είναι ο Hierarchical Clustering Algorithm.

- **Μοντέλα Γεωμετρικού Κέντρου:**

  Αυτά τα μοντέλα είναι αλγόριθμοι επαναλαμβανόμενης ανάλυσης συστάδων, οι οποίοι βασίζονται στη λογική ότι η απόσταση των δεδομένων απ΄ τα γεωμετρικά κέντρα των συστάδων υποδηλώνουν την υπάρχουσα ομοιότητα μεταξύ των δεδομένων και των συστάδων. Ο K-Means είναι ο πιο δημοφιλής αλγόριθμος, που υπάγεται σε αυτή την κατηγορία μοντέλων. Το κύριο χαρακτηριστικό των μοντέλων αυτών είναι η απαραίτητη

γνώση του πλήθους των τελικών συστάδων, πριν το στάδιο της ανάλυσης των δεδομένων. Η εκτέλεση των συγκεκριμένων μοντέλων γίνεται επαναληπτικά, με σκοπό την εύρεση του τοπικού βέλτιστου.

- **Μοντέλα Κατανομής:**

  Αυτά τα μοντέλα ομαδοποίησης βασίζονται στην πιθανότητα να ανήκουν όλα τα στοιχεία μίας συστάδας στην ίδια στατιστική κατανομή (Gaussian). Στην περίπτωση τέτοιων μοντέλων, συχνά παρουσιάζεται το πρόβλημα της άνισης κατανομής των δεδομένων σε συστάδες.

- **Μοντέλα Πυκνότητας:**

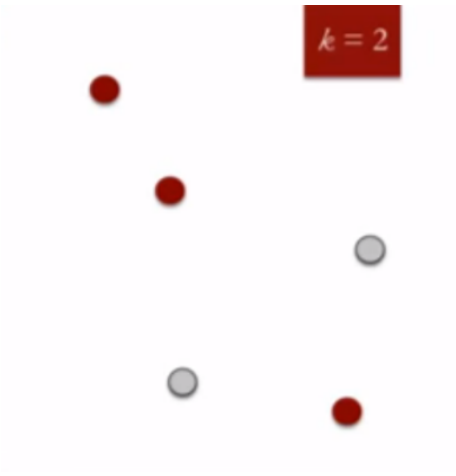
  Τέτοιου είδους μοντέλα κάνουν αναζήτηση, στο χώρο δεδομένων, για περιοχές διαφορετικής πυκνότητας στοιχείων. Έπειτα, απομονώνουν τις περιοχές που έχουν διαφορετική πυκνότητα και τοποθετούν τα στοιχεία που ανήκουν στις περιοχές αυτές σε αντίστοιχες συστάδες. Σημαντικά παραδείγματα τέτοιων μοντέλων αποτελούν οι αλγόριθμοι DBSCAN και OPTICS.

5.1.3 Αλγόριθμος K-Means Clustering

Ο K-Means [12] είναι, ίσως, ο πιο γνωστός αλγόριθμος ανάλυσης συστάδων. Είναι εύκολος στην κατανόηση και στην υλοποίησή του. Ακόμη, χαρακτηρίζεται ως ένας επαναλαμβανόμενος αλγόριθμος ανάλυσης συστάδων, που αποσκοπεί στην εύρεση τοπικού μεγίστου σε κάθε επανάληψη. Η λειτουργία αυτού του αλγορίθμου συνοψίζεται στα παρακάτω έξι βήματα:

1. **Βήμα πρώτο:**

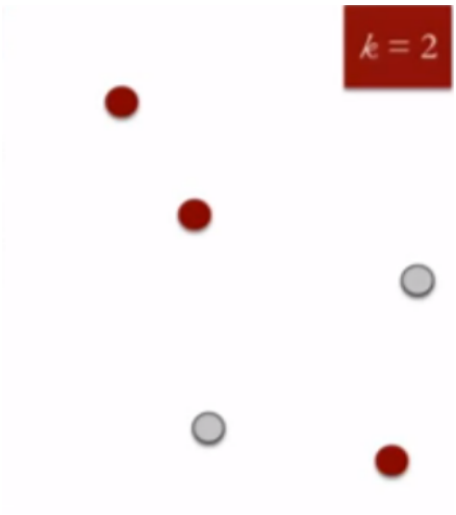
   Καθορίζουμε το επιθυμητό πλήθος συστάδων K. Έστω κ = 2 για αυτά τα 5 στοιχεία του δυσδιάστατου χώρου.



Σχήμα 5.1: K-Means: Βήμα 1ο

2. **Βήμα δεύτερο:**

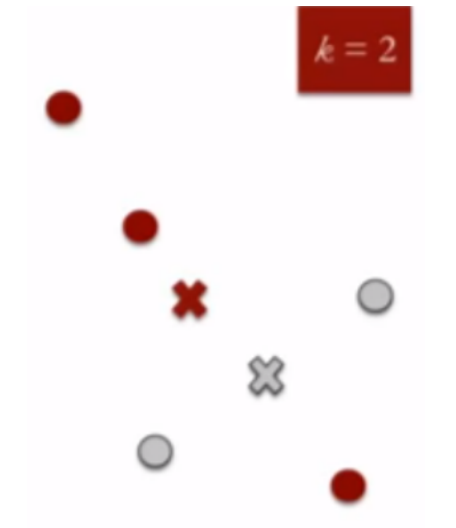
   Τοποθετούμε τυχαία κάθε στοιχείο σε συστάδα. Έστω ότι τοποθετούμε 3 σημεία στη συστάδα 1, χρησιμοποιώντας το κόκκινο χρώμα και τα υπόλοιπα 2 στη συστάδα 2, με γκρι χρώμα.



Σχήμα 5.2: K-Means: Βήμα 2ο

3. **Βήμα τρίτο:**

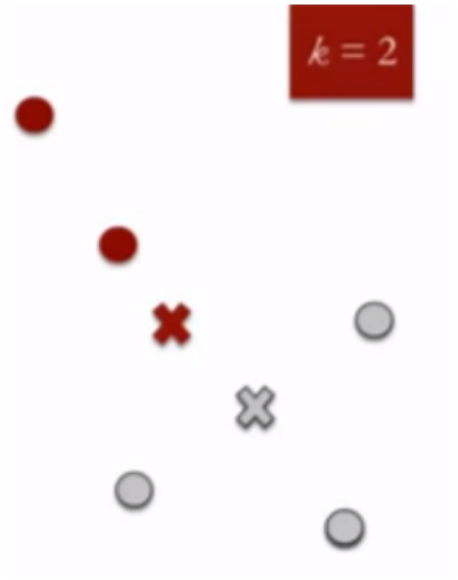
   Υπολογίζουμε τα γεωμετρικά κέντρα κάθε συστάδας. Το γεωμετρικό κέντρο της συστάδας 1 συμβολίζεται με κόκκινο σταυρό, ενώ το γεωμετρικό κέντρο της συστάδας 2 συμβολίζεται με γκρι σταυρό.



Σχήμα 5.3: K-Means: Βήμα 3ο

4. **Βήμα τέταρτο:**

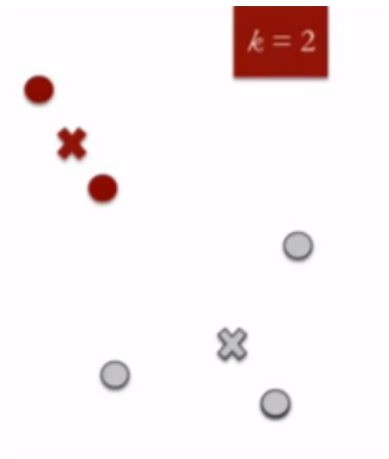
   Επανατοποθετούμε κάθε στοιχείο στη συστάδα που αντιστοιχεί στο κοντινότερο γεωμετρικό κέντρο. Αξίζει να αναφερθεί ότι μόνο το στοιχείο που βρίσκεται στο κάτω μέρος της εικόνας είναι τοποθετημένο στη συστάδα 1, παρόλο που το κοντινότερο γεωμετρικό κέντρο από αυτό, αντιστοιχεί στη συστάδα 2. Για το λόγο αυτό, το επανατοποθετούμε στη συστάδα 2.



Σχήμα 5.4: K-Means: Βήμα 4ο

5. **Βήμα πέμπτο:**

   Επανυπολογίζουμε τα γεωμετρικά κέντρα των συστάδων. Στο παράδειγμά μας, τα νέα κέντρα φαίνονται στην κάτωθι εικόνα.



Σχήμα 5.5: K-Means: Βήμα 5ο

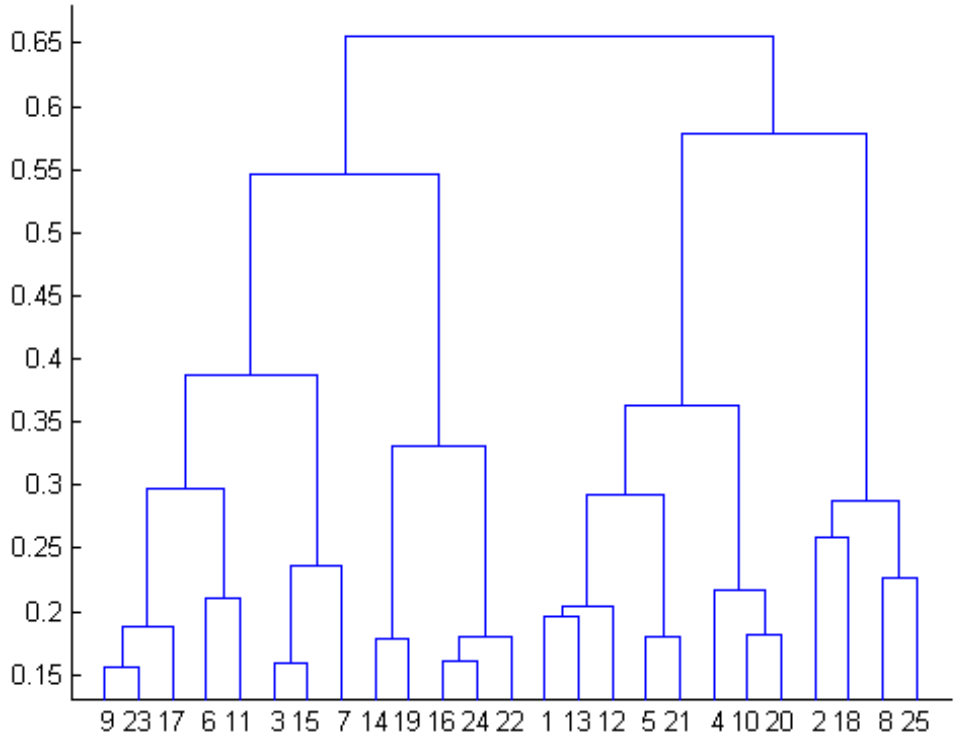
6. **Βήμα έκτο:**

   Επαναλαμβάνουμε τα βήματα 4 και 5, μέχρι να μη χρειάζονται βελτιώσεις. Όταν σταματήσουν να γίνονται ανακατατάξεις στοιχείων ανάμεσα στις 2 συστάδες, για δύο διαδοχικές επαναλήψεις, σημαίνει ότι έχουμε βρει τη βέλτιστη κατανομή και ο αλγόριθμος τερματίζει.

5.1.4 Αλγόριθμος Hierarchical Clustering

Η ιεραρχική ανάλυση συστάδων (Hierarchical Clustering [13]), όπως υποδηλώνει η ονομασία της, είναι ένας αλγόριθμος που βασίζεται στην ιεραρχική δημιουργία συστάδων. Ο αλγόριθμος αυτός ξεκινά με την ανάθεση κάθε στοιχείου σε ξεχωριστή συστάδα. Στη συνέχεια, οι δύο πλησιέστερες συστάδες συγχωνεύονται σε μία συστάδα και το βήμα αυτό επαναλαμβάνεται. Ο τερματισμός του αλγορίθμου επέρχεται, όταν απομείνει μόνο μία συστάδα.

Τα αποτελέσματα της ιεραρχικής ανάλυσης συστάδων μπορούν να αναπαρασταθούν απο δενδρικά διαγράμματα.

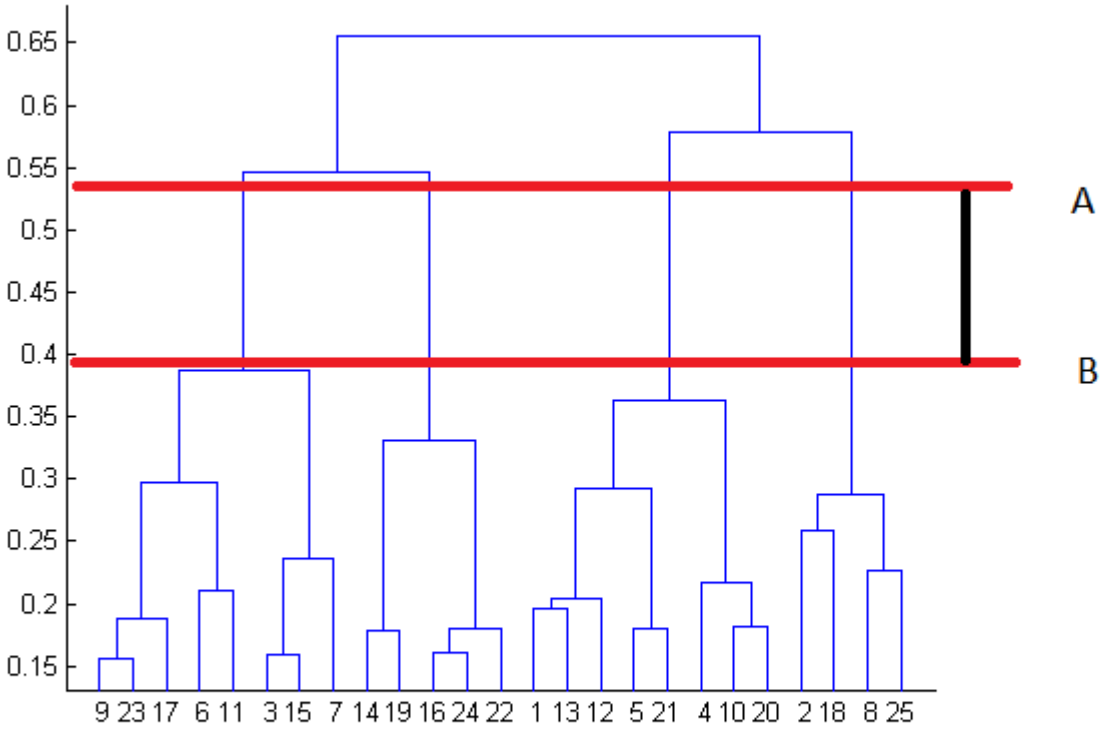


Σχήμα 5.6: Δενδρική Αναπαράσταση Hierarchical Clustering

Επί του οριζόντιου άξονα του διαγράμματος βρίσκονται 25 στοιχεία, τα οποία είναι τοποθετημένα σε αντίστοιχου πλήθους ξεχωριστές συστάδες. Έπειτα, οι κάθε δύο πλησιέστερες συστάδες συγχωνεύονται επαναληπτικά, μέχρι να ενοποιηθούν όλες σε μία τελική, η οποία βρίσκεται στην κορυφή του διαγράμματος. Το ύψος του δενδρικού διαγράμματος, στο οποίο δύο συστάδες συγχωνεύονται, αντιπροσωπεύει την απόσταση των δύο συστάδων στο χώρο των δεδομένων.

Η απόφαση επιλογής του πλήθους των συστάδων, για την καλύτερη απεικόνιση των διαφορετικών ομάδων, βασίζεται στην παρατήρηση του δενδρικού διαγράμματος. Η βέλτιστη επιλογή του πλήθους των συστάδων ανέρχεται στο πλήθος των κατακόρυφων γραμμών του δενδρικού διαγράμματος, που τέμνουν μία οριζόντια γραμμή, η οποία διανύει τη μέγιστη δυνατή κάθετη απόσταση, χωρίς να παρεμβάλλεται ενδιάμεσα συστάδα.

Στο παραπάνω παράδειγμα, η βέλτιστη επιλογή επιτυγχάνεται για πλήθος τεσσάρων συστάδων. Στο παρακάτω σχήμα, αιτιολογείται γραφικά αυτή η επιλογή.



Σχήμα 5.7: Γραφική Αναπαράσταση βέλτιστου πλήθους συστάδων - Hierarchical Clustering

Υπάρχουν δύο γνωρίσματα της ιεραρχικής ανάλυσης συστάδων που αξίζει να αναφερθούν. Το πρώτο γνώρισμα είναι ότι, στο παραπάνω παράδειγμα, ο αλγόριθμος υλοποιήθηκε κάνοντας χρήση της από κάτω προς τα πάνω προσέγγισης. Η από πάνω προς τα κάτω προσέγγιση είναι, επίσης, εφικτή, αν ξεκινήσουμε έχοντας όλα τα στοιχεία σε μία συστάδα και τμηματοποιήσουμε αναδρομικά, μέχρι κάθε στοιχείο να ανήκει σε ξεχωριστή συστάδα. Ως δεύτερο γνώρισμα, διακρίνεται η πληθώρα επιλογών για τη μέτρηση απόστασης δύο συστάδων. Βάσει αυτής της επιλογής αποφασίζεται ποιες συστάδες θα συγχωνευτούν. Στη συνέχεια, παρατίθενται κάποιες απ΄ τις επιλογές μέτρησης της απόστασης μεταξύ των συστάδων.

- Euclidean distance: $||a-b||_2 = \sqrt{(\Sigma(a_i-b_i))}$
- Squared Euclidean distance: $||a-b||_2^2 = \Sigma((a_i-b_i)^2)$
- Manhattan distance: $||a-b||_1 = \Sigma|a_i-b_i|$
- Maximum distance:$||a-b||_{INFINITY} = max_i|a_i-b_i|$
- Mahalanobis distance: $\sqrt{((a-b)^T S^{-1} (-b))}$ {where, s : covariance matrix}

Σχήμα 5.8: Τρόποι μέτρησης απόστασης

5.1.5 Διαφορές μεταξύ των αλγορίθμων K-Means και Hierarchical Clustering

Οι δύο αλγόριθμοι παρουσιάζουν μία σειρά διαφορών μεταξύ τους. Πρώτ΄ απ΄ όλα, ο αλγόριθμος K-Means Clustering είναι πιο αποδοτικός στη διαχείριση μεγάλου όγκου δεδομένων, συγκριτικά με τον Hierarchical Clustering. Αυτό συμβαίνει γιατί η χρονική πολυπλοκότητα του πρώτου είναι γραμμική O(n), ενώ του δεύτερου είναι ανάλογη του τετραγώνου της αύξησης της εισόδου $O(n^2)$. Ακόμη, στον αλγόριθμο K-Means Clustering, επειδή η αρχική ανάθεση των στοιχείων σε συστάδες είναι τυχαία, τα αποτελέσματα, που παράγονται με την πολλαπλή εκτέλεση του αλγορίθμου, παρουσιάζουν διαφορές. Αντιθέτως, στον αλγόριθμο Hierarchical Clustering τα αποτελέσματα είναι αναπαράξιμα. Επιπλέον, ο αλγόριθμος K-Means Clustering φαίνεται να λειτουργεί καλύτερα, όταν το σχήμα που αναπαριστά τις συστάδες στο χώρο είναι υπερ-σφαιρικό, δηλαδή διατηρεί τις ιδιότητες του γεωμετρικού τόπου του κύκλου, αναγωγικά σε όσες διαστάσεις αποτελούν το χώρο δεδομένων. Τέλος, ο αλγόριθμος K-Means Clustering απαιτεί προηγούμενη γνώση του πλήθους των κατηγοριών-συστάδων, στις οποίες θέλουμε να χωρίσουμε τα δεδομένα. Εντούτοις, στον αλγόριθμο Hierarchical Clustering, ερμηνεύοντας το δενδρικό σχεδιάγραμμα, μπορούμε να τον τερματίσουμε σε οποιοδήποτε πλήθος συστάδων θεωρούμε ικανοποιητικό.

5.2 Αναγνώριση Συμβάντων

Στην παραπάνω ενότητα, έγινε περιγραφή των διαφορετικών τύπων ανάλυσης συστάδων και δύο εκ των πιο δημοφιλών αλγορίθμων που χρησιμοποιούνται για αυτό το σκοπό. Ο λόγος περιγραφής τους είναι να καταλάβει ο αναγνώστης πού βασίστηκε η υλοποίηση, που θα παρουσιαστεί στην παρούσα ενότητα.

5.2.1 Προοίμιο

Η υλοποίηση του συστήματος αναγνώρισης συμβάντων εστιάζει τόσο στα δεδομένα που περιέχουν κείμενο, όσο και στα μεταδεδομένα κάθε ανάρτησης. Ως σημείο αναφοράς για τις αποδεκτές μορφές της εισόδου του συστήματος θεωρήθηκε η μορφή των συνόλων δεδομένων που συλλέχθηκαν από το API του Twitter. Ωστόσο, θεωρείται αποδεκτή κάθε μορφή συνόλου δεδομένων που συμβαδίζει με αυτή του Twitter και περιέχει τις ακόλουθες πληροφορίες:

- Το όνομα χρήστη του συντάκτη της ανάρτησης (μεταδεδομένο)
- Την ημερομηνία δημοσίευσης της ανάρτησης (μεταδεδομένο)
- Το κειμενικό περιεχόμενο (μικρής έκτασης)

Η πληροφορία του ονόματος του συντάκτη είναι σημαντική, διότι αναμένει κανείς ένας συντάκτης να δημοσιεύει αναρτήσεις συγκεκριμένης θεματολογίας, δεδομένης της μικρής διαθέσιμης έκτασης κειμένου της κάθε ανάρτησης σε ένα συγκεκριμένο χρονικό πλαίσιο. Σε περιπτώσεις που η ιδιωτικότητα απασχολεί τους χρήστες του κοινωνικού δικτύου, δίνεται η δυνατότητα χρήσης ψευδωνύμων, τα οποία δημιουργούνται μέσω hash-συναρτήσεων και αντικαθιστούν τα πραγματικά ονόματα των χρηστών. Αυτό δεν επηρεάζει το αποτέλεσμα της υλοποίησης, καθώς δεν βασίζεται στην ταυτότητα του χρήστη αλλά στο περιεχόμενο της δημοσίευσης. Η παροχή της ημερομηνίας των αναρτήσεων ως μεταδεδομένο είναι επίσης σημαντική, αν αναλογιστούμε ότι υπάρχει μεγάλη πιθανότητα, κοντινές χρονικά αναρτήσεις, να αναφέρονται στο ίδιο γεγονός. Τέλος, απαραίτητη προϋπόθεση είναι η ύπαρξη κειμένου σε κάθε ανάρτηση, γιατί αναρτήσεις που φέρουν κοινό σημασιολογικά περιεχόμενο, συνήθως αντιστοιχούν σε ίδιο συμβάν.

5.2.2 Εννοιολογικός ορισμός ομοιότητας

Μία κεντρική αρχή στους αλγορίθμους ανάλυσης συστάδων είναι αυτή της ομοιότητας μεταξύ των δεδομένων. Παρόμοια στοιχεία αναμένεται να ομαδοποιηθούν σε κοινή συστάδα, ενώ ανόμοια πρέπει να τοποθετούνται σε διαφορετικές συστάδες.

Στην παρούσα έρευνα, οι παράμετροι που τέθηκαν για τον ορισμό της ομοιότητας είναι οι τρεις που αναφέρθηκαν νωρίτερα (όνομα χρήστη, ημερομηνία, κειμενικό περιεχόμενο) και κάποιες ακόμη. Κατά τη διάρκεια της προεπεξεργασίας των δεδομένων, είχαν εξαχθεί από το κειμενικό περιεχόμενο τρία είδη πληροφορίας, τα οποία μετέπειτα χρησιμοποιούνται ως

παράμετροι για τον ορισμό της ομοιότητας. Τα στοιχεία αυτά είναι τα hashtags, οι επισημάνσεις χρηστών και οι διασυνδεδεμένες οντότητες.

Τα hashtags αποτελούν μείζονος σημασίας χαρακτηριστικά των αναρτήσεων, για την ανίχνευση της μεταξύ τους ομοιότητας. Αυτό συμβαίνει επειδή hashtags που περιέχουν ίδιες λέξεις ή φράσεις, συνήθως έχουν την ίδια θεματολογία. Οι επισημάνσεις χρηστών βασίζονται περίπου στην ίδια λογική με αυτή των hashtags. Κάθε ανάρτηση με συγκεκριμένο περιεχόμενο μπορεί να περιέχει επισημάνσεις ατόμων που έχουν αναφερθεί ή σχετίζονται με την θεματολογία αυτής της ανάρτησης. Έτσι, η ανάρτηση ενός χρήστη συνδέεται με την ανάρτηση κάποιου άλλου και η σύνδεσή τους αποτελεί ένδειξη ομοιότητας. Τέλος, οι διασυνδεδεμένες οντότητες συμβάλλουν και αυτές με τη σειρά τους στον εντοπισμό ομοιότητας μεταξύ αναρτήσεων. Αναρτήσεις που περιέχουν τις ίδιες διασυνδεδεμένες οντότητες αυξάνουν την πιθανότητα να αναφέρονται στο ίδιο γεγονός, επομένως, βρίσκονται πλησίον σημασιολογικά.

5.2.3 Μαθηματικός ορισμός ομοιότητας

Ως συνάρτηση ομοιότητας ορίζουμε την συνάρτηση αυτή, που παίρνει ως ορίσματα τις πληροφορίες που περιέχουν δύο αναρτήσεις και επιστρέφει την συνολική τους ομοιότητα, η οποία εξαρτάται απ΄ τις ομοιότητες των επιμέρους στοιχείων τους.

Η συνάρτηση αυτή εμφανίζεται παρακάτω:
$sim(p_i, p_j) = a_1 \cdot I_{author} + a_2 \cdot f_t(t_i, t_j) + a_3 \cdot f_w(txt_i, txt_j) + a_4 \cdot f_h(h_i, h_j) + a_5 \cdot f_m(m_i, m_j) + a_6 \cdot f_l(l_i, l_j)$

- Ως α_i ορίζουμε τους συντελεστές βαρύτητας, οι οποίοι θα προκύψουν κατά τη διαδικασία εκπαίδευσης του μοντέλου. Καθώς η τελική ομοιότητα πρέπει να έχει ως σύνολο τιμών το διάστημα [0, 1] και η κάθε επιμέρους συνάρτηση επιστρέφει τιμές που ανήκουν στο ίδιο σύνολο, είναι απαραίτητο να ισχύει η παρακάτω συνθήκη:

$$\sum_{i=1}^{6} \alpha_i = 1$$

- Ως I_{author} ορίζουμε τη συνάρτηση που συγκρίνει τους συντάκτες των δύο αναρτήσεων και επιστρέφει 1 αν οι αναρτήσεις ανήκουν στον ίδιο συντάκτη, 0 εναλλακτικά

- Ως $f_t(t_i, t_j)$ ορίζουμε τη συνάρτηση που υπολογίζει την χρονική απόσταση μεταξύ δύο αναρτήσεων. Η συνάρτηση αυτή είναι μία γνησίως φθίνουσα συνάρτηση, η οποία δέχεται ως είσοδο τις ημερομηνίες μεταφόρτωσης δύο αναρτήσεων και επιστρέφει μία τιμή ομοιότητας βασισμένη στην χρονική τους διαφορά. Για το σκοπό αυτό, επιλέγουμε μία εκθετική συνάρτηση της μορφής:

$$f_t(t_i, t_j) = e^{-|t_i - t_j|/q}$$

  η οποία χρησιμοποιεί μία παράμετρο q ως ένα μέγιστο χρονικό παράθυρο. Επιστρέφει 1 όταν οι χρόνοι δημοσίευσης των αναρτήσεων είναι παραπλήσιοι $t_i \approx t_j$ και τείνει προς το 0, καθώς η απόλυτη διαφορά των χρόνων $|t_i - t_j|$ αυξάνεται.

- Ως $f_w(txt_i, txt_j)$ ορίζουμε τη συνάρτηση που δέχεται ως ορίσματα δύο κειμενικά περιεχόμενα και επιστρέφει την σημασιολογική τους ομοιότητα. Αυτό επιτυγχάνεται κάνοντας χρήση των διανυσματικών αναπαραστάσεων του συνόλου των λέξεων κάθε κειμενικού περιεχομένου. Στη συνέχεια, υπολογίζουμε το συνημίτονο της γωνίας μεταξύ των διανυσμάτων, το οποίο μας δίνει την σημασιολογική ομοιότητα. Το σύνολο τιμών αυτής της συνάρτησης αποτελείται από το διάστημα [0,1] καθώς κάνει χρήση των τιμών που παίρνει η συνάρτηση του συνημιτόνου στο πρώτο τεταρτημόριο. Η συνάρτηση έχει την εξής μορφή:

$$f_w(txt_i, txt_j) = w2v(txt_i).similarity(w2v(txt_j))$$

- Ως $f_h(h_i, h_j)$ ορίζουμε τη συνάρτηση, η οποία παίρνοντας ως είσοδο δύο λίστες με hashtags επιστρέφει το ποσοστό των κοινών hashtags. Αξίζει να σημειωθεί ότι λόγω της διαφορετικής προσέγγισης των χρηστών στον τρόπο γραφής των hashtags η ομοιότητα ενός ζεύγους υπόκειται στην υπέρβαση ενός ορίου ποσοστού κοινών γραμμάτων που περιέχουν τα δύο hashtags που συγκρίνονται. Τελικά, η συνάρτηση παίρνει την μορφή:

$$f_h(h_i, h_j) = (2 \cdot no - of - same)/((len(h_i) + len(h_j))$$

Αλγόριθμος 5.1 Ψευδοκώδικας για την συνάρτηση σύγκρισης hashtags

H_i: input list of $post_i$ hashtags
H_j: input list of $post_j$ hashtags
sum: number of common hashtags between H_i, H_j
R: computed ratio

```
1: for i in H_i do:
2:     for j in H_j do:
3:         if computedWordRatio(i, j) ≥ 0.8 then:
4:             sum += 1;
5:             break;
6: R = (2 · sum)/(len(H_i) + len(H_j))
```

- Ως $f_m(m_i, m_j)$ ορίζουμε τη συνάρτηση που δέχεται ως είσοδο δύο λίστες επισημάνσεων χρηστών και επιστρέφει το ποσοστό των κοινών.

$$f_m(m_i, m_j) = (2 \cdot no - of - same)/((len(m_i) + len(m_j))$$

Αλγόριθμος 5.2 Ψευδοκώδικας για την συνάρτηση σύγκρισης επισημάνσεων

M_i: input list of $post_i$ mentions
M_j: input list of $post_j$ mentions
sum: number of common mentions between M_i, M_j
R: computed ratio

```
for i in M_i do:
    if i in M_j then:
        sum += 1;
R = (2 · sum)/(len(M_i) + len(M_j))
```

- Ως $f_l(l_i, l_j)$ ορίζουμε τη συνάρτηση που δέχεται ως είσοδο δύο λίστες διασυνδεδεμένων οντοτήτων και επιστρέφει το ποσοστό των κοινών.

$$f_l(l_i, l_j) = (2 \cdot no - of - same)/((len(l_i) + len(l_j))$$

Αλγόριθμος 5.3 Ψευδοκώδικας για την συνάρτηση σύγκρισης οντοτήτων

L_i: input list of $post_i$ entity links
L_j: input list of $post_j$ entity links
sum: number of common entities between L_i, L_j
R: computed ratio

```
for i in L_i do:
    if i in L_j then:
        sum += 1;
R = (2 · sum)/(len(L_i) + len(L_j))
```

5.2.4 Μεθοδολογία

Σε αυτή την ενότητα, παρουσιάζονται συνοπτικά οι δύο προσεγγίσεις που ακολουθήσαμε για την ανίχνευση των συμβάντων, ανάλογα με τα χαρακτηριστικά του εξεταζόμενου συνόλου δεδομένων.

Η πρώτη προσέγγιση, που ακολουθήθηκε, κάνει χρήση μίας παραμετροποιημένης μορφής του αλγορίθμου Hierarchical Clustering. Αξίζει να επισημανθεί, για την προσέγγιση αυτή, ότι δεδομένης της χρονικής της πολυπλοκότητας, η οποία είναι ανάλογη του τετραγώνου της αύξησης της εισόδου $O(n^2)$, υπάρχει περιορισμός ως προς τον όγκο του συνόλου δεδομένων που επιθυμούμε να υποβληθεί σε ανάλυση. Για τον λόγο αυτό, το μοναδικό σύνολο δεδομένων που υποβλήθηκε σε ανάλυση, με αυτόν τον τρόπο, είναι το FSD. Κύρια διαφορά της παρούσας υλοποίησης του αλγορίθμου που χρησιμοποιήθηκε, συγκριτικά με τον αλγόριθμο Hierarchical

Clustering, είναι ότι η δημιουργία των συστάδων γίνεται δυναμικά, χωρίς την παρέμβαση του ερευνητή.

Η δεύτερη προσέγγιση προέκυψε αναπόφευκτα, καθώς, με τη χρήση συνόλων δεδομένων αυξανόμενου όγκου, η πρώτη προσέγγιση απεδείχθη χρονικά ανεπαρκής. Η εν λόγω προσέγγιση βασίζεται στον αλγόριθμο K-Means Clustering. Η επιλογή του προήλθε απ΄ την ανάγκη ελάττωσης της χρονικής πολυπλοκότητας και βασίζεται στο γεγονός ότι ο αλγόριθμος K-Means Clustering έχει γραμμική πολυπλοκότητα, σε αντίθεση με τον Hierarchical Clustering, που έχει "τετραγωνική". Οι διαφορές που παρουσιάζει η δική μας υλοποίηση σε σχέση με τον αλγόριθμο K-Means Clustering συνοψίζονται στο ότι, αφενός, παραλείπεται το στάδιο της τυχαίας κατανομής των δεδομένων σε συστάδες, και αφετέρου, δεν απαιτείται η προηγούμενη γνώση του πλήθους των συστάδων, καθώς η δημιουργία τους πραγματοποιείται, και πάλι, δυναμικά.

5.2.5 Hierarchical Clustering Προσέγγιση

Η πρώτη προσέγγιση προϋποθέτει την προηγούμενη γνώση των αποστάσεων μεταξύ όλων των πιθανών διαφορετικών ζευγών των δεδομένων. Για τον υπολογισμό των αποστάσεων, κατασκευάσαμε μία συνάρτηση, η οποία μέσω μίας εμφωλευμένης επανάληψης, υπολογίζει σειριακά, για κάθε μία ανάρτηση, την απόστασή της απ΄ όλες τις υπόλοιπες και αποθηκεύει κάθε απόσταση σε έναν δισδιάστατο πίνακα. Ο υπολογισμός κάθε απόστασης κάνει χρήση της συνάρτησης ομοιότητας που περιγράφηκε σε προηγούμενη ενότητα. Επίσης, ο πίνακας που χρησιμοποιήθηκε για την αποθήκευση των αποστάσεων παρέχεται από τη βιβλιοθήκη Numpy. Αφού ολοκληρωθεί το στάδιο εύρεσης και αποθήκευσης των αποστάσεων, ξεκινά το στάδιο της κύριας υλοποίησης. Αρχικά, δημιουργούμε μία κενή λίστα, η οποία, τελικά, θα περιέχει το σύνολο των συστάδων. Στη συνέχεια, επαναλαμβάνουμε την παρακάτω διαδικασία για κάθε ανάρτηση. Βρες την ανάρτηση με την μικρότερη απόσταση. Αν η απόσταση αυτή δεν ξεπερνά ένα μέγιστο όριο απόστασης, το οποίο καθορίζει την θεματολογική ταύτιση δύο αναρτήσεων, τότε ενοποίησε τις συστάδες των αναρτήσεων και πρόσθεσε την τελική συστάδα στην λίστα συστάδων, αλλιώς πρόσθεσε τη μεμονωμένη συστάδα στη λίστα συστάδων.

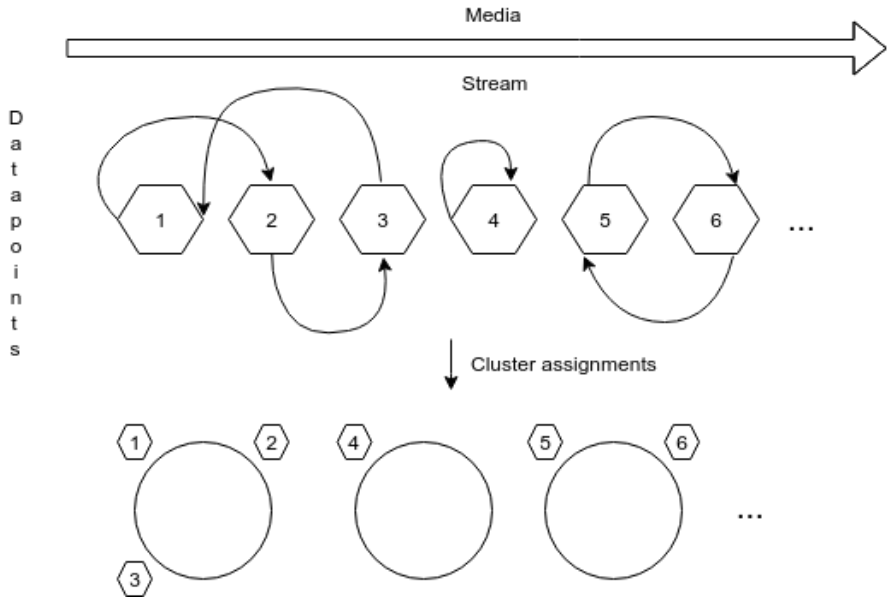
Αλγόριθμος 5.4 Ψευδοκώδικας για την συνάρτηση υπολογισμού αποστάσεων

df: dataframe containing all posts data
D: numpy 2D array containing all computed distances
1: **for** $post_i$ **in** *df* **do**:
2: **for** $post_j$ **in** *df* **do**:
3: **if** $post_i \neq post_j$ **then**:
4: $D[post_i, post_j] = 1 - sim(post_i, post_j)$;
5: **else**:
6: $D[post_i, post_j] = 2$
7: **return** *D*

Αλγόριθμος 5.5 Ψευδοκώδικας για την προσέγγιση Hierarchical Clustering

C: list containing all clusters
D: numpy 2D array containing all computed distances
1: **for** i **in** no_of_posts **do**:
2: $j = find\_closest\_point(i, D)$
3: $temp\_cluster = merge\_clusters(i, j)$
4: $C.append(temp\_cluster)$
5: **return** C

Η εικόνα 5.9 παρουσιάζει γραφικά τον τρόπο ομαδοποίησης, αυτής της προσέγγισης, των δεδομένων σε συστάδες:



Σχήμα 5.9: Hierarchical Clustering προσέγγιση

5.2.6 Βελτιωμένη Hierarchical Clustering Προσέγγιση

Αρχικά, παρατηρούμε ότι ο δισδιάστατος πίνακας που περιέχει τις αποστάσεις μεταξύ όλων των πιθανών διαφορετικών ζευγών των δεδομένων είναι άνω τριγωνικός. Αυτό συμβαίνει, διότι οι αποστάσεις δύο αναρτήσεων παραμένουν σταθερές ανεξάρτητα απ΄ το ποια ανάρτηση θεωρούμε ως αρχή και ποια ως τέλος. Επίσης, η διαγώνιος δεν χρειάζεται να ληφθεί υπόψιν, καθώς η απόσταση μίας ανάρτησης από τον εαυτό της δεν συνιστά χρήσιμη πληροφορία. Με βάση τα προαναφερθέντα, αρκεί να υπολογίσουμε μόνο το άνω τριγωνικό μέρος του πίνακα, μειώνοντας, έτσι, στο μισό, τον απαιτούμενο χρόνο εκέλεσης του συγκεκριμένου σταδίου.

Παράδειγμα 5 στοιχείων

Ο πίνακας αποστάσεων A, που δημιουργείται, είναι μεγέθους 5×5. Ο πρώτος αριθμός κάθε κελιού αντιστοιχεί στο αρχικό σημείο και ο δεύτερος στο τελικό σημείο της απόστασης την οποία μετράμε. Όπως αναφέραμε, κάθε απόσταση $A(i, j)$ θα είναι ίδια με κάθε απόσταση μορφής $A(j, i)$. Επίσης, οι αποστάσεις της μορφής $A(i, j)$ με $i = j$ δεν μας ενδιαφέρουν, γι' αυτό τους αναθέτουμε ένα μεγάλο αριθμό που δεν θα επηρεάσει τη μετέπειτα υλοποίηση.

A(1,1)	A(1,2)	A(1,3)	A(1,4)	A(1,5)
A(2,1)	A(2,2)	A(2,3)	A(2,4)	A(2,5)
A(3,1)	A(3,2)	A(3,3)	A(3,4)	A(3,5)
A(4,1)	A(4,2)	A(4,3)	A(4,4)	A(4,5)
A(5,1)	A(5,2)	A(5,3)	A(5,4)	A(5,5)

Πίνακας 5.1: Πίνακας αποστάσεων 5 στοιχείων

Αλγόριθμος 5.6 Ψευδοκώδικας για την βελτιωμένη συνάρτηση υπολογισμού αποστάσεων

df: dataframe containing all posts data
D: numpy 2D array containing all computed distances
for $post_i$ **in** *df* **do**:
  for $post_j$ **in** $range(index(post_i), df)$ **do**:
    if $post_i \neq post_j$ **then**:
      $D[post_i, post_j] = D[post_j, post_i] = 1 - sim(post_i, post_j)$;
    else:
      $D[post_i, post_j] = 2$
return *D*

Το στάδιο αυτό επιδέχεται περαιτέρω βελτίωσης, αν κάνουμε χρήση πολλαπλών πυρήνων, για την ταυτόχρονη επεξεργασία των δεδομένων. Αυτό προϋποθέτει την ισόποση κατανομή των δεδομένων στους πυρήνες που θα αξιοποιηθούν. Με τον τρόπο αυτό, παρατηρείται επιπλέον μείωση του χρόνου εκτέλεσης, ανάλογη του πλήθους των χρησιμοποιηθέντων πυρήνων. Ακόμη, κατά τη διαδικασία υπολογισμού του δισδιάστατου πίνακα, παρατηρείται η επανάληψη της διανυσματικής αναπαράστασης κειμενικών περιεχομένων που, ήδη, έχουν αναπαρασταθεί σε προηγούμενες επαναλήψεις. Αυτό μπορεί να αποφευχθεί, αν υπολογίσουμε τις διανυσματικές αναπαραστάσεις όλων των κειμενικών περιεχομένων πριν το στάδιο του υπολογισμού των αποστάσεων και τις αποθηκεύσουμε σε ένα νέο πίνακα (memoization). Επομένως, κάθε φορά που υπολογίζεται η σημασιολογική απόσταση δύο κειμενικών περιεχομένων, μπορούμε να κάνουμε χρήση των αποθηκευμένων διανυσματικών αναπαραστάσεων, αποφεύγοντας τον εκ νέου υπολογισμό τους.

Οι βελτιώσεις που περιγράψαμε, μέχρι στιγμής, αφορούν αποκλειστικά την επίδοση του υλοποιημένου συστήματος. Απ' το σημείο αυτό και έπειτα, θα αναλύσουμε τον τρόπο επίτευξης υψηλότερης ακρίβειας στην ομαδοποίηση των δεδομένων. Αν λάβουμε υπόψιν μας τον

περιορισμό στην έκταση του κειμενικού περιεχομένου, που θέτουν τα κοινωνικά δίκτυα που εξετάζουμε, είναι ασφαλές να καταλήξουμε στο συμπέρασμα ότι κάθε ανάρτηση θα αντιστοιχεί σε ένα μοναδικό γεγονός. Επομένως, το βήμα του αλγορίθμου που εκτελεί την εύρεση της ανάρτησης με την μικρότερη απόσταση δεν αποτελεί την πλέον ιδανική λύση. Ας αναλογιστούμε το παράδειγμα που η πρώτη ανάρτηση ζευγαρώνει με τη δεύτερη αμφίδρομα. Στην περίπτωση αυτή, θα επιθυμούσαμε να αντιστοιχίσουμε την δεύτερη ανάρτηση με την αμέσως επόμενη κοντινότερη σε απόσταση ανάρτηση, εάν υπάρχει και δεν ξεπερνά το όριο ομοιότητας που έχουμε καθορίσει. Διαφορετικά, χάνουμε πιθανές ομαδοποιήσεις αναρτήσεων, με αποτέλεσμα την αύξηση της διασποράς της κατανομής των συστάδων. Αυτό αντιμετωπίζεται, αν στο βήμα αυτό αντικαταστήσουμε την διαδικασία εύρεσης της κοντινότερης ανάρτησης με αυτήν της εύρεσης της κοντινότερης ανάρτησης, που δεν ανήκει στην ίδια συστάδα με την υπο εξέταση ανάρτηση. Με τον τρόπο αυτό, επιτυγχάνουμε μεγαλύτερη ακρίβεια αλλά ταυτόχρονα αυξάνουμε και την χρονική πολυπλοκότητα από $O(n^2)$ σε $O(n^3)$. Για την αποφυγή αυτού του αποτελέσματος, μπορούμε να χρησιμοποιήσουμε τη δομή των σωρών. Αυτό θα έχει ως συνέπεια, να διατηρήσουμε την πολυπλοκότητα σε $O(n^2)$ επιτυγχάνοντας, παράλληλα, την μεγαλύτερη απόδοση.

Αλγόριθμος 5.7 Ψευδοκώδικας για την βελτιωμένη προσέγγιση Hierarchical Clustering

```
C: list containing all clusters
D: numpy 2D array containing all computed distances
for i in no_of_posts do:
    heap = heapify_asc(D[i])
    while heap:
        j = heap.pop()
        if not_in_same_cluster(i, j) then:
            temp_cluster = merge_clusters(i, j)
    C.append(temp_cluster)
return C
```

5.2.7 K-Means Clustering Προσέγγιση

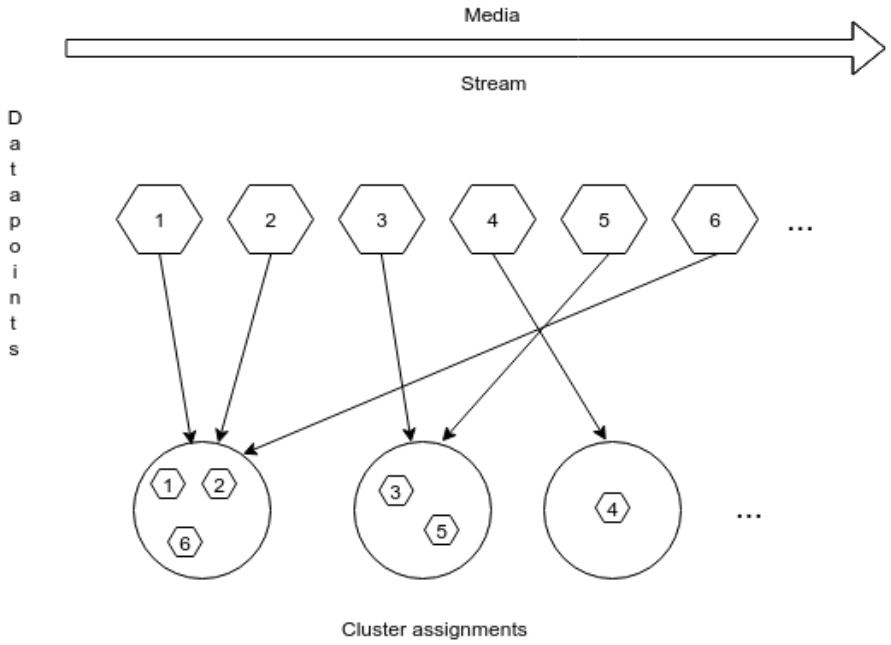
Παρόλες τις βελτιώσεις που πετύχαμε στην πρώτη προσέγγιση, η συνέχιση ύπαρξης του προβλήματος της αδυναμίας διαχείρισης μεγάλου όγκου συνόλων δεδομένων, μάς ώθησε στην εύρεση μιας αποτελεσματικότερης προσέγγισης. Η δεύτερη προσέγγιση δεν απαιτεί την προηγούμενη γνώση των αποστάσεων μεταξύ των δεδομένων, καθώς, όποτε κρίνεται απαραίτητο,οι αποστάσεις αυτές θα υπολογίζονται δυναμικά. Αρχικά, δημιουργείται ξανά μία κενή λίστα, η οποία, τελικά, θα περιέχει το σύνολο των συστάδων. Έπειτα, δημιουργούμε μία συστάδα με μοναδικό περιεχόμενο την πρώτη ανάρτηση, υπολογίζουμε το "κέντρο" της και το προσθέτουμε στη λίστα συστάδων. Κατόπιν, επαναλαμβάνουμε την ακόλουθη διαδικασία για κάθε ανάρτηση πέραν της πρώτης. Για κάθε υπάρχουσα συστάδα, σειριακά, έλεγξε την απόσταση που έχει η ανάρτηση από το κέντρο της εξεταζόμενης συστάδας. Αν η απόσταση δεν ξεπερνάει ένα προκαθορισμένο μέγιστο όριο, το οποίο καθορίζει την θεματολογική ταύ-

τιση της ανάρτησης με το σύνολο των αναρτήσεων που περιέχει η συστάδα, πρόσθεσε την ανάρτηση στη συστάδα και επαναπροσδιόρισε το κέντρο της. Διαφορετικά, συνέχισε, μέχρι να μην υπάρχουν άλλες συστάδες για έλεγχο και δημιούργησε μία νέα συστάδα, στην οποία θα ενταχθεί η υπό εξέταση ανάρτηση. Η προσέγγιση αυτή παρουσιάζει χρονική πολυπλοκότητα $O(n \cdot k)$, όπου k θεωρούμε το πλήθος των συστάδων και n το πλήθος των αναρτήσεων. Αυτό σημαίνει ότι, σε περιπτώσεις συνόλων δεδομένων μικρού πλήθους κατηγοριών, ο αλγόριθμος υπάγεται στην κατηγορία των γραμμικών, εμφανίζοντας σημαντική βελτίωση, συγκριτικά με την πρώτη προσέγγιση.

Αλγόριθμος 5.8 Ψευδοκώδικας για την προσέγγιση K-Means Clustering

C: list containing all clusters
$C.append([post_0])$
for i **in** $range(1, no\_of\_posts)$ **do**:
    for j **in** C **do**:
        $dist = 1 - sim(post_i, j)$
        if $dist < threshold$ **then**:
            $append\_and\_find\_new\_center(j, post_i)$
return C

Η εικόνα 5.10 παρουσιάζει γραφικά τον τρόπο ομαδοποίησης, αυτής της προσέγγισης, των δεδομένων σε συστάδες:



Σχήμα 5.10: K-Means Clustering προσέγγιση

5.2.8 Διαφορές μεταξύ των προσεγγίσεων

Οι κύριες διαφορές των προσεγγίσεων που μόλις περιγράψαμε συνοψίζονται στην, αφενός, δραματική βελτίωση της χρονικής και χωρικής πολυπλοκότητας του συστήματος και στην, αφετέρου, μείωση της ακρίβειας από την πρώτη στην δεύτερη προσέγγιση. Αυτό οφείλεται στην καλύτερη διαχείριση χρόνου και χώρου που παρουσιάζει η δεύτερη, η οποία μας επιτρέπει την επεξεργασία συνόλων δεδομένων αρκετά μεγαλύτερου όγκου, καθώς και την μοναδική προσπέλαση των συστάδων κατά την διαδικασία αντιστοιχίας. Η μικρότερη ακρίβεια αυτού του αλγορίθμου, συγκριτικά με τον πρώτο, μπορεί να αυξηθεί, αν επαναλάβουμε το βήμα προσπέλασης των συστάδων, μεχρι να μην παρατηρούνται ανακατανομές στοιχείων για δύο συνεχόμενες επαναλήψεις. Η τροποποίηση αυτή, με τη σειρά της, επιφέρει ταυτόχρονη αύξηση, τόσο στην χρονική πολυπλοκότητα του αλγορίθμου, όσο και στην ακρίβειά του. Τελικά, εναπόκειται στην ευχέρεια του εκάστοτε ερευνητή να βρει τη χρυσή τομή ακρίβειας - χρονικής πολυπλοκότητας που επιθυμεί.

Κεφάλαιο 6

Παρουσίαση και Εκτίμηση Αποτελεσμάτων

Στο κεφάλαιο αυτό γίνεται η παρουσίαση των αποτελεσμάτων και ο έλεγχος απόδοσης του συστήματος.

6.1 Μεθοδολογία Ελέγχου

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση και την εκτίμηση απόδοσης του συστήματος αναγνώρισης συμβάντων είναι η Python. Για την αξιολόγηση του συστήματός μας, έγινε σύγκριση των στοιχείων κάθε συστάδας που δημιουργήσαμε, με την αντιστοίχιση κάθε στοιχείου σε πραγματικό γεγονός. Οι μετρικές που χρησιμοποιήσαμε παρουσιάζονται αναλυτικότερα παρακάτω [14]:

- Normalized Mutual Information (NMI):

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{(H(\Omega) + H(C))/2}$$

  Όπου $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ είναι το σύνολο συστάδων, $C = \{c_1, c_2, ..., c_k\}$ είναι το σύνολο των κλάσεων, $I(\Omega, C)$ είναι η αμοιβαία πληροφορία μεταξύ των Ω και C και τέλος, $H(\Omega)$ και $H(C)$ είναι οι εντροπίες των Ω και C αντίστοιχα.

- F1-Measure:

$$F1 - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

  Παρατηρούμε ότι η φόρμουλα υπολογισμού της μετρικής F1-Measure χρησιμοποιεί τα Precision και Recall, τα οποία ορίζονται ακολούθως:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Όπου TP (True Positive) θεωρούμε την σωστή ανάθεση δύο όμοιων δεδομένων στην ίδια συστάδα και TN (True Negative) την ανάθεση δύο ανόμοιων δεδομένων σε διαφορετικές συστάδες. Από αυτή τη διαδικασία, μπορεί να προκύψουν δύο είδη σφαλμάτων. Το FP (False Positive), το οποίο προκύπτει απ΄ την ανάθεση δύο ανόμοιων δεδομένων στην ίδια συστάδα και το FN (False Negative), το οποίο με τη σειρά του προκύπτει απ΄ την ανάθεση δύο όμοιων δεδομένων σε διαφορετικές συστάδες.

6.2 Αναλυτική παρουσίαση αποτελεσμάτων

Στην ενότητα αυτή, παρουσιάζονται τα αποτελέσματα του συστήματός μας για τα τέσσερα διαφορετικά σύνολα δεδομένων που χρησιμοποιήσαμε. Για την καλύτερη κατανόηση του ρόλου των επιμέρους παραμέτρων αυτής της διαδικασίας, θα παρουσιαστούν τα αποτελέσματα που εξήχθησαν από διαφορετικές παραμετροποιήσεις της συνάρτησης ομοιότητας.

- Πρώτη περίπτωση συνάρτησης ομοιότητας: Στην περίπτωση αυτή συμπεριλάβαμε μόνο τα μεταδεδομένα.

	F1-Measure	NMI
FSD	20.94	48.86
Zubianga	45.81	60.93
Sed2013	82.23	95.9
Sed2014	88.07	97.21

Πίνακας 6.1: Πίνακας αποτελεσμάτων πρώτης συνάρτησης ομοιότητας

- Δεύτερη περίπτωση συνάρτησης ομοιότητας: Στην περίπτωση αυτή συμπεριλάβαμε τα μεταδεδομένα, μαζί με τις διασυνδεδεμένες οντότητες.

	F1-Measure	NMI
FSD	65.11	64.25
Zubianga	70.26	77.86
Sed2013	84.67	96.49
Sed2014	88.41	97.28

Πίνακας 6.2: Πίνακας αποτελεσμάτων δεύτερης συνάρτησης ομοιότητας

- Τρίτη περίπτωση συνάρτησης ομοιότητας: Στην περίπτωση αυτή συμπεριλάβαμε τα μεταδεδομένα, μαζί με τη διαδικασία λεξικής ενσωμάτωσης στην συνάρτηση ομοιότητας.

	F1-Measure	NMI
FSD	84.7	84.56
Zubianga	89.47	90.23
Sed2013	83.64	96.03
Sed2014	88.18	97.2

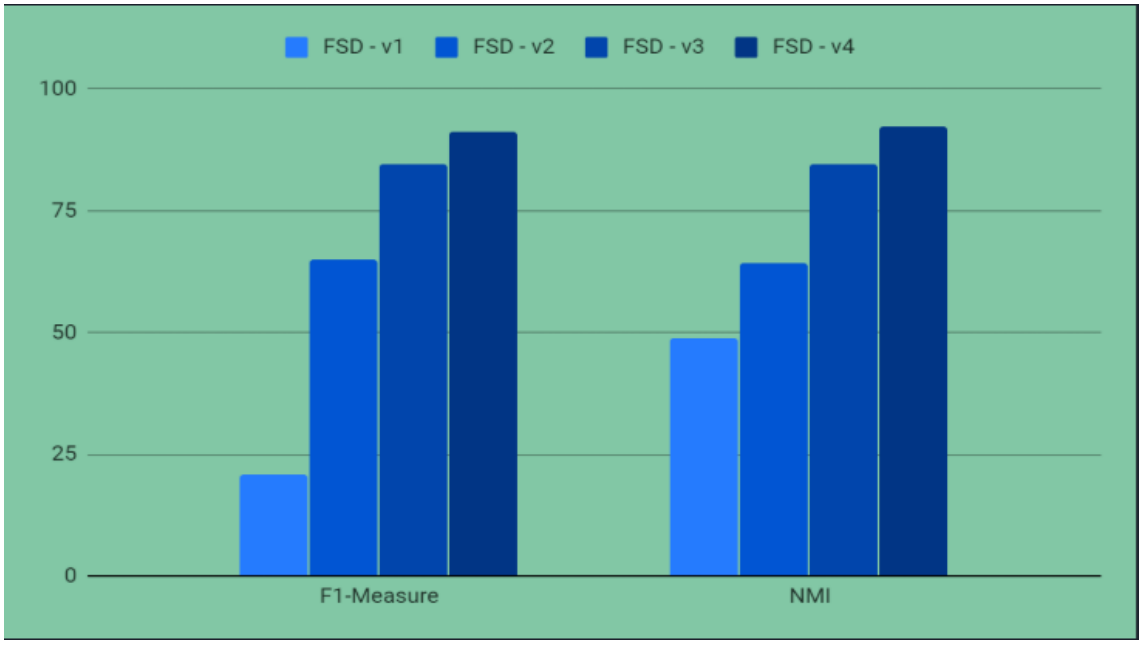
Πίνακας 6.3: Πίνακας αποτελεσμάτων τρίτης συνάρτησης ομοιότητας

- Τέταρτη περίπτωση συνάρτησης ομοιότητας: Στην περίπτωση αυτή συμπεριλάβαμε όλες τις παραπάνω τεχνικές.

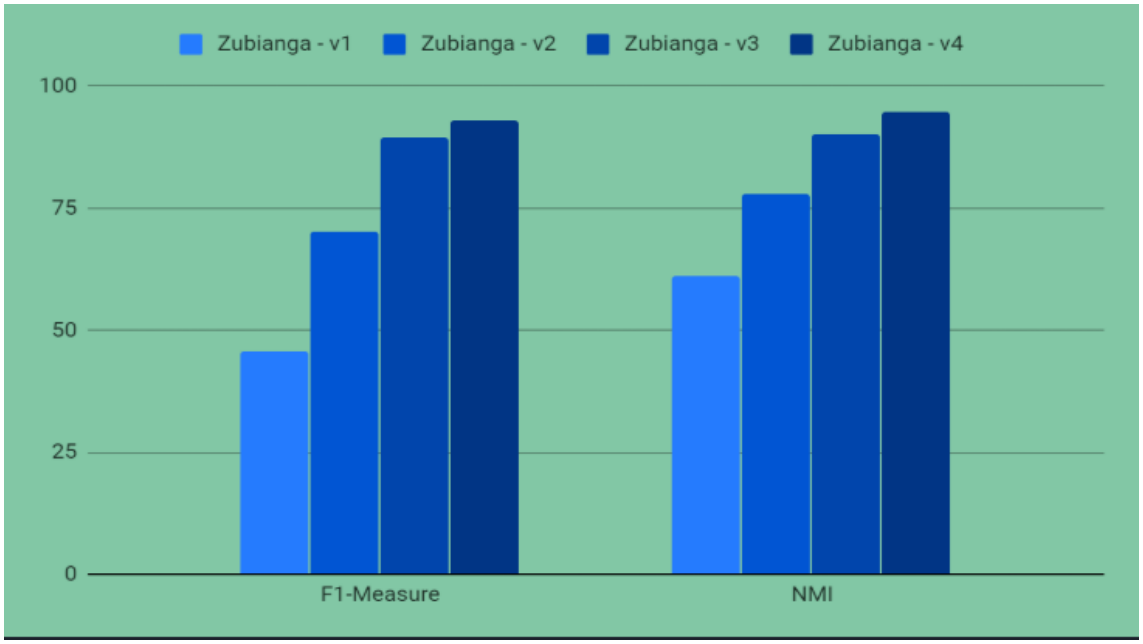
	F1-Measure	NMI
FSD	91.1	92.36
Zubianga	93.12	94.8
Sed2013	89.75	97.35
Sed2014	88.58	97.42

Πίνακας 6.4: Πίνακας αποτελεσμάτων τέταρτης συνάρτησης ομοιότητας

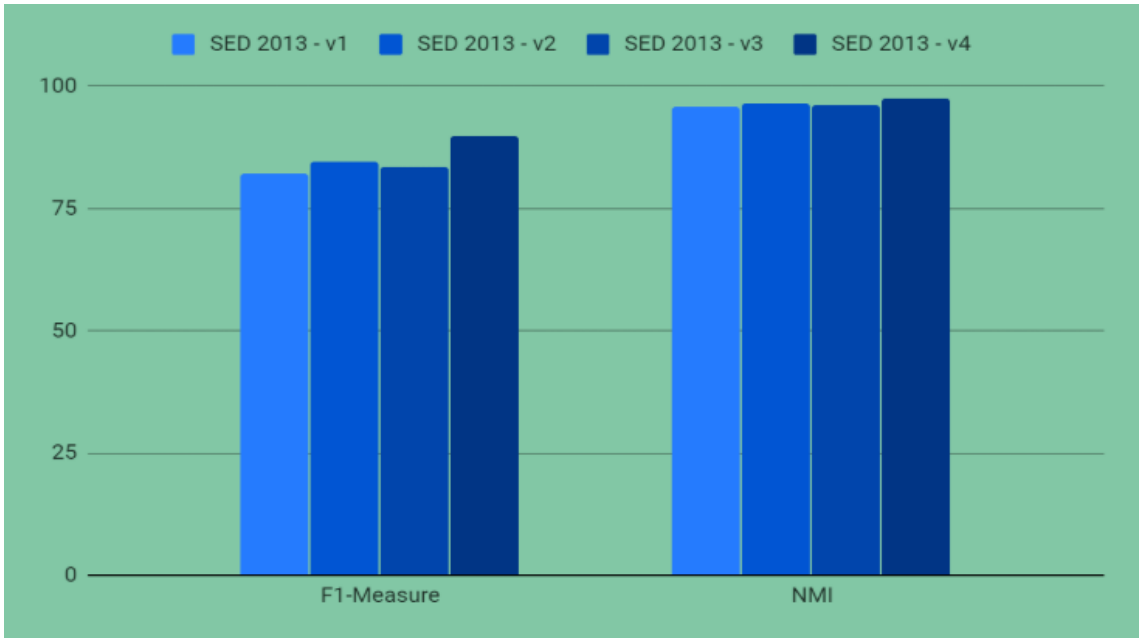
6.3 Βελτιώσεις ανά Σύνολο Δεδομένων



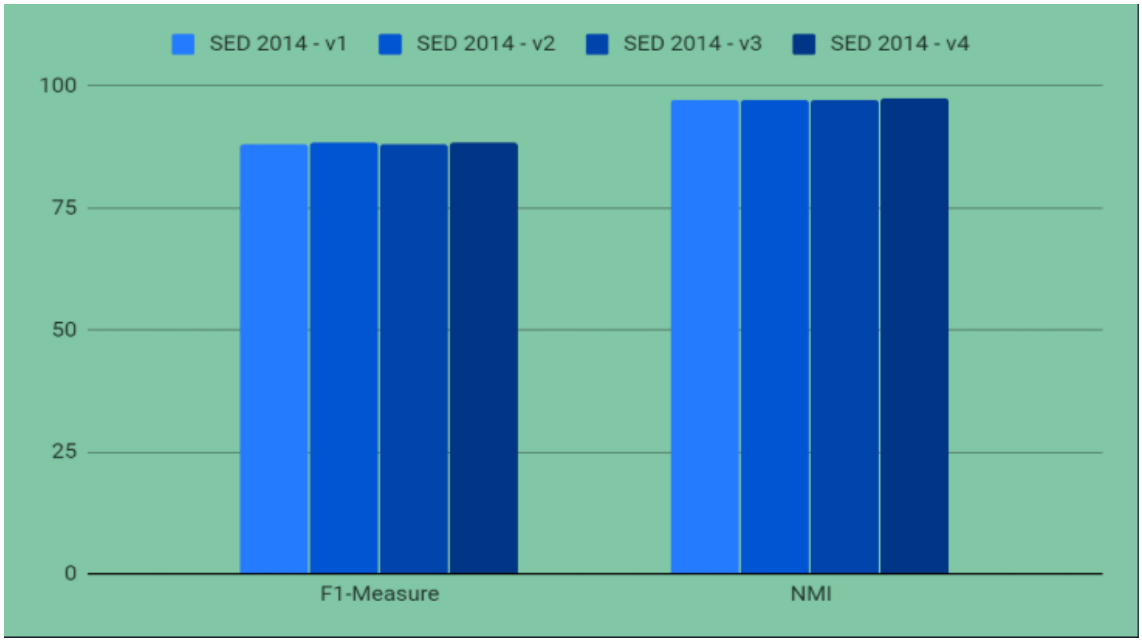
Σχήμα 6.1: FSD Improvements



Σχήμα 6.2: Zubianga Improvements



Σχήμα 6.3: SED2013 Improvements



Σχήμα 6.4: SED2014 Improvements

6.4 Εκτίμηση αποτελεσμάτων

Αρχικά, παρατηρούμε ότι οι διαφορετικές προσεγγίσεις στη συνάρτηση ομοιότητας διαδραματίζουν διαφορετικό ρόλο σε κάθε τύπο συνόλου δεδομένων. Πιο συγκεκριμένα, στα πρώτα δύο σύνολα δεδομένων, που περιλαμβάνουν δεδομένα του Twitter, η πρώτη περίπτωση συνάρτησης ομοιότητας παρουσίασε χαμηλά αποτελέσματα, συγκριτικά με τα σύνολα δεδομένων άλλου τύπου. Αυτό είναι αναμενόμενο, καθώς σε σύνολα δεδομένων των οποίων το κύριο περιεχόμενο είναι οπτικοακουστικής μορφής, τα μεταδεδομένα έχουν μεγαλύτερη σημασία στη σύγκριση της ομοιότητας σε σχέση με το κειμενικό περιεχόμενο, που τις περισσότερες φορές παραλείπεται (περιγραφή). Τέλος, όσον αφορά τις επόμενες τρεις περιπτώσεις συναρτήσεων ομοιότητας, παρατηρούμε ότι η βελτίωση της απόδοσης του συστήματος είναι πολύ σημαντική στα πρώτου τύπου σύνολα δεδομένων. Αυτό συμβαίνει διότι οι τεχνικές διασύνδεσης οντοτήτων και λεξικής ενσωμάτωσης εφαρμόζονται σε κειμενικά περιεχόμενα, που, και πάλι, είναι το κύριο χαρακτηριστικό μόνο των συνόλων δεδομένων που εξήχθησαν απ΄ το Twitter. Καταλήγοντας, είναι ξεκάθαρο ότι ο συνδυασμός όλων των τεχνικών που εφαρμόστηκαν, αποτελούν καθοριστικό παράγοντα στην επίδοση των συστημάτων αναγνώρισης συμβάντων.

Κεφάλαιο 7

Επίλογος

7.1 Συμπεράσματα

Τα συστήματα αναγνώρισης συμβάντων σε κοινωνικά δίκτυα, τα τελευταία χρόνια, παρουσιάζουν σημαντική εξέλιξη. Η παρούσα διπλωματική εργασία βοηθά στην πρόσθετη εξέλιξή τους, κυρίως από δύο σκοπιές. Η πρώτη, αφορά τη σημασία που έχει ο τρόπος προσέγγισης του προβλήματος της αναγνώρισης συμβάντων, ανάλογα με το κοινωνικό δίκτυο που χρησιμοποιείται. Κατ΄ επέκταση, οποιαδήποτε μορφή σύστηματος καταχώρησης πληροφοριών, που έχουν δημιουργηθεί απ΄ τους χρήστες, μπορεί να υποβληθεί σε διαδικασία ανάλυσης και κατηγοριοποίησης των παραχθέντων δεδομένων. Η δεύτερη αφορά τους διαφορετικούς ρόλους που διαδραματίζουν οι διάφορες τεχνικές επεξεργασίας φυσικής γλώσσας και τη σημασία τους στην απόδοση του τελικού συστήματος. Ένα τέτοιο σύστημα θα μπορούσε να φανεί χρήσιμο, τόσο σε επιχειρήσεις μάρκετινγκ και ηλεκτρονικού τύπου, όσο και στον κλάδο της ψυχολογίας. Αυτό συμβαίνει γιατί δίνεται η δυνατότητα σε ερευνητές να μελετήσουν την επίδραση διαφορετικών χαρακτηριστικών, όπως φυλετικά ή γεωγραφικά κλπ, στην διαμόρφωση και στην έκφραση της κοινής γνώμης.

Συμπερασματικά, το σύστημα που αναπτύχθηκε, στο πλαίσιο αυτής της διπλωματικής εργασίας, είναι ένα πλήρες σύστημα αναγνώρισης συμβάντων βασισμένο σε τεχνικές επεξεργασίας φυσικής γλώσσας και μηχανικής μάθησης, το οποίο καθιστά δυνατή την αναζήτηση και ανάλυση σημαντικών συμβάντων ανά τον κόσμο με ένα διαφορετικό τρόπο απ΄ ότι τα προϋπάρχοντα συστήματα.

7.2 Μελλοντικές Επεκτάσεις

Το τελικό σύστημα που υλοποιήσαμε θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω, τουλάχιστον ως προς τρεις κατευθύνσεις. Συγκεκριμένα, αναφέρονται τα ακόλουθα:

- Ενσωμάτωση διαδικασίας αναγνώρισης πολυγλωσσικού περιεχομένου στα κατηγοριοποιημένα συμβάντα. Η λειτουργία αυτή θα μπορούσε να αποτελέσει το επόμενο στάδιο της ομαδοποίησης των συμβάντων. Με τον τρόπο αυτό, θα επιτυγχάνεται ευκολότε-

ρη απομόνωση των ελάχιστου ενδιαφέροντος συμβάντων, καθώς και η επισήμανση των σημαντικών συμβάντων, ανεξαρτήτως γλωσσικών περιορισμών.

- Δυνατότητα εφαρμογής του συστήματος σε ζωντανή ροή δεδομένων. Η λειτουργία αυτή θα μπορούσε να υλοποιηθεί, κάνοντας χρήση τεχνικών ανάλυσης Big Data, όπως είναι το Spark, για κατανεμημένα συστήματα.

- Επιπλέον επέκταση του συστήματος, ως προς την διαχείριση μεγάλου όγκου δεδομένων, με τεχνικές locality sensitive hashing, για την περαιτέρω βελτίωση της χρονικής του πολυπλοκότητας.

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Αλγορίθμων

Βιβλιογραφία

[1] Daiber, Joachim, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. "Improving Efficiency and Accuracy in Multilingual Entity Extraction." In Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS '13, 121. Graz, Austria: ACM Press. https://doi.org/10.1145/2506182.2506198.

[2] Rong, Xin. 2014. "Word2vec Parameter Learning Explained." ArXiv:1411.2738 [Cs], November. http://arxiv.org/abs/1411.2738.

[3] Wevi, word embedding visual inspector, https://ronxin.github.io/wevi/

[4] Spacy, Industrial-Strength Natural Language Processing, https://spacy.io/

[5] Twiiter, Twitter API, https://developer.twitter.com/en/docs.html

[6] Flickr, Flickr API, https://www.flickr.com/services/api/

[7] Zubiaga, Arkaitz, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. "Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads." Edited by Naoki Masuda. PLOS ONE 11 (3): e0150989. https://doi.org/10.1371/journal.pone.0150989.

[8] Petrović, Saša, Miles Osborne, and Victor Lavrenko. 2012. "Using Paraphrases for Improving First Story Detection in News and Twitter." In , 338–46.

[9] Papadopoulos, Symeon, Raphaël Troncy, Vasileios Mezaris, Benoit Huet, and Ioannis Kompatsiaris. 2013. "Social Event Detection at MediaEval 2013: Challenges, Dataset and Evaluation." In .

[10] Manchon-Vizuete, D., Gris-Sarabia, I., Giro-i-Nieto, G. UPC at MediaEval 2014 Social Event Detection Task. Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain, October 16-17, 2014, CEUR-WS.org, online ceur-ws.org/Vol-1263/mediaeval2014_submission_58.pdf

[11] Pandas, Python Data Analysis Library, https://pandas.pydata.org/

[12] K-Means Clustering Algorithm, https://en.wikipedia.org/wiki/K-means_clustering

[13] Hierarchical Clustering Algorithm, https://en.wikipedia.org/wiki/Hierarchical_clustering

[14] SUPER - Social sensors for security assessments and proactive emergencies management

Συντομογραφίες - Αρκτικόλεξα - - Ακρωνύμια

κλπ	και λοιπά
CSV	Comma-Seperated Values
JSON	JavaScript Object Notation
XML	Extensible Markup Language
FSD	First Story Detection
API	Application Programming Interface
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
OPTICS	Ordering Points To Identify the Clustering Structure
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
NLP	Natural Language Processing
w2v	Words to Vectors
CBOW	Continuous Bag of Words
CSG	Continuous Skip Gram

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
Ανεξέλεγκτη Πληροφορία	Noise
Αναγνώριση Συμβάντων	Event Detection
Επεξεργασία Φυσικής Γλώσσας	Natural Language Processing
Μηχανική Μάθηση	Machine Learning
Πρόβλημα Κατηγοριοποίησης	Classification Problem
Πρόβλημα Ομαδοποίησης	Clustering Problem
Συστάδα	Cluster
Διεπαφή	Interface
Ομαδοποίηση	Clustering
Αλγόριθμοι Ομαδοποίησης	Clustering Algorithms
Επιστήμη των Υπολογιστών	Computer Science
Μηχανική Πληροφοριών	Information Engineering
Τεχνητή Νοημοσύνη	Artificial Intelligence
Διασύνδεση Οντοτήτων	Entity Linking
Διασυνδεδεμένο Ανοιχτό Σύννεφο Δεδομένων	Linked Open Data Cloud
Λεξική Ενσωμάτωση	Word Embeddings
Ρηχά Νευρωνικά Δίκτυα	Shallow Neural Networks
κωδικοποιημένο	one-hot encoded
κωδικοποιημένες αναπαραστάσεις	one-hot encoding representations
Βαθιά Μάθηση (Μηχανική)	Deep Learning
μη-καταστρεπτική αναγνώριση λεκτικών μονάδων	non-destructive tokenization
Επισήμανση μερών του λόγου	POS tagging
τελικά σημεία	API endpoints
πλαίσιο δεδομένων	Dataframe
Σύνολα Δεδομένων	Datasets
σχεδιάγραμμα	schema
Αυστηρή ανάλυση συστάδων	Hard Clustering
Πιθανοτική ανάλυση συστάδων	Soft Clustering
επισήμανση χρήστη	mention
αυτοματοποιημένη διαφοροποίηση	Backpropagation