

**Ανάλυση δικτύων
που σχηματίζονται γύρω από λογαριασμούς
πολιτικών κομμάτων στο Twitter**

Καραμπίνας Ευάγγελος

Διπλωματική Εργασία

Επιβλέπων: Ε. Πιτουρά



**ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA**

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πολύ την καθηγήτρια κυρία Ευαγγελία Πιτουρά για την πολύτιμη βοήθεια και καθοδήγηση κατα τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Οκτώβριος 2020

Καραμπίνας Ευάγγελος

Περίληψη

Το περιεχόμενο της συγκεκριμένης διπλωματικής εργασίας είναι η μελέτη των δικτύων που σχηματίζονται στο Twitter γύρω από δύο μεγάλους λογαριασμούς πολιτικών κομμάτων στην Ελλάδα με στόχο την εξαγωγή πληροφορίας που αφορά τους πιο σημαντικούς χρήστες σε κάθε γράφο, τα communities που σχηματίζονται μεταξύ των χρηστών, τον προσδιορισμό του περιεχομένου του κάθε community με βάση του hashtag που έχουν χρησιμοποιήσει πιο πολύ οι χρήστες του και τέλος την εξέλιξη αυτών των communities στο χρόνο. Επίσης παρουσιάζεται ένα σύνολο από αποδοτικές τεχνικές που χρησιμοποιήσαμε για να συλλέξουμε τα δεδομένα που μας ενδιαφέρουν από το Twitter και ο τρόπος με τον οποίο τα αποθηκεύσαμε ώστε να είναι εύκολη η προσπέλασή τους. Όλα τα κομμάτια της εργασίας έχουν υλοποιηθεί σε Python3. Για να συνδεθούμε στο TwitterAPI χρησιμοποιήσαμε τη βιβλιοθήκη Tweepy. Η κατασκευή των γραφών έχει γίνει με το module NetworkX ενώ η οπτικοποίηση με το εργαλείο Gephi.

Λέξεις Κλειδιά: Κοινωνικά δίκτυα, Εντοπισμός Communities, Εξέλιξη Communities, Twitter, Python3

Abstract

The purpose of this thesis is to study the Twitter networks that form around the two most popular political party accounts in Greece, so that we can extract information about the most important Twitter users in each network, the communities that form between the users, the context of each community based on the most used hashtag in the community and finally the way these communities evolve in time. Also, we present a set of efficient techniques we used to collect data from Twitter and how we saved this data so that it is easily accessible. All the code parts of this thesis are implemented in Python3. To connect to the TwitterAPI we used the Tweepy library. The construction of the graphs was made using the NetworkX module. To visualise the graphs we used the graph visualisation tool Gephi.

Keywords: Social Networks, Communities Detection, Communities Evolution, Twitter, Python3

Πίνακας Περιεχομένων

1. Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.2 Οργάνωση τόμου	1
2. Συλλογή δεδομένων	3
2.1 Συλλογή λογαριασμών χρηστών.....	3
2.1.1 Περιγραφή	3
2.1.2 Σχεδίαση	3
2.2 Συλλογή πληροφορίας των σχέσεων μεταξύ λογαριασμών χρηστών.....	6
2.2.1 Περιγραφή	6
2.2.2 Σχεδίαση	7
2.3 Συλλογή των tweets και των hashtags	8
2.2.1 Περιγραφή	8
2.2.2 Σχεδίαση	8
2.4 Συλλογή retweets	10
2.2.1 Περιγραφή	10
2.2.2 Σχεδίαση	10
3. Ανάλυση Δεδομένων	13
3.1 Dataset	13
3.2 Κατασκευή δικτύων	13
3.3 Οπτικοποίηση δικτύων	15
3.4 Εύρεση και ανάλυση communities του Retweet γράφου	16
3.5 Μελέτη communities στον χρόνο	21
3.6 Εντοπισμός δυναμικών communities	34
4. Επίλογος	41
4.1 Σύνοψη	41
5. Βιβλιογραφία	43

1. Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Το Twitter αποτελεί ένα τεράστιο κοινωνικό δίκτυο μέσα στο οποίο οι χρήστες του ανταλάσουν καθημερινά μεγάλο όγκο πληροφορίας. Αυτή την πληροφορία μπορούμε να τη συλλέξουμε, να τη μοντελοποιήσουμε και να κατασκευάσουμε διάφορους γράφους όπως για παράδειγμα γράφους που αφορούν το ποιος χρήστης στο Twitter ακολουθεί ποιόν ή γράφους για το ποιός χρήστης έχει κάνει retweet μια δημοσίευση ενός άλλου χρήστη. Έχοντας στη διάθεση μας τους παραπάνω γράφους μπορούμε να τους μελετήσουμε και να εφαρμόσουμε επάνω τους γνωστούς αλγορίθμους με στόχο την εξαγωγή πληροφορίας. Με αυτόν τον τρόπο είμαστε σε θέση να παρατηρήσουμε τις σχέσεις μεταξύ των στοιχείων που περιέχονται σε ένα γράφο, τόσο σε μια συγκεκριμένη χρονική στιγμή όσο και κατα τη διάρκεια της εξέλιξής τους στο χρόνο.

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η μελέτη των δικτύων γύρω από δύο μεγάλους λογαριασμούς ελληνικών πολιτικών κομμάτων στο Twitter, της Νεάς Δημοκρατίας (@neademokratia) και του Σύριζα (@syriza_gr), με στόχο τον εντοπισμό ομάδων (communities) που σχημαντίζονται μεταξύ των χρηστών σε κάθε δίκτυο, τα πιο δημοφιλή hashtags σε κάθε community και τέλος την εξέλιξη των communities στο χρόνο (dynamic communities). Με αυτόν τον τρόπο μπορούμε να προσδιορίσουμε το περιεχόμενο του κάθε community βάση του hashtag που έχουν χρησιμοποιήσει περισσότερο οι χρήστες του, και πως το περιεχόμενο κάθε community αλλάζει σε διαφορετικές χρονικές περιόδους. Πιο συγκεκριμένα θα εστιάσουμε στην ανάλυση του Retweet γράφου γύρω από τους λογαριασμούς neademokratia, syriza_gr αλλά και του γράφου που σχηματίζεται γύρω και από τους δύο λογαριασμούς. Θα εντοπίσουμε σε κάθε γράφο τα communities μεταξύ των χρηστών συνολικά στο χρόνο, αλλά και ξεχωριστά για τις χρονικές περιόδους που αφορούν τους μήνες Ιανουάριο, Φεβρουάριο, Μάρτιο, Απρίλιο του 2020. Θα δούμε την αντιστοιχία μεταξύ των communities μιας χρονικής περιόδου με τα communities επόμενων χρονικών περιόδων και θα ανακαλύψουμε ποιο είναι το περιέχομενο του κάθε community, για ποιό θέμα δηλαδή μιλάνε οι χρήστες του, κάνοντας χρήση των hashtags.

Προκειμένου να πραγματοποιήσουμε την παραπάνω ανάλυση πρέπει πρώτα να συλλέξουμε τα απαραίτητα δεδομένα, για την κατασκευή των γραφών γύρω από τους δύο λογαριασμούς. Η συλλογή των δεδομένων και η επεξεργασία τους μπορεί να αποτελέσει μια αρκετά χρονοβόρα διαδικασία, άρα καλούμαστε να υλοποιήσουμε ένα αποδοτικό εργαλείο που θα συλλέγει τα δεδομένα σε αποδεκτό χρονικό διάστημα και θα τα οργανώνει κατάλληλα ώστε η επεξεργασία τους να γίνεται με εύκολο τρόπο. Θα δούμε άρα τις κατηγορίες των δεδομένων που χρειάστηκε να συλλέξουμε και θα περιγράψουμε αναλυτικά την σχεδίαση του εργαλείου που μας βοήθησε ώστε να συλλέξουμε αυτά τα δεδομένα.

1.2 Οργάνωση τόμου

Η διπλωματική εργασία αποτελείται από πέντε κεφάλαια. Το περιεχόμενο κάθε κεφαλαίου περιγράφεται παρακάτω:

Το κεφάλαιο δύο αφορά την συλλογή των δεδομένων γύρω από τους δύο πολιτικούς λογαριασμούς. Πιο συγκεκριμένα περιγράφει τις κατηγορίες των δεδομένων που χρειαζόμαστε και τον σχεδιασμό/ υλοποίηση του εργαλείου που θα συλλέξει και θα αποθηκεύσει αυτά τα δεδομένα.

Το κεφάλαιο τρία αφορά την ανάλυση των δεδομένων. Περιέχει πληροφορίες για το πως από τα δεδομένα μπορούμε να κατασκευάσουμε και οπτικοποιήσουμε τους γράφους που μας ενδιαφέρουν. Επίσης περιγράφει τον τρόπο με τον οποίο εντοπίσαμε τα communities σε κάθε γράφο και παρουσιάζει αναλυτικά τα αποτελέσματα από τον εντοπισμό των communities για κάθε χρονική στιγμή. Τέλος, περιγράφεται ο τρόπος με τον οποίο μελετήσαμε την εξέλιξη των communities στο χρόνο και παρουσιάζονται τα αποτελέσματα.

Το τέταρτο κεφάλαιο αποτελεί τον επίλογο και περιέχει μια σύνοψη της διπλωματικής εργασίας και μελλοντικές επεκτάσεις που μπορούμε να υλοποιήσουμε.

Τέλος, στο πέμπτο κεφάλαιο γίνεται αναφορά στην βιβλιογραφία που χρησιμοποιήθηκε για την υλοποίηση της συγκεκριμένης διπλωματικής εργασίας.

2.

Συλλογή Δεδομένων

Το Twitter αποτελεί ένα πολύ μεγάλο κοινωνικό δίκτυο, με τεράστιο αριθμό χρηστών και μεγάλο όγκο πληροφορίας. Οι λογαριασμοί των πολιτικών κομμάτων που επιλέξαμε είναι μέρος αυτού του δικτύου καθιστώντας έτσι τη συλλογή και την αποθήκευση ολόκληρης της πληροφορίας υπερβολικά χρονοβόρα. Σε αυτή την ενότητα εστιάζουμε στο πως θα επιλέξουμε ένα μικρότερο δίκτυο γύρω από κάθε πολιτικό λογαριασμό, στον τρόπο με τον οποίο θα φιλτράρουμε και θα αποθηκεύσουμε αυτόν τον τεράστιο όγκο πληροφορίας ώστε να κρατήσουμε τις κατηγορίες των δεδομένων που μας ενδιαφέρουν και τέλος στις τεχνικές που εφαρμόσαμε για να επιταχύνουμε την παραπάνω διαδικασία.

2.1 Συλλογή λογαριασμών χρηστών

2.1.1 Περιγραφή

Όπως αναφέραμε παραπάνω οι λογαριασμοί των πολιτικών κομμάτων που επιλέξαμε αποτελούν μέρος ενός τεράστιου δικτύου. Για να μειώσουμε την πολυπλοκότητα και τον χρόνο συλλογής της πληροφορίας που περιέχεται σε αυτό το δίκτυο, θα επιλέξουμε ένα μικρότερο υποδίκτυο γύρω από κάθε πολιτικό λογαριασμό. Σε κάθε υποδίκτυο μας ενδιαφέρει να συλλέξουμε τους χρήστες από τους οποίους αποτελείται και να τους αποθηκεύσουμε με τέτοιο τρόπο ώστε η προσπέλασή τους και η αναζήτηση πάνω σε αυτούς να γίνεται εύκολα.

2.1.2 Σχεδίαση

Προκειμένου να διευκολύνουμε την διαδικασία κατασκευής των δύο δικτύων μπορούμε να αναπαραστήσουμε το κάθε δίκτυο ως ένα γράφο (G) όπου:

- Κάθε κόμβος u αντιστοιχεί σε ένα λογαριασμό χρήστη στο Twitter. Δύο κόμβοι u1, u2 δεν μπορούν να αντιστοιχούν στον ίδιο λογαριασμό χρήστη.
- Κάθε ακμή v από έναν κόμβο u1 σε ένα κόμβο u2 προσδιορίζει ότι ο λογαριασμός χρήστη που αντιστοιχεί στον u1 ακολουθεί τον λογαριασμό χρήστη που αντιστοιχεί στον u2.

Μπορούμε έτσι να εκφράσουμε οποιοδήποτε δίκτυο ως ένα σύνολο από κόμβους και ακμές.

Έχοντας τον παραπάνω ορισμό θα σχεδιάσουμε έναν αλγόριθμο για την κατασκευή του δικτύου γύρω από κάθε πολιτικό λογαριασμό. Η ιδέα του αλγορίθμου είναι απλή, ξεκινώντας από τον λογαριασμό που μας ενδιαφέρει παίρνουμε ένα συγκεκριμένο αριθμό χρηστών που τον ακολουθούν (σύνολο S1), ύστερα για καθένα χρήστη από το σύνολο S1 παίρνουμε ένα συγκεκριμένο αριθμό χρηστών που τον ακολουθούν και τους προσθέτουμε στο σύνολο S2, ύστερα για καθένα χρήστη από το σύνολο S2 παίρνουμε ένα συγκεκριμένο αριθμό χρηστών που τον ακολουθούν και

τους προσθέτουμε στο σύνολο S3 και ούτω καθεξής. Με αυτή τη διαδικασία έχουμε καταφέρει να πάρουμε ένα σύνολο χρηστών (δείγμα) που βρίσκεται γύρω από τον λογαριασμό που μας ενδιαφέρει. Παρακάτω βλέπουμε τον αλγόριθμο σε ψευδοκώδικα και ένα παράδειγμα του δικτύου που κατασκευάζεται κατά την εκτέλεση του.

Αρχή

`root = λογαριασμός χρήστη γύρω από τον οποίο θέλουμε να κατασκευάσουμε το δίκτυο`

`Users = [root]`

Για κάθε i από 1 έως n :

`Followers = []`

Για κάθε χρήστη u στη λίστα `Users`:

Πάρε k χρήστες που ακολουθούν τον u και πρόσθεσε τους στην λίστα `Followers`

Για κάθε χρήστη u' στη λίστα `Followers`:

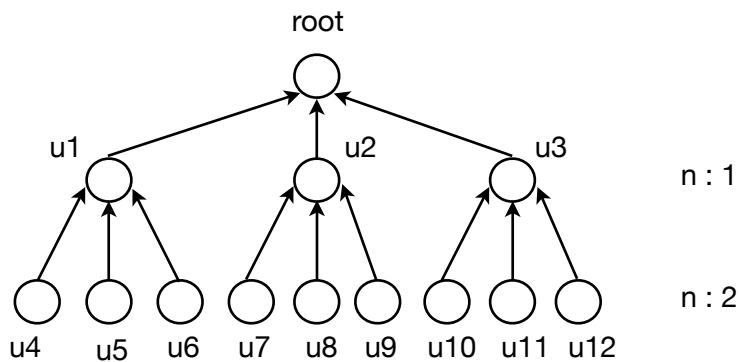
Αποθήκευσε τον u' στον δίσκο αν δεν είναι ήδη αποθηκευμένος

`Users = Followers`

Τέλος

Εικόνα 2.1.1 Αλγόριθμος συλλογής λογαριασμών χρηστών σε ψευδοκώδικα

Παράδειγμα εκτέλεσης του αλγορίθμου για $n=2$, $k=3$



Λογαριασμοί χρηστών αποθηκευμένοι στο δίσκο:

`root, u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11`

Εικόνα 2.1.2 Παράδειγμα εκτέλεσης αλγορίθμου για $n=2$, $k=3$

Ο αλγόριθμος όπως είδαμε παραπάνω παίρνει με έναν απλό τρόπο ένα δείγμα των χρηστών που βρίσκονται γύρω από έναν λογαριασμό στο δίκτυο του Twitter. Παρόλα αυτά, η εκτέλεση του είναι αρκετά χρονοβόρα λόγω της πολιτικής που επιβάλει το Twitter στον αριθμό των αιτημάτων που φτάνουν στους servers. Πιο συγκεκριμένα, κάθε συνδυασμός {api_key , api_secret_key}, με τα οποία γίνεται το authentication μιας εφαρμογής από το Twitter API, έχει ένα άνω όριο στον ρυθμό με τον οποίο μια εφαρμογή μπορεί να στέλνει αιτήματα και να λαμβάνει απαντήσεις από τους servers.

Καταλήξαμε έτσι, σε μια πολυνηματική εκδοχή του παραπάνω αλγόριθμου, όπου το κάθε νήμα χρησιμοποιεί τον δικό του μοναδικό συνδυασμό κλειδιού, μυστικού κλειδιού για να στείλει οποιοδήποτε αίτημα στους servers. Παρακάτω περιγράφεται ο αλγόριθμος σε ψευδοκώδικα.

Αρχή

Αντιστοίχησε σε κάθε νήμα ένα μοναδικό συνδυασμό {api_key, api_secret_key}

root = λογαριασμός χρήστη γύρω από τον οποίο θέλουμε να κατασκευάσουμε το δίκτυο

Users = [root]

Για κάθε i από 1 έως n:

Followers = []

Για κάθε thread:

Εκτέλεσε τη συνάρτηση fetch_friends(api_key ,api_secret_key ,Users ,Followers)

Περίμενε μέχρι να τελειώσουν όλα τα threads

Users = Followers

Τέλος

fetch_friends(api_key, api_secret_key, Users, Followers):

Όσο υπάρχουν χρήστες στη λίστα Users:

u = Users.pop()

tmp_followers = Πάρε k χρήστες που ακολουθούν τον u χρησιμοποιώντας το μοναδικό για κάθε thread api_key,api_secret_key

Πρόσθεσε τους tmp_followers στην λίστα Followers

Για κάθε χρήστη u' στη λίστα tmp_followers:

Αποθήκευσε τον u' στον δίσκο αν δεν είναι ήδη αποθηκευμένος

Εικόνα 2.1.3. Αλγόριθμος συλλογής λογαριασμών χρηστών με χρήση νημάτων σε ψευδοκώδικα

Ο αλγόριθμος διατηρεί ένα pool από χρήστες (Users), κάθε thread αφαιρεί έναν χρήστη από το pool, βρίσκει k χρήστες που τον ακολουθούν, τους προσθέτει στη λίστα Followers και τους αποθηκεύει στο δίσκο. Η διαδικασία αυτή συνεχίζεται μέχρι το pool να αδειάσει. Όταν αδειάσει, προσθέτουμε τους χρήστες της λίστας Followers στο pool (Users) και ο αλγόριθμος ξεκινάει από την αρχή.

Με τον παραπάνω αλγόριθμο καταφέραμε να μειώσουμε τον χρόνο συλλογής των χρηστών σημαντικά. Πιο συγκεκριμένα, με την χρήση n πλήθους νημάτων ο χρόνος για να ολοκληρωθεί η εκτέλεση του αλγορίθμου είναι περίπου n φορές πιο γρήγορη από αυτή του αλγορίθμου που δεν χρησιμοποιεί νήματα.

Μέχρι στιγμής είδαμε τον τρόπο με τον οποίο συλλέγουμε τους λογαριασμούς χρηστών, παρακάτω περιγράφεται ο τρόπος με τον οποίο αποθηκεύονται στον δίσκο.

Η σωστή οργάνωση των χρηστών και η εύκολη προσπέλασή\αναζήτησή τους είναι σημαντική καθώς διευκολύνει αρκετά την διαδικασία της ανάλυσης των δεδομένων που ακολουθεί αργότερα. Για αυτό το λόγω αποφασίσαμε για την αποθήκευση των δεδομένων να χρησιμοποιήσουμε μια σχεσιακή βάση δεδομένων. Για την αποθήκευση των χρηστών σχιδιάσαμε ένα πίνακα στην βάση με όνομα `twitter_user` ο οποίος περιέχει τα παρακάτω πεδία:

`id_str` αντιστοιχεί στο μοναδικό id του χρήστη στο Twitter, `screen_name` αντιστοιχεί στο όνομα του χρήστη στο Twitter, `followers_count` αντιστοιχεί στο αριθμό των λογαριασμών που ακολουθεί ο χρήστης, `friends_count` αντιστοιχεί στον αριθμό των λογαριασμών που ακολουθούν τον χρήστη, `level`, `statuses_count` αντιστοιχεί στον αριθμό των tweets που έχει κάνει ο χρήστης, `graph_name` αντιστοιχεί στο όνομα του υποδικτύου που ανήκει ο χρήστης.

twitter_user	
<code>id_str</code>	text
<code>screen_name</code>	text
<code>followers_count</code>	int
<code>friends_count</code>	int
<code>level</code>	int
<code>statuses_count</code>	int
<code>graph_name</code>	text

Εικόνα 2.1.4. Σχήμα της βάσης

Ο αλγόριθμος συλλογής των λογαριασμών χρηστών υλοποιείται από το python script `data/collect_users.py`. Για την σύνδεση και την εύκολη αποστολή αιτήματων στο TwitterAPI χρησιμοποιήθηκε το module Tweepy. Η σχεσιακή βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των χρηστών είναι η Postgresql.

2.2 Συλλογή πληροφορίας των σχέσεων μεταξύ των λογαριασμών χρηστών

2.2.1 Περιγραφή

Εκτός από τους χρήστες που περιέχονται στα δύο δίκτυα μας ενδιαφέρουν και οι σχέσεις που έχουν αυτοί οι χρήστες μεταξύ τους. Μας ενδιαφέρει δηλαδή να βρούμε ποιους χρήστες ακολουθεί ένας χρήστης και από ποιους άλλους ακολουθείται. Παρακάτω θα περιγράψουμε τον τρόπο με τον οποίο θα συγκεντρώσουμε αυτές τις πληροφορίες και το πως θα τις αποθηκεύσουμε στο δίσκο.

2.2.2 Σχεδίαση

Ο τρόπος με τον οποίο θα βρούμε τις σχέσεις μεταξύ των χρηστών είναι απλός, αρκεί για κάθε χρήστη να βρούμε ποιους χρήστες ακολουθεί μέσα στο δίκτυο. Όπως και πριν έτσι και εδώ αντιμετωπίζουμε το πρόβλημα του μεγάλου χρόνου εκτέλεσης λόγω των ορίων που επιβάλει το Twitter στα αιτήματα που δέχεται. Έτσι και εδώ χρειαζόμαστε έναν απλό αλγόριθμο ο οποίος χρησιμοποιεί νήματα και αντιστοιχεί σε κάθε νήμα το δικό του {api_key,api_secret_key}, αποστέλοντας έτσι μεγαλύτερο όγκο αιτημάτων στο ίδιο χρονικό διάστημα. Ο αλγόριθμος περιγράφεται παρακάτω:

Αρχή
Users = Πάρε από την βάση τους χρήστες για ένα δίκτυο
Χώρισε τους Users σε τόσα batches όσος είναι ο αριθμός των νημάτων
Αντιστοίχησε σε κάθε νήμα ένα μοναδικό {api_key,api_secret_key} και ένα batch από Users
Κάθε νήμα:
Για κάθε χρήστη u στο batch:
follow_users = Πάρε όλους τους χρήστες που ακολουθεί ο u
Για κάθε χρήστη u' στη λίστα follow_users:
Αν ο u' υπάρχει στους Users τοτε:
Αποθήκευσε το ζευγάρι (u,u')
Τέλος

Εικόνα 2.2.1. Αλγόριθμος συλλογής πληροφορίας των σχέσεων μεταξύ των χρηστών με νήματα σε ψευδοκώδικα

Ο αλγόριθμος χωρίζει τους χρήστες σε n ισομεγέθη τμήματα(batches), όπου n είναι ο αριθμός των νημάτων που έχουμε επιλέξει για να τρέξουμε τον αλγόριθμο. Κάθε τμήμα από χρήστες αντιστοιχίζεται σε ένα νήμα. Κάθε νήμα διατρέχει τους χρήστες που περιέχονται στο batch του, και για κάθε χρήστη(u) βρίσκει τους χρήστες που ακολουθεί(follow_users). Για κάθε χρήστη u' από τους follow_users, ελέγχει αν ο u' υπάρχει στη λίστα Users και αν ναι αποθηκένει το ζευγάρι (u,u') στο δίσκο. Όπως και στον αλγόριθμο της ενότητας 2.1.2 έτσι και εδώ χρησιμοποιώντας n αριθμό από νήματα ο χρόνος εκτέλεσης είναι η φορές πιο γρήγορος από το να χρησιμοποιούσαμε ένα νήμα.

twitter_user		followship	
id_str	text	follower_id	text
screen_name	text	followee_id	text
followers_count	int	graph_name	text
friends_count	int		
level	int		
statuses_count	int		
graph_name	text		

Για την αποθήκευση του ζευγαριού (u,u') (ο χρήστης u ακολουθεί το χρήστη u') προσθέτουμε έναν καινούργιο πίνακα στη βάση δεδομένων με όνομα followship. Ο πίνακας followship περιέχει τα παρακάτω πεδία: **follower_id** αυτός που ακολουθεί, **followee_id** αυτός που ακολουθείται, **graph_name** το όνομα του δικτύου στο οποίο ανήκει η σχέση.

Εικόνα 2.2.2. Σχήμα της βάσης

Ο αλγόριθμος συλλογής πληροφορίας των σχέσεων μεταξύ των χρηστών υλοποιείται από το python script data/complete_relations.py. Για την σύνδεση και την εύκολη αποστολή αιτήματων στο TwitterAPI χρησιμοποιήθηκε το module Tweepy. Η σχεσιακή βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των σχέσεων είναι η Postgresql.

2.3 Συλλογή των tweets και των hashtags

2.3.1 Περιγραφή

Σε αυτή την ενότητα εστιάζουμε στην συγκέντρωση όλων των tweets που έχουν δημοσιεύσει οι χρήστες σε κάθε δίκτυο και το φιλτράρισμα της πληροφορίας που περιέχει κάθε tweet ώστε να αποθηκεύσουμε μόνο τα πεδία που μας ενδιαφέρουν. Ιδιαίτερη σημασία έχει το πεδίο των hashtags που περιέχει κάθε tweet καθώς θα χρειαστεί στην φάση της ανάλυσης των δεδομένων.

2.3.2 Σχεδίαση

Ο αλγόριθμος για την συλλογή των tweets και των hashtags έχει παρόμοια δομή με αυτόν της συλλογής των σχέσεων μεταξύ των λογαριασμών χρηστών (Ενότητα 2.2.2, σχήμα 4). Ο αλγόριθμος περιγράφεται παρακάτω:

Αρχή

Users = Πάρε από την βάση τους χρήστες για ένα δίκτυο

Χώρισε τους Users σε τόσα batches όσος είναι ο αριθμός των νημάτων

Αντιστοίχησε σε κάθε νήμα ένα μοναδικό {api_key,api_secret_key} και ένα batch από Users

Κάθε νήμα:

Για κάθε χρήστη u στο batch:

fetched_tweets = Πάρε τα k τελευταία tweets του χρήστη u

Για κάθε tweet στη λίστα fetched_tweets:

Αποθήκευσε το ζευγάρι (tweet, u_id)

Για κάθε hashtag που περιέχει το tweet

Αποθήκευσε το ζευγάρι (hashtag, tweet_id)

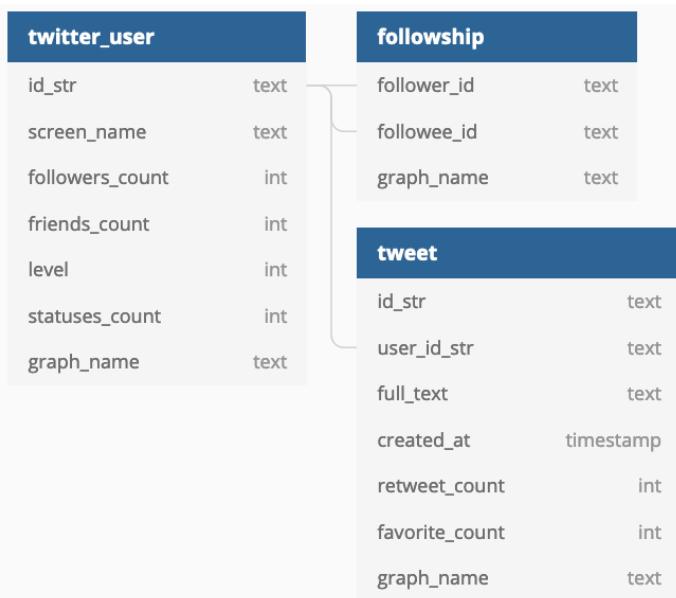
Τέλος

Εικόνα 2.3.1. Αλγόριθμος συλλογής tweets και hashtags με νήματα σε ψευδοκώδικα

Οπως βλέπουμε στον αλγόριθμο κάθε νήμα αναλαμβάνει ένα ξεχωριστό κομμάτι (batch) χρηστών. Για κάθε χρήστη u στο batch παίρνει από το Twitter τα k τελευταία tweets του u. Για κάθε tweet

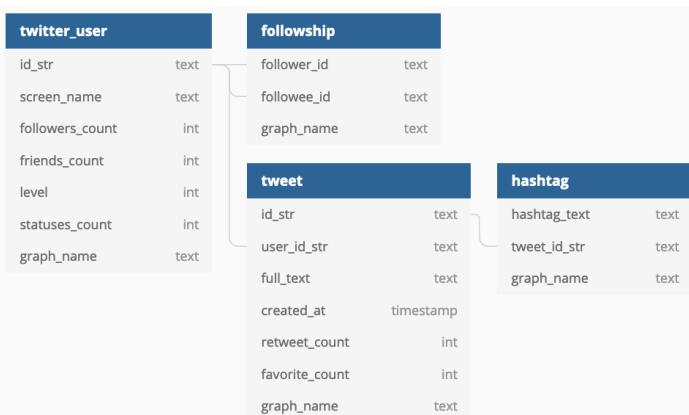
αποθηκεύει το ζευγάρι (tweet,u_id) (το tweet ανήκει στο χρήστη u), και για κάθε hashtag που περιέχεται στο tweet αποθηκεύει το ζευγάρι (hashtag,tweet_id) (το hashtag ανήκει στο tweet). Η χρήση των νημάτων και εδώ μειώνει τον χρόνο εκτέλεσης σημαντικά αφού για την αριθμό νημάτων ο χρόνος εκτέλεσης είναι η φορές μικρότερος.

Για την αποθήκευση των tweets στην βάση δεδομένων προσθέσαμε έναν καινούργιο πίνακα με όνομα tweet. Ο πίνακας tweet περιέχει τα πεδία: **id_str** αντιστοιχεί στο μοναδικό id του tweet στο Twitter, **user_id_str** αντιστοιχεί στο id του χρήστη στον οποίο ανήκει το tweet, **full_text** αντιστοιχεί στο κείμενο που περιέχει το tweet, **created_at** αντιστοιχεί στην ημερομηνία δημιουργίας του tweet, **retweet_count** αντιστοιχεί στο πόσες φορές έχεις γίνει retweet το συγκεκριμένο tweet, **favorite_count** αντιστοιχεί στο πόσα likes έχει το tweet, **graph_name** αντιστοιχεί στο όνομα του δικτύου στο οποίο ανήκει ο χρήστης που έκανε το tweet.



Εικόνα 2.3.2. Σχήμα της βάσης

Για την αποθήκευση των hashtags στην βάση προσθέσαμε ένα πίνακα με όνομα hashtag. Ο πίνακας hashtag περιέχει τα πεδία: **hashtag_text** αντιστοιχεί στο κείμενο του hashtag, **tweet_id_str** αντιστοιχεί στο id του tweet στο οποίο ανήκει το hashtag, **graph_name** αντιστοιχεί στο όνομα του δικτύου στο οποίο ανήκει ο χρήστης που χρησιμοποίησε το συγκεκριμένο hashtag.



Εικόνα 2.3.3. Σχήμα της βάσης

Ο αλγόριθμος συλλογής των tweets και των hashtags υλοποιείται από το python script data/collect_tweets.py. Για την σύνδεση και την εύκολη αποστολή αιτήματων στο TwitterAPI χρησιμοποιήθηκε το module Tweepy. Η σχεσιακή βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των tweets και των hashtags είναι η Postgresql.

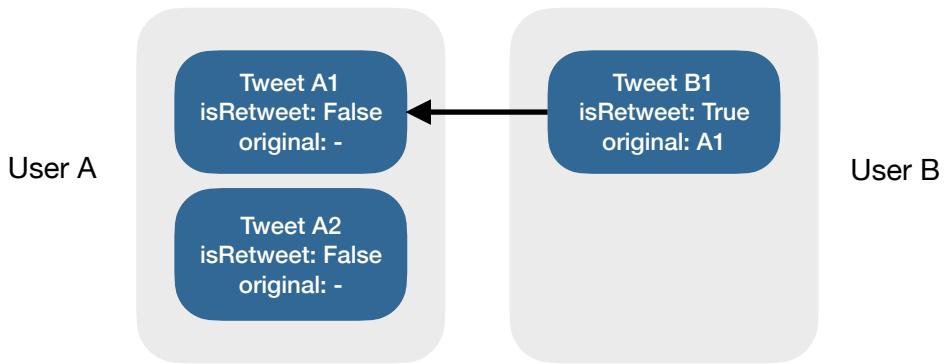
2.4 Συλλογή retweets

2.4.1 Περιγραφή

Έχοντας συγκεντρώσει και αποθηκεύσει τα tweets των χρηστών σε κάθε δίκτυο, εδώ μας ενδιαφέρει να βρούμε ποιά tweets αποτελούν retweet. Για την ανάλυση των δεδομένων χρειαζόμαστε επίσης για κάθε retweet το αρχικό tweet το οποίο έγινε retweet, καθώς και τον χρήστη στον οποίο ανήκει το αρχικό tweet.

2.4.2 Σχεδίαση

Στο Twitter ένας χρήστης μπορεί να κάνει retweet κάποιο tweet ενός άλλου χρήστη ή ακόμα και ένα δικό του. Στο παρκάτω σχήμα βλέπουμε τη σχέση μεταξύ ενός retweet και του αρχικού tweet:



Εικόνα 2.4.1. Σχέση μεταξύ retweet και αρχικού tweet

Η πληροφορία που θέλουμε να συγκεντρώσουμε είναι ότι το tweet B1 αποτελεί retweet, ότι το αρχικό tweet που τελικά έγινε retweet είναι το A1, και ότι το A1 ανήκει στον χρήστη User A. Ο αλγόριθμος που υλοποιεί την παραπάνω διαδικασία περιγράφεται παρακάτω:

Αρχή

tweets = Πάρε από την βάση τα tweets για ένα δίκτυο

Χώρισε τα tweets σε τόσα batches όσος είναι ο αριθμός των νημάτων

Αντιστοίχησε σε κάθε νήμα ένα μοναδικό {api_key,api_secret_key} και ένα batch από tweets

Κάθε νήμα:

Χώρισε το batch σε ίσα κομμάτια (chunks) των 100 tweets το καθένα

Για κάθε chunk:

fetched_tweets = Βρες από το Twitter όλες τις πληροφορίες για κάθε tweet στο chunk

Για κάθε tweet στο fetched_tweets:

Αν το tweet είναι retweet τότε:

Αποθήκευσε το ζευγάρι (tweet.id, tweet.original_tweet_id)

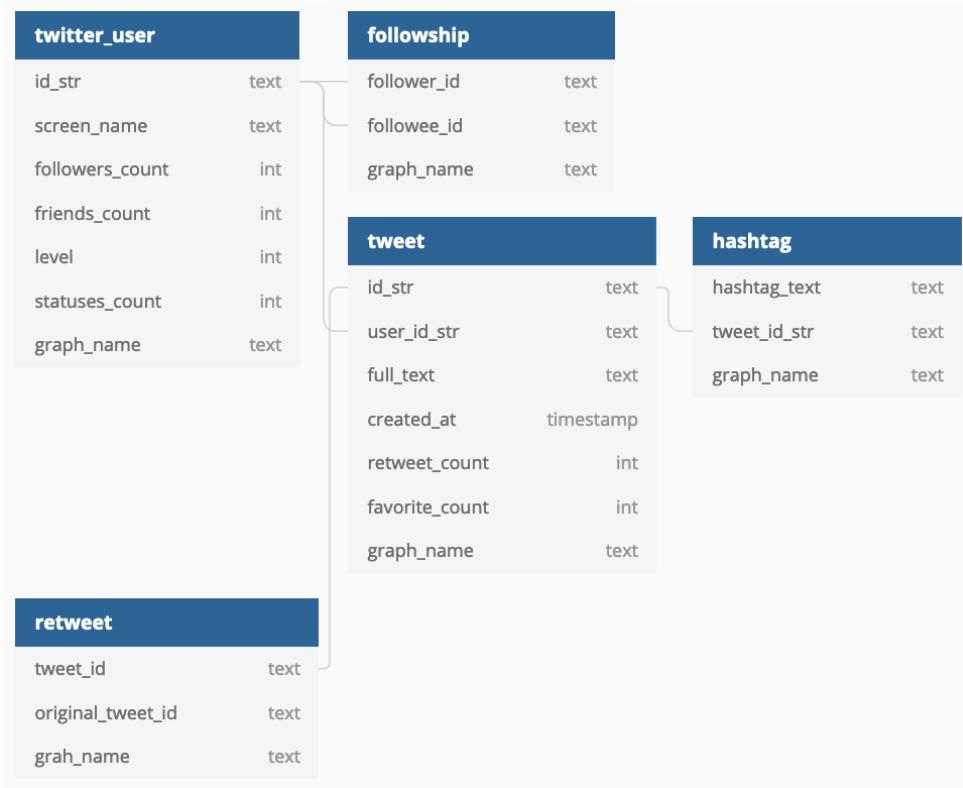
Τέλος

Εικόνα 2.4.2. Αλγόριθμος συλλογής retweet με νήματα σε ψευδοκώδικα

Παρατηρούμε ότι αλγόριθμος αφού χωρίσει τα tweets σε batches, ώστε κάθε νήμα να επεξεργαστεί το δικό του batch, μετά το κάθε batch χωρίζεται σε μικρότερα chunks των 100 tweets το καθένα. Ο λόγος που χωρίζουμε τα batches σε chunks είναι διότι το API του Twitter μπορεί σε ένα αίτημα να πάρει σαν παράμετρο μια λίστα από 100 tweets αντί να στέλνουμε ένα αίτημα για κάθε tweet ξεχωριστά. Με τον τρόπο αυτό και σε συνδυασμό με την χρήση των νημάτων μειώνουμε σημαντικά τον χρόνο εκτέλεσης του αλγορίθμου. Τέλος παρατηρούμε ότι για κάθε tweet που είναι retweet αποθηκεύουμε μόνο το id του και το id του αρχικού tweet, δεν χρειάζεται να αποθηκεύσουμε τον χρήστη στον οποίο ανήκει το αρχικό tweet καθώς μπορούμε να το βρούμε εύκολα από τον πίνακα tweet.

Για την αποθήκευση των retweets στην βάση δεδομένων προσθέσαμε ένα πίνακα με το όνομα retweet. Ο πίνακας περιέχει τα παρακάτω πεδία: **tweet_id** το οποίο αντιστοιχεί στο id του retweet, **original_tweet_id** το οποίο αντιστοιχεί στο id του αρχικού tweet που έγινε retweet, **graph_name** αντιστοιχεί στο όνομα του δικτύου που ανήκει ο χρήστης που έκανε το retweet.

Στο σχήμα της βάσης που φαίνεται παρακάτω παρατηρούμε ότι το πεδίο original_tweet_id του πίνακα retweet δεν έχει ξένο κλειδί το πεδίο id_str του πίνακα tweet. Αυτό επιτρέπει να μπαίνουν στην βάση retweets που το αρχικό τους tweet δεν ανήκει στον πίνακα των tweets και άρα μπορεί να ανήκει σε έναν χρήστη έξω από το δίκτυο χρηστών που έχουμε συλλέξει. Η αποφυγή αυτού του προβλήματος σε αυτή τη φάση είναι πολύ χρονοβόρα καθώς για κάθε αρχικό tweet θα πρέπει να ελέγχουμε αν περιέχεται στον πίνακα των tweets. Η επίλυση του προβλήματος γίνεται αργότερα στην ανάλυση των δεδομένων με ένα απλό join των πινάκων tweet και retweet.



Εικόνα 2.4.3. Σχήμα της βάσης

Ο αλγόριθμος συλλογής των retweets υλοποιείται από το python script `data/collect_retweets.py`. Για την σύνδεση και την εύκολη αποστολή αιτήματων στο TwitterAPI χρησιμοποιήθηκε το module Tweepy. Η σχεσιακή βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των retweets είναι η Postgresql.

3.

Ανάλυση Δεδομένων

Σε αυτό το κεφάλαιο παρουσιάζουμε τα χαρακτηριστικά των δεδομένων που συλλέξαμε στο κεφάλαιο 2, κατασκευάζουμε και οπτικοποιούμε δίκτυα με βάση αυτά τα δεδομένα και εφαρμόζουμε αλγορίθμους για την εύρεση κοινωνιών (communities) που διαμορφώνονται από τις σχέσεις μεταξύ των χρηστών. Τέλος, βλέπουμε σε κάθε δίκτυο ποιοι είναι οι πιο σημαντικοί λογαριασμοί χρήστη και για κάθε community ποιά είναι τα hashtags που χρησιμοποιήθηκαν περισσότερο.

3.1 Datasets

Κατά την συλλογή των λογαριασμών χρηστών συγκεντρώσαμε 9.629 λογαριασμούς χρηστών για το δίκτυο που αντιστοιχεί στο λογαριασμό @neademokratia και 9.922 λογαριασμούς χρηστών για το δίκτυο του @syriza_gr αντίστοιχα.

Από την συλλογή των δεδομένων για το ποιος χρήστης ακολουθεί ποιον συγκεντρώσαμε 170.329 σχέσεις μεταξύ των χρηστών για το δίκτυο του λογαριασμού @neademokratia και 273.430 για το δίκτυου του λογαριασμού @syriza_gr αντίστοιχα.

Από την συλλογή των tweets και των hashtags συγκεντρώσαμε 5.119.024 tweets και 2.153.293 hashtags για το δίκτυο του λογαριασμού @neademokratia, από τα παραπάνω hashtags ο αριθμός των μοναδικών hashtag (ένα hashtag μπορεί να έχει χρησιμοποιηθεί παραπάνω από μια φορές, εδώ μετράμε μόνο την πρώτη εμφανισή του) είναι 343.979. Αντίστοιχα για το δίκτυο που αντιστοιχεί στο λογαριασμό @syriza_gr συγκεντρώσαμε, 7.308.853 tweets και 2.240.258 hashtags, από τα παραπάνω hashtags ο αριθμός των μοναδικών hashtag είναι 329.613.

Τα παραπάνω αποτελέσματα παρουσιάζονται συνοπτικά στον πίνακα που ακολουθεί:

	Λογαριασμοί χρηστών	Σχέσεις μεταξύ χρηστών (follow)	Tweets	Hashtags
@neademokratia	9.629	170.329	5.119.024	2.153.293
@syriza_gr	9.922	273.430	7.308.853	2.240.258

Πίνακας 3.1 Παρουσίαση δεδομένων

3.2 Κατασκευή δικτύων

Εργαλεία: Python3, Postgresql, NetworkX, Gephi

Μέχρι αυτό το σημείο τα δεδομένα που έχουμε συλλέξει είναι αποθηκευμένα σε μια βάση δεδομένων. Προκειμένου να κατασκευάσουμε και να οπτικοποιήσουμε τα δίκτυα, πάνω στα οποία αργότερα θα βασιστεί η ανάλυση μας, πρέπει να αναπαραστήσουμε τα δεδομένα ως ένα σύνολο από κόμβους και

ακμές. Με αυτόν τον τρόπο θα να είναι εύκολο να διαβαστούν αργότερα από εργαλεία ανάλυσης και οπτικοποίησης δικτύων. Το εργαλείο που επιλέξαμε για να το πετύχουμε αυτό είναι το NetworkX. Το networkX είναι ένα python πακέτο για την κατασκευή, την επεξεργασία, την αποθήκευση και την ανάλυση δικτύων.

Η κατασκευή ενός δικτύου με το NetworkX γίνεται με την χρήση των παρακάτων συναρτήσεων:

`G = networkx.Graph()`: Δημιουργεί ένα κενό γράφο G

`G.add_node(u)`: Προσθέτει τον κόμβο u στο G

`G.add_edge(u, u')`: Προσθέτει μια ακμή που ενώνει τους κόμβους u, u'

Για την ανάλυση που θέλουμε να πραγματοποιήσουμε μας ενδιαφέρουν δύο κατηγορίες δικτύων:

- Το δίκτυο όπου οι κόμβοι είναι λογαριασμοί χρηστών και μια ακμή συνδέει δύο κόμβους (u, u') αν ο κόμβος u ακολουθεί τον κόμβο u'.
- το δίκτυο όπου οι κόμβοι είναι λογαριασμοί χρηστών και μια ακμή συνδέει δύο κόμβους (u, u') αν ο κόμβος u έχει κάνει retweet ένα tweet του κόμβου u'.

Για να κατασκευάσουμε ένα δίκτυο που έχει ως κόμβους λογαριασμούς χρηστών και ως ακμές το αν ένας χρήστης ακολουθεί έναν άλλο θα χρησιμοποιήσουμε τους πίνακες twitter_user και followship από την βάση δεδομένων. Όπως αναφέραμε στο κεφάλαιο 2.1 ο πίνακας twitter_user περιέχει τους λογαριασμούς χρηστών, ενώ ο πίνακας followship ζευγάρια από id χρηστών όπου ο πρώτος ακολουθεί τον δεύτερο. Άρα κάνοντας χρήση του networkX κατσκευάζουμε ένα γράφημα όπου για κάθε εγγραφή από τον πίνακα twitter_user παίρνουμε το πεδίο id_str και το προσθέτουμε στο γράφημα ως κόμβο και για κάθε εγγραφή από τον πίνακα followship παίρνουμε τα πεδία follower_id, followee_id και τα προσθέτουμε στο γράφημα σαν ακμή.

Αντίστοιχα για να κατασκευάσουμε ένα δίκτυο που έχει ως κόμβους λογαριασμούς χρηστών και ως ακμές αν ένας χρήστης έχει κάνει retweet το tweet ενός άλλου χρήστη μπορούμε να χρησιμοποιήσουμε τους πίνακες retweet, tweet και twitter_user. Ο πίνακας αποτελείται από τα πεδία tweet_id, original_tweet_id που σημαίνει ότι αν έχουμε την εγγραφή (a,b) τότε το tweet που έχει id την τιμή a αποτελεί retweet του tweet που έχει την τιμή b. Σημαντική παρατήρηση είναι ότι για τον πίνακα retweet αντιμετοπίζουμε ακόμα το πρόβλημα που αναφέραμε στην ενότητα 2.4.2, δηλαδή ότι μπορεί να υπάρχουν εγγραφές όπου το πεδίο original_tweet_id να ανήκει σε tweet που έχει δημοσιευθεί από χρήστη που δεν έχουμε συλλέξει. Για να επιλύσουμε το παραπάνω πρόβλημα κάνουμε join τους πίνακες retweet και tweet με `retweet.original_tweet_id = tweet.id_str`, κρατώντας έτσι μόνο εγγραφές του retweet όπου το πεδίο orginal_tweet_id εμφανίζεται και στον πίνακα tweet. Παρόλα αυτά για να κατασκευάσουμε το γράφημα χρειαζόμαστε τους χρήστες στους οποίους ανήκουν οι τιμές των πεδίων tweet_id, original_tweet_id. Αυτή την πληροφορία μπορούμε να την πάρουμε από το join που αναφέραμε παραπάνω. Κάνοντας join τους πίνακες retweet και tweet με `retweet.original_tweet_id = tweet.id_str` κρατάμε τα πεδία:

tweet_id, original_tweet_id, original_tweet_user_id

Για να βρούμε σε ποιους χρήστες ανήκουν οι τιμές του πεδίου tweet_id κάνουμε αντίστοιχα άλλο ένα join πάλι με τον πίνακα tweet με `retweet(tweet_id = tweet.id_str)` και έτσι έχουμε τα πεδία:

tweet_id, tweet_user_id, original_tweet_id, original_tweet_user_id

Έχοντας αυτή την πληροφορία μπορούμε να κατασκευάσουμε με το networkX τον retweet γράφο προσθέτοντας ως κόμβους τα id των λογαριασμών χρηστών που περιέχονται στα πεδία tweet_user_id και original_tweet_user_id και ως ακμές τα ζευγάρια (tweet_user_id, original_tweet_user_id) που εμφανίζονται στις εγγραφές που βρήκαμε παραπάνω.

Παρατηρούμε έτσι ότι μπορούμε να κατασκευάσουμε οποιοδήποτε παραλλαγή ένος δικτύου θέλουμε κάνοντας το κατάλληλο query στην βάση δεδομένων. Για παράδειγμα για να δημιουργήσουμε τον retweet γράφο που αφορά μόνο χρήστες γύρω από το λογαριασμό του @syriza_gr τότε αρκεί να ακολουθήσουμε την διαδικασία που περιγράψαμε επιλέγοντας όμως μόνο τις εγγραφές όπου το πεδίο graph_name είναι ίσο με syriza_gr.

Τέλος, το networkX μπορεί να αποθηκεύσει οποιδήποτε γράφημα στο δίσκο καλώντας την συνάρτηση networkx.write_gefx(output_file_name). Το αρχείο που παράγεται έχει την κατάληξη .gefx και μπορούμε να το φορτώσουμε στο εργαλείο που χρησιμοποιούμε για την οπτικοποίηση των δικτύων.

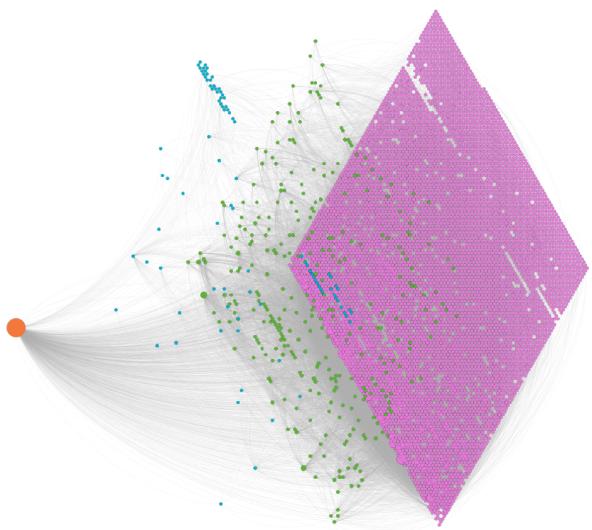
3.3 Οπτικοποίηση δικτύων

Για την οπτικοποίηση των δικτύων χρησιμοποιήσαμε το εργαλείο Gephi. Το Gephi είναι ένα εργαλείο το οποίο παίρνοντας ως είσοδο ένα αρχείο (.gefx) από κομβούς και ακμές μπορεί να το αναπαραστήσει στο επίπεδο, αντιστοιχίζοντας τους κόμβους σε σημεία στο επίπεδο και τις ακμές σε ευθύγραμμα τμήματα που συνδέουν αυτά τα σημεία. Έτσι μπορούμε να έχουμε μια οπτική αναπαράσταση των δικτύων που κατασκευάζουμε μέσα από το networkX.

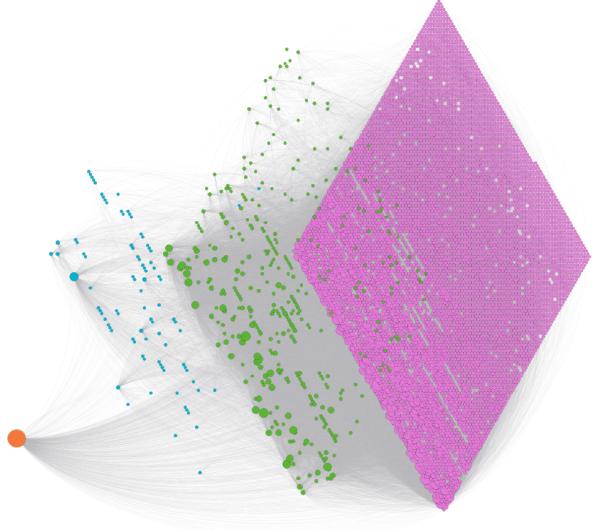
Το Gephi διαθέτει μια πληθώρα από λειτουργίες αλλά άυτες που χρησιμοποιήσαμε περισσότερο είναι:

- Η οργάνωση των κόμβων στο επίπεδο (layout) με συγκεκριμένο τρόπο
- Ο διαχωρισμός (partition) των κόμβων σε ομάδες και ανάθεση ενός χρώματος σε κάθε ομάδα
- Η αυξομείωση του μεγέθους των κόμβων βάση ενός χαρακτηριστικού (πχ. τον αριθμό των εισερχόμενων ακμών που έχει ο κάθε κόμβος)

Παρακάτω βλέπουμε ένα παράδειγμα οπτικοποίησης δύο δικτύων κάνοντας χρήση του Gephi. Η εικόνα 3.1 αντιστοιχεί στο δίκτυο γύρω από το λογαριασμό @neademokratia όπου οι κόμβοι είναι λογαριασμοί χρηστών και οι ακμές αντιστοιχούν στο αν ένας χρήστης ακολουθεί. Αντίστοιχα η εικόνα 3.2 αντιστοιχεί στο δίκτυο γύρω από τον λογαριασμό @syriza_gr.



Εικόνα 3.1. Δίκτυο γύρω από το λογαριασμό @neademokratia



Εικόνα 3.2. Δίκτυο γύρω από το λογαριασμό @syriza_gr

Και στις 2 εικόνες (3.1 , 3.2) για την οργάνωση των κόμβων εφαρμόσαμε το isometric layout. Το συγκεκριμένο layout χωρίζει τους κόμβους σε επίπεδα με βάση μια τιμή z που έχει ο κάθε κόμβος. Στις συγκεκριμένες περιπτώσεις διαλέξαμε ως z τιμή κάθε κόμβου τον αριθμό της επανάληψης που βρισκόταν ο αλγόριθμος συλλογής λογαριασμών χρηστών, όταν σύλλεξε τον συγκεκριμένο κόμβο. Πιο συγκεκριμένα στην εικόνα 3.1, όταν ξεκίνησε ο αλγόριθμος συλλογής χρηστών (1η επανάληψη) είχε αποθηκεύσει μόνο τον αρχικό κόμβο που αντιστοιχεί στο λογαριασμό @neademokratia (πορτοκαλί χρώμα) , στην επόμενη επανάληψη(2) ο αλγόριθμος αποθήκευσε τους χρήστες που έχουν μπλέ χρώμα, ενώ στην 3η και 4η επανάληψη αποθήκευσε τους χρήστες με πράσινο και μωβ χρώμα αντίστοιχα. Τέλος παρατηρούμε ότι οι κόμβοι έχουν διαφορετικό μέγεθος μεταξύ τους, αυτό συμβαίνει διότι επιλέξαμε το μέγεθος του κάθε κόμβου να είναι ανάλογο του in-degree που έχει ο κάθε κόμβος, δηλαδή του αριθμού των εισερχόμενων ακμών.

3.4 Εύρεση και ανάλυση communities του Retweet γράφου

Στην προηγούμενη ενότητα είδαμε πως μπορούμε να κατασκευάσουμε έναν retweet γράφο που περιέχει πληροφορία για το αν κάποιος χρήστης έχει κάνει retweet ένα tweet ενός άλλου χρήστη. Σε αυτή την ενότητα εστιάζουμε στην χρήση αλγορίθμων πάνω σε αυτό το γράφο ώστε να εντοπίσουμε κοινωνίες/ομάδες (communities) στις οποίες χωρίζονται οι κόμβοι (λογαριασμοί χρηστών).

Ως community ορίζουμε ένα σετ από κόμβους οι οποίοι έχουν μεταξύ τους ισχυρές διασυνδέσεις. Έτσι ένας γράφος μπορεί να διαχωριστεί σε communities αν εμφανίζονται κόμβοι που είναι ισχυρά συνδεδεμένοι μεταξύ τους σε σχέση με τους υπόλοιπους κόμβους του δικτύου.

Για τον εντοπισμό των communities σε ένα γράφο θα χρησιμοποιήσουμε την συνάρτηση greedy_modularity_communities() του networkX. Η συνάρτηση υλοποιεί τον αλγόριθμο Clauset-Newman-Moore greedy modularity maximisation για την εύρεση των communities [1].

Ο retweet γράφος που κατασκευάσαμε για αυτή την ανάλυση συμπεριλαμβάνει όλα τα retweets που έχουμε συλλέξει χωρίς να εφαρμόζει κάποιο περιορισμό στην χρονική στιγμή που δημοσιεύθηκε ένα retweet. Παρακάτω παρουσιάζουμε 3 παραλλαγές του παραπάνω γράφου, η πρώτη παραλλαγή είναι ο retweet γράφος που περιέχει χρήστες μέσα από ολόκληρο το σύνολο χρηστών που έχουμε συλλέξει, η δεύτερη παραλλαγή περιέχει μόνο χρήστες γύρω από το λογαριασμό neademokratia και η τρίτη παραλλαγή περιέχει μόνο χρήστες γύρω από το λογαριασμό syriza_gr.

Το πρώτο βήμα για την ανάλυση των τριών γραφών είναι οι εύρεση των communities σε κάθε γράφο ξεχωριστά. Τα αποτελέσματα από την εκτέλεση του αλγορίθμου φαίνονται στον πίνακα 3.2.

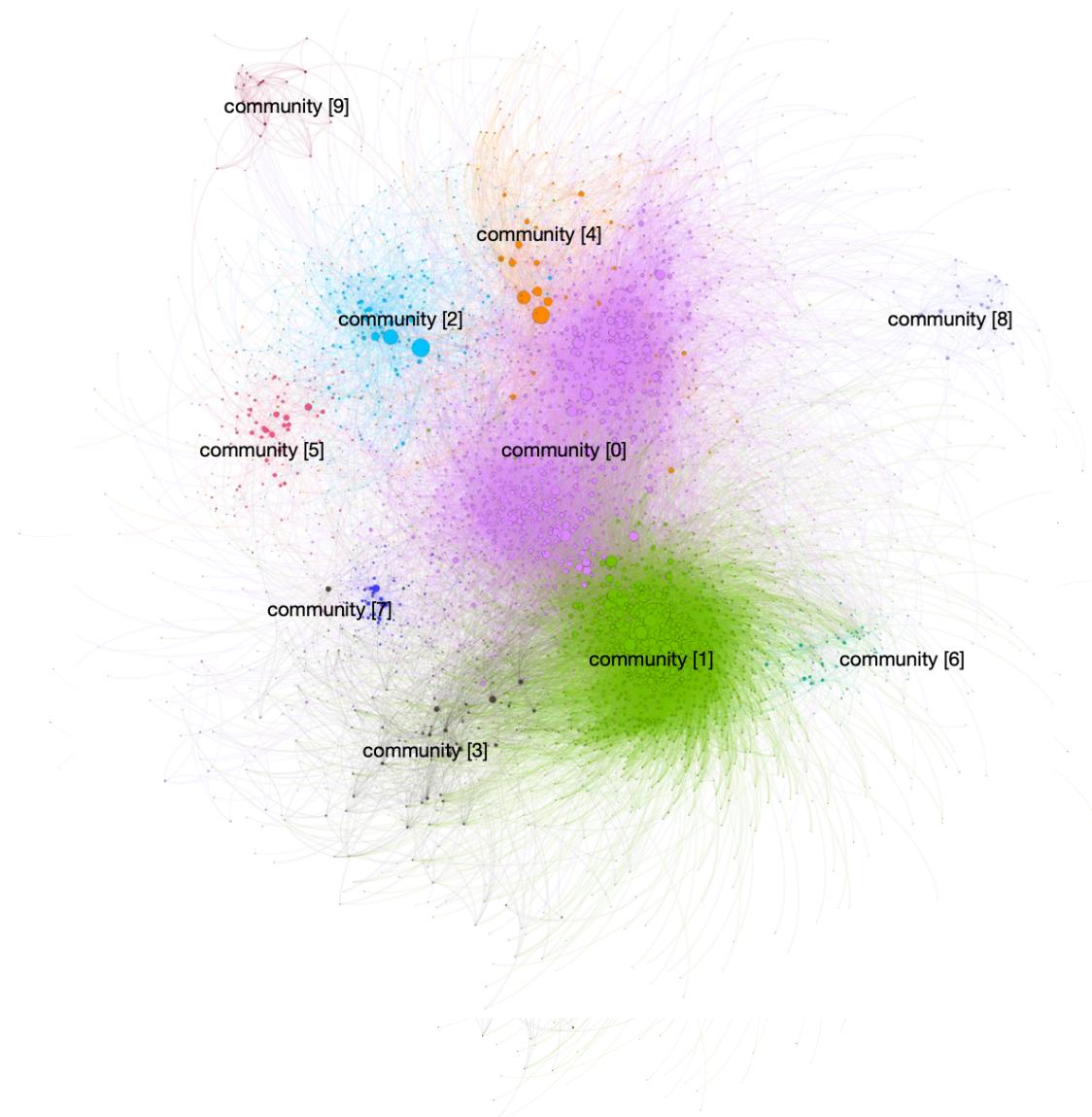
	Κόμβοι	Ακμές	Communities
Όλοι χρήστες	6.150	65.687	813
neademokratia	3.763	14.522	119
syriza_gr	4.748	51.165	811

Πίνακας 3.2.

Παρατηρούμε ότι ο αριθμός των communities σε κάθε γράφο είναι αρκετά μεγάλος. Αυτό συμβαίνει διότι υπάρχουν πολλά communities που αποτελούνται μόνο από έναν λογαριασμό χρήστη επειδή αυτός ο λογαριασμός έχει κάνει retweet μόνο ένα δικό του tweet. Άρα υπάρχουν μερικοί κόμβοι μέσα σε κάθε γράφο που έχουν μόνο μια ακμή προς τον εαυτό τους και αποτελούν από μόνοι τους ένα community. Στην δικιά μας μελέτη θα εστιάσουμε στα δέκα μεγαλύτερα communities σε κάθε γράφο.

Αφού έχουμε εντοπίσει τα communities θέλουμε να δούμε σε κάθε community ποιό είναι το hashtag που έχουν χρησιμοποιήσει οι χρήστες περισσότερο στα retweet τους. Έχοντας τους χρήστες που ανήκουν σε κάθε community και τα retweets που έχουν κάνει, μπορούμε να πάρουμε από την βάση τα hashtags που περιέχονται σε αυτά τα retweets. Φτιάχνουμε έτσι ένα hash map όπου έχουμε ως κλειδία τα id των communities και ως τιμή σε κάθε κλειδί μια λίστα με τα hashtags που έχουν χρησιμοποιηθεί σε αυτό το community. Με βάση αυτό το hash map μπορούμε να υπολογίσουμε ποιο hashtag εμφανίζεται περισσότερες φορές σε κάθε community. Τέλος σε κάθε γράφο τρέχουμε τον αλγόριθμο page rank για να βρούμε τους σημαντικότερους κόμβους του δικτύου και ανζομειώνουμε το μέγεθος του κάθε κόμβου βάση της τιμής page rank που έχει.

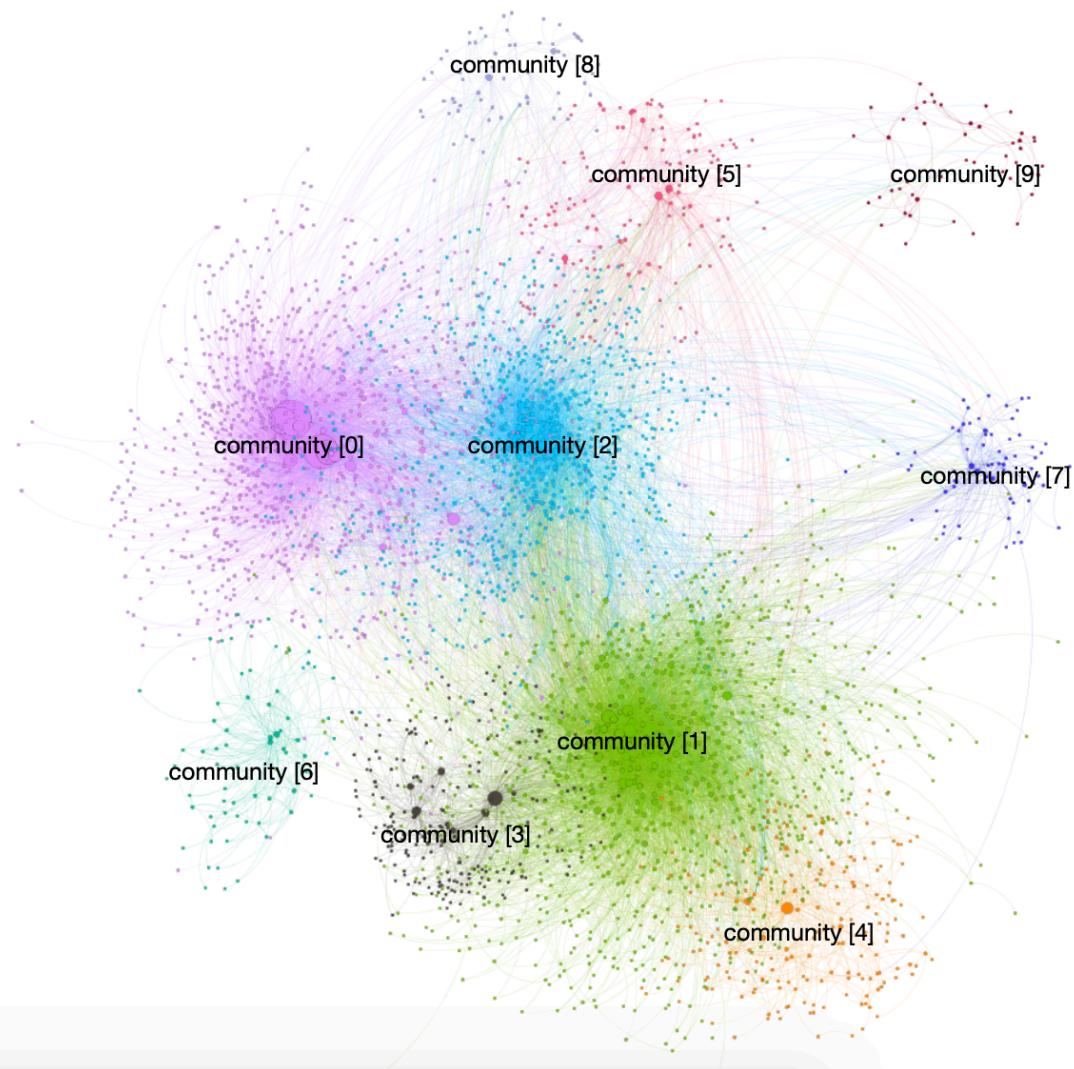
Τα αποτελέσματα από την παραπάνω ανάλυση φαίνονται στις εικόνες 3.3.1 , 3.4.1 , 3.5.1



Εικόνα 3.3.1 Δίκτυο retweet γύρω από τον λογαριασμό syriza_gr

	Most Used hashtag	Σύνολο κόμβων
Community [0]	MasterChefGR	1593
Community [1]	ΝΔ_ξεφτιλες	1064
Community [2]	Blog1600Penn	313
Community [3]	ΣΥΡΙΖΑ_ξεφτιλες	167
Community [4]	onnedtalks	150
Community [5]	TeamGabs	114
Community [6]	Cyprus	94
Community [7]	China	94
Community [8]	ελλήνωνσυνέλευσις	29
Community [9]	AkinAkınözü	25

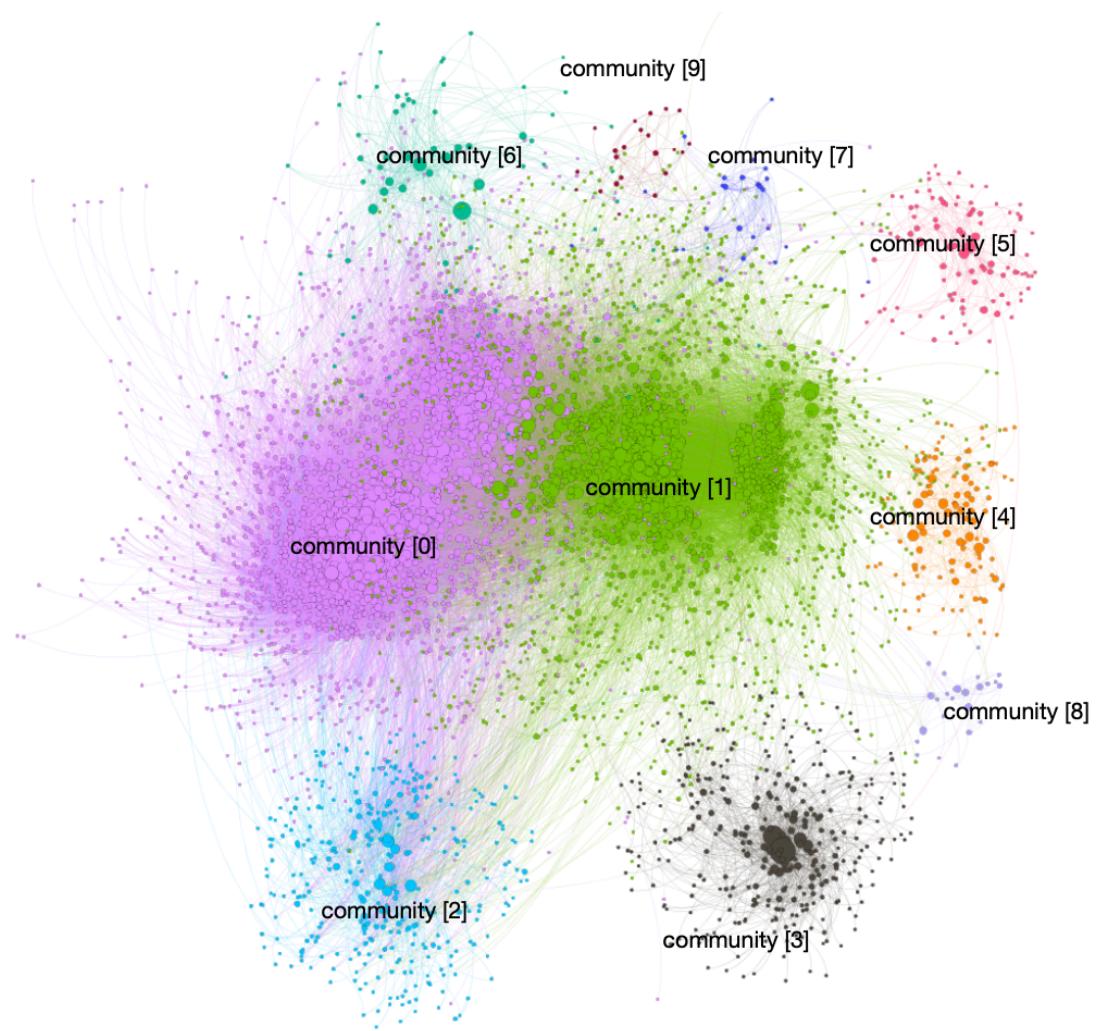
Πίνακας 3.3.2 Most used hashtags για το δίκτυο retweet γύρω από τον λογαριασμό syriza_gr



Εικόνα 3.4.1 Δίκτυο retweet γύρω από τον λογαριασμό neademokratia

	Most Used hashtag	Σύνολο κόμβων
Community [0]	chalkidiki	928
Community [1]	IStandwithGreece	922
Community [2]	MasterChefGR	739
Community [3]	onnedtalks	219
Community [4]	θεος	211
Community [5]	kinima_allagis	151
Community [6]	αρτεμησσωρας	69
Community [7]	olympiacosbc	69
Community [8]	chalkidiki	63
Community [9]	survivorgr	48

Πίνακας 3.4.2 Most used hashtags για το δίκτυο retweet γύρω από τον λογαριασμό neademokratia



Εικόνα 3.5.1 Δίκτυο retweet γύρω και από τους δύο λογαριασμούς

	Most Used hashtag	Σύνολο κόμβων
Community [0]	NΔ_ξεφτιλες	2036
Community [1]	onnedtalks	1924
Community [2]	topotami	337
Community [3]	Blog1600Penn	323
Community [4]	TeamGabs	114
Community [5]	China	102
Community [6]	αρτεμησσωρας	74
Community [7]	AkinAkinözü	25
Community [8]	Cyprus	24
Community [9]	Limassol	22

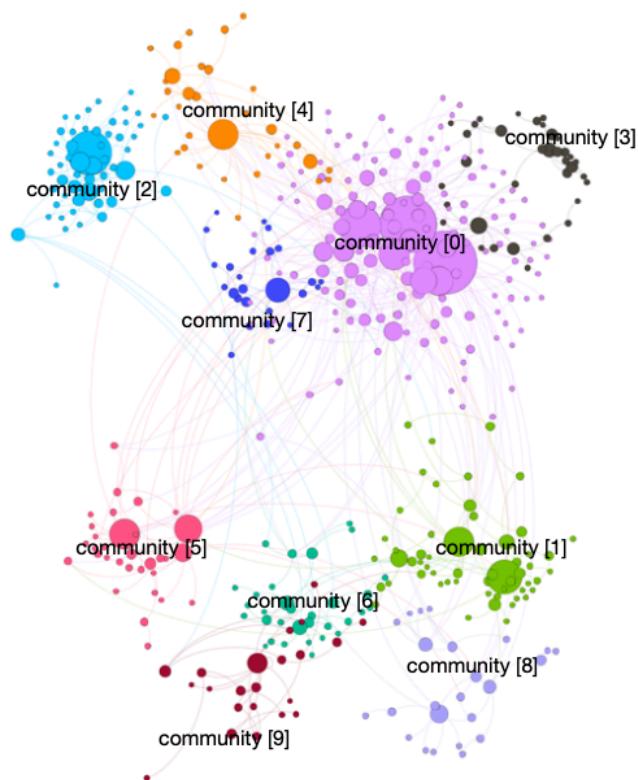
Πίνακας 3.5.2 Most used hashtags για το δίκτυο retweet γύρω και από τους δύο λογαριασμούς

3.5 Μελέτη communities στον χρόνο

Ο τρόπος με τον οποίο πραγματοποιήθηκε η παρακάτω ανάλυση είναι ίδιος με αυτόν στην Ενότητα 3.4. Πιο συγκεκριμένα εφαρμόζουμε την ανάλυση των communities που χρησιμοποιήσαμε στην Ενότητα 3.4 μια φορά για κάθε διαφορετική χρονική περίοδο. Τα αποτλέσματα φαίνονται παρακάτω:

Ιανουάριος 2020

syriza_gr



Εικόνα 3.6. Δίκτυο retweet γύρω από το λογαριασμό *syriza_gr* για τον μήνα Ιανουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	κυβέρνηση_απάτη	148
Community [1]	Syriza_TV	58
Community [2]	XA	54
Community [3]	Cyprus	35
Community [4]	Χίλη	32
Community [5]	Κουλης	32
Community [6]	EastMed	32
Community [7]	Μητσοτακης	25
Community [8]	crete	24
Community [9]	ΣΥΡΙΖΑ_ξεφτιλες	22

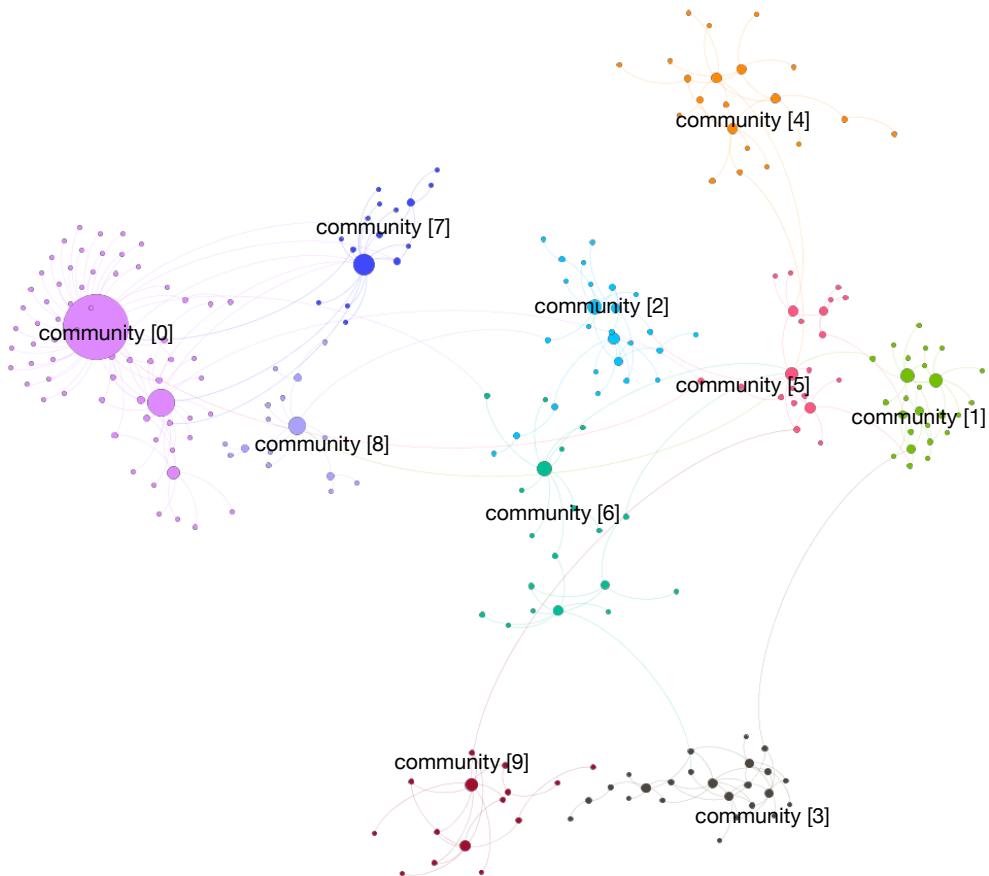
Πίνακας 3.6.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.6

Όνομα Λογαριασμού	Community
yianderm	0
penelceram	0
michaloliakosn	2
NAspetos	0
Wyrlpx99hhKzSky	1
NtinaMpatzia	0
masterchrp	4
BrazucaWhore	5
imatinib100mg	1
Dikefa_Litsa	0

Πίνακας 3.6.2. Κατάταξη των πιο σημαντικών hashtags στο δίκτυο της Εικόνας 3.6 και του community στο οποίο ανήκει ο καθένας

Ιανουάριος 2020

neademokratia



Εικόνα 3.7. Δίκτυο retweet γύρω από το λογαριασμό neademokratia για τον μήνα Ιανουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	EnwpiosEnopiw	67
Community [1]	CeJourLa	27
Community [2]	cyprus	26
Community [3]	olympiaekosbc	24
Community [4]	ΝΔ_ξεφτιλες	21
Community [5]	Εστία	21
Community [6]	κυβερνηση_τσιρκο	19
Community [7]	Ζαγορακης	16
Community [8]	bestfriendsfear	16
Community [9]	kinima_allagis	15

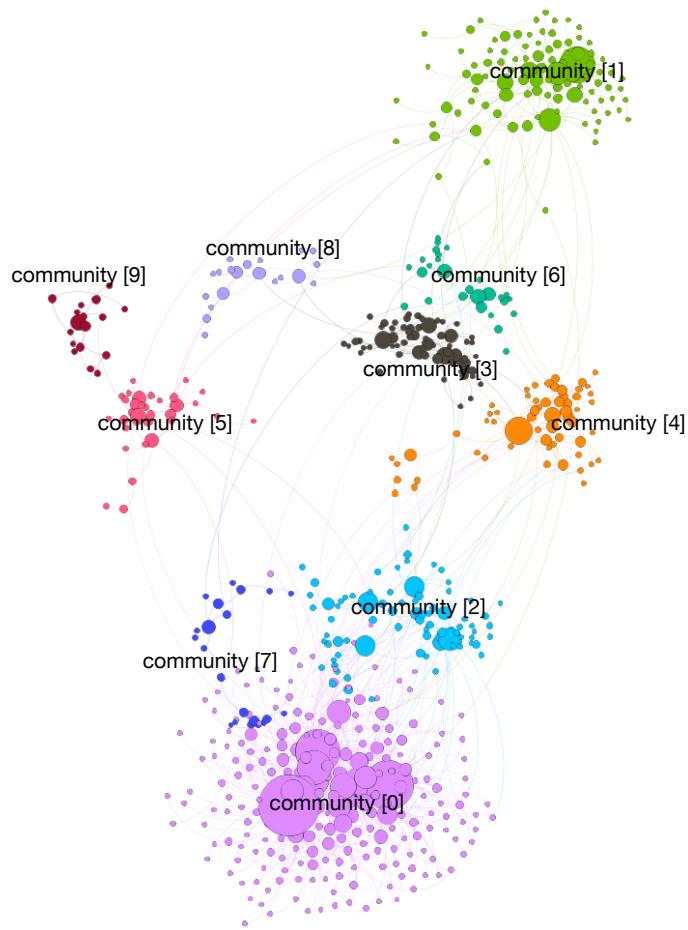
Πίνακας 3.7.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.7

Όνομα Λογαριασμού	Community
NikosKotzias	0
kostasbarkas	0
olgagerovasili	7
blackfe_	8
CPanoudis	6
loucas_fourlas	2
	1
KyvernitiParat1	1
christidisP	9
Cocodi01891627	5

Πίνακας 3.7.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.7 και του community στο οποίο ανήκει ο καθένας

Ιανουάριος 2020

Και για τους δύο λογαριασμούς



Εικόνα 3.8. Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Ιανουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	κυβέρνηση_απάτη	230
Community [1]	XA	99
Community [2]	AgriesMelisses	80
Community [3]	kinima_allagis	65
Community [4]	ΠΑΟΚ	59
Community [5]	θεος	33
Community [6]	Cyprus	27
Community [7]	νεοκυμα	21
Community [8]	EastMed	20
Community [9]	AgriesMelisses	17

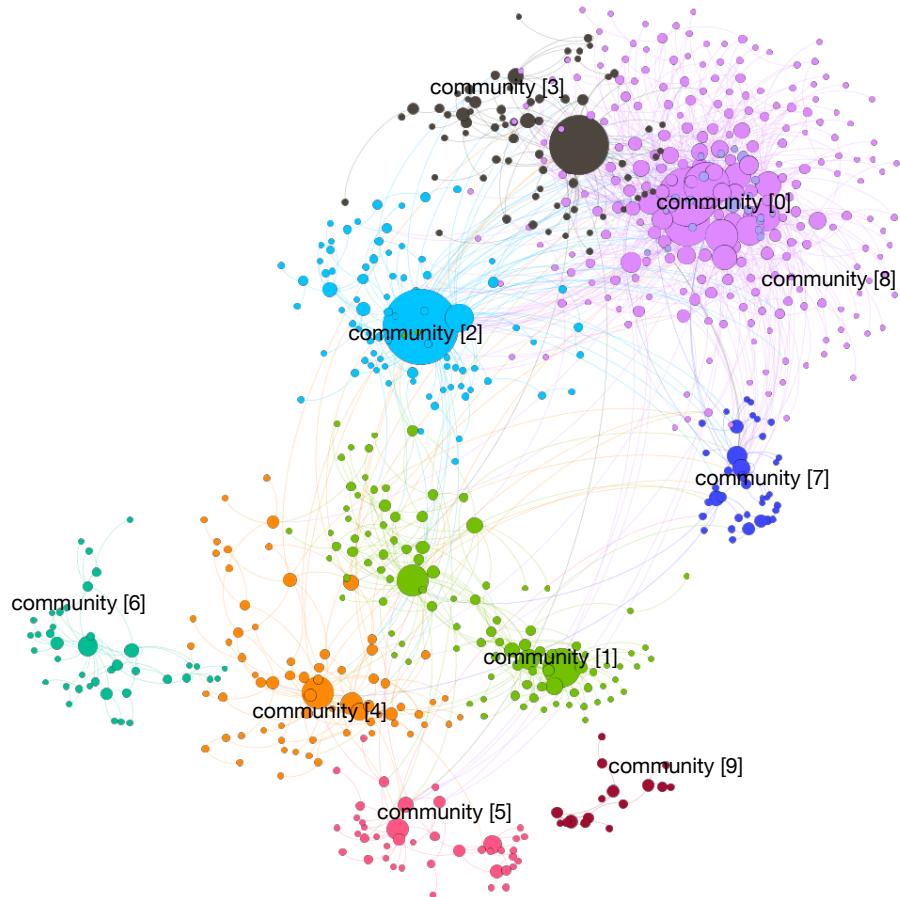
Πίνακας 3.8.1. Πίνακας με τα πιο χρησιμοποιημένα hashtag σε κάθε community του δικτύου της Εικόνας 3.8

Όνομα Λογαριασμού	Community
NikosKotzias	0
yianderm	0
penelcecream	0
michaloliakosn	1
NAspetos	0
Wyrlpx99hhKzSky	4
NtinaMpatzia	0
kostasbarkas	0
masterchrp	0
imatinib100mg	0

Πίνακας 3.8.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.8 και του community στο οποίο ανήκει ο καθένας

Φεβρουάριος 2020

syriza_gr



Εικόνα 3.9. Δίκτυο retweet γύρω από το λογαριασμό syriza_gr για τον μήνα Φεβρουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	7minesND	243
Community [1]	XA	104
Community [2]	MEGA	82
Community [3]	petralona	74
Community [4]	ARSOLY	69
Community [5]	MasterChefGR	42
Community [6]	Thueringen	41
Community [7]	mega	38
Community [8]	Χιος	22
Community [9]	China	16

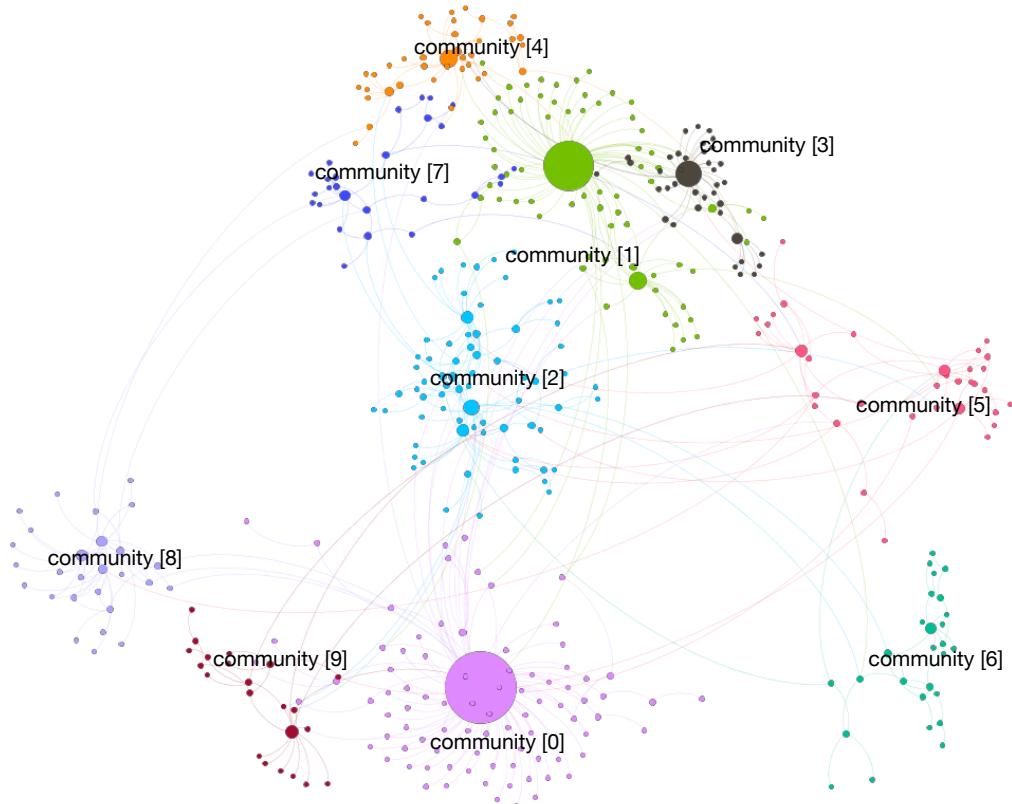
Πίνακας 3.9.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.9

Όνομα Λογαριασμού	Community
AndroniEv	2
815_1979	3
penelceram	0
yianderm	0
AlexandrosNikos	0
michaloliakosn	1
Ymbjmrf	0
Wyrlpx99hhKzSky	4
Epicrotus	1
NAspetos	0

Πίνακας 3.9.2. Κατάταξη των πιο σημαντικών hashtags στο δίκτυο της Εικόνας 4.4 και του community στο οποίο ανήκει ο καθένας

Φεβρουάριος 2020

neademokratia



Εικόνα 3.10. Δίκτυο retweet γύρω από το λογαριασμό neademokratia για τον μήνα Φεβρουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	chalkidiki	87
Community [1]	chalkidiki	70
Community [2]	chalkidiki	70
Community [3]	chalkidiki	39
Community [4]	ΣΥΡΙΖΑ_ξεφτίλες	38
Community [5]	IStandWithGreece	34
Community [6]	Cyprus	27
Community [7]	AgriesMelisses	27
Community [8]	ARSOLY	27
Community [9]	chalkidiki	22

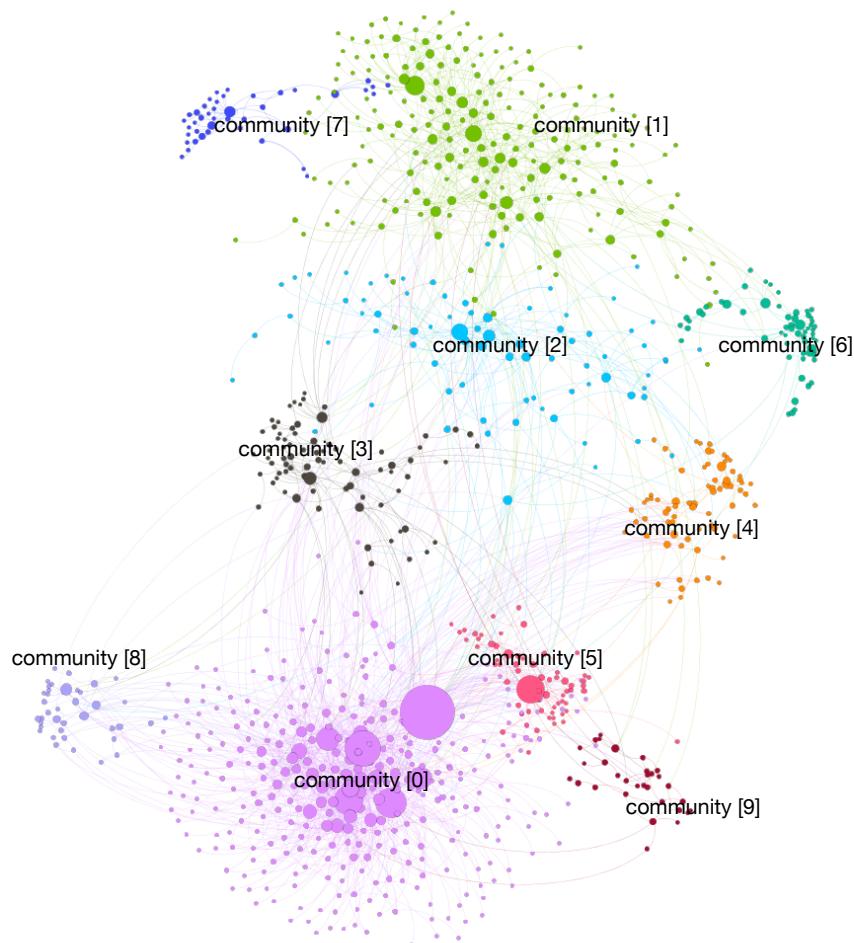
Πίνακας 3.10.1. Πίνακας με τα πιο χρησιμοποιημένα hashtag σε κάθε community του δικτύου της Εικόνας 3.10

Όνομα Λογαριασμού	Community
igissippi	0
NikosKotzias	1
kostasbarkas	3
aaterammos	1
christidisp	4
THEOCHAROUSE	2
KyvernitiParat1	9
PGfgjyMeNtf7L2N	10
GreColombia	2
MitiliniEllhnas	2

Πίνακας 3.10.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.10 και του community στο οποίο ανήκει ο καθένας

Φεβρουάριος 2020

Και για τους δύο λογαριασμούς



Εικόνα 3.11. Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Φεβρουάριο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	chalkidiki	329
Community [1]	XA	204
Community [2]	AgriesMelisses	78
Community [3]	chalkidiki	78
Community [4]	MasterChefGR	71
Community [5]	chalkidiki	64
Community [6]	Cyprus	61
Community [7]	Thueringen	42
Community [8]	ΝΔ_ξεφτιλες	40
Community [9]	raoraok	31

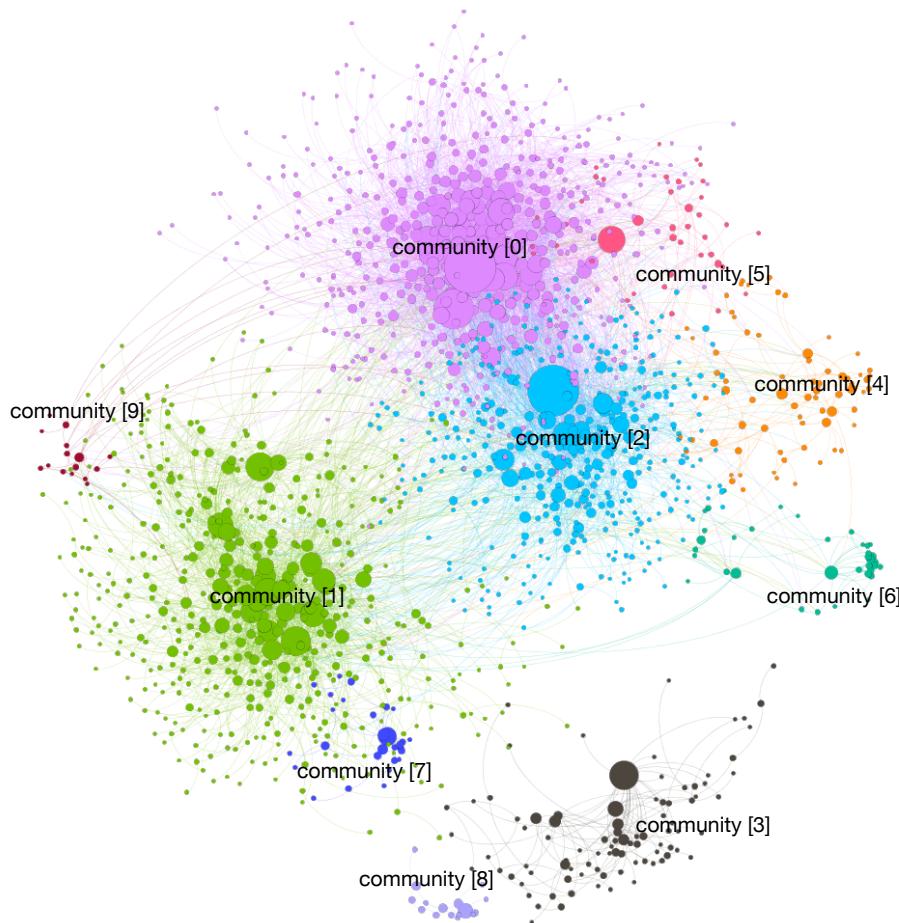
Πίνακας 3.11.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.11

Όνομα Λογαριασμού	Community
igissipos	0
AndroniEv	0
NikosKotzias	0
penelceram	0
815_1979	5
yianderm	0
AlexandrosNik	0
michaloliakosn	1
Wyrpx99hhKzSky	2
Ymbjmrf	0

Πίνακας 3.11.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.11 και του community στο οποίο ανήκει ο καθένας

Μάρτιος 2020

syriza_gr



Εικόνα 3.12. Δίκτυο retweet γύρω από το λογαριασμό syriza_gr για τον μήνα Μάρτιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	κυβερνηση_τσίρκο	511
Community [1]	XA	460
Community [2]	καραντίνα	377
Community [3]	Thüringen	90
Community [4]	Cyprus	71
Community [5]	COVID—19	42
Community [6]	Hercai	36
Community [7]	China	28
Community [8]	ForaPombo	17
Community [9]	κορωνοϊος	15

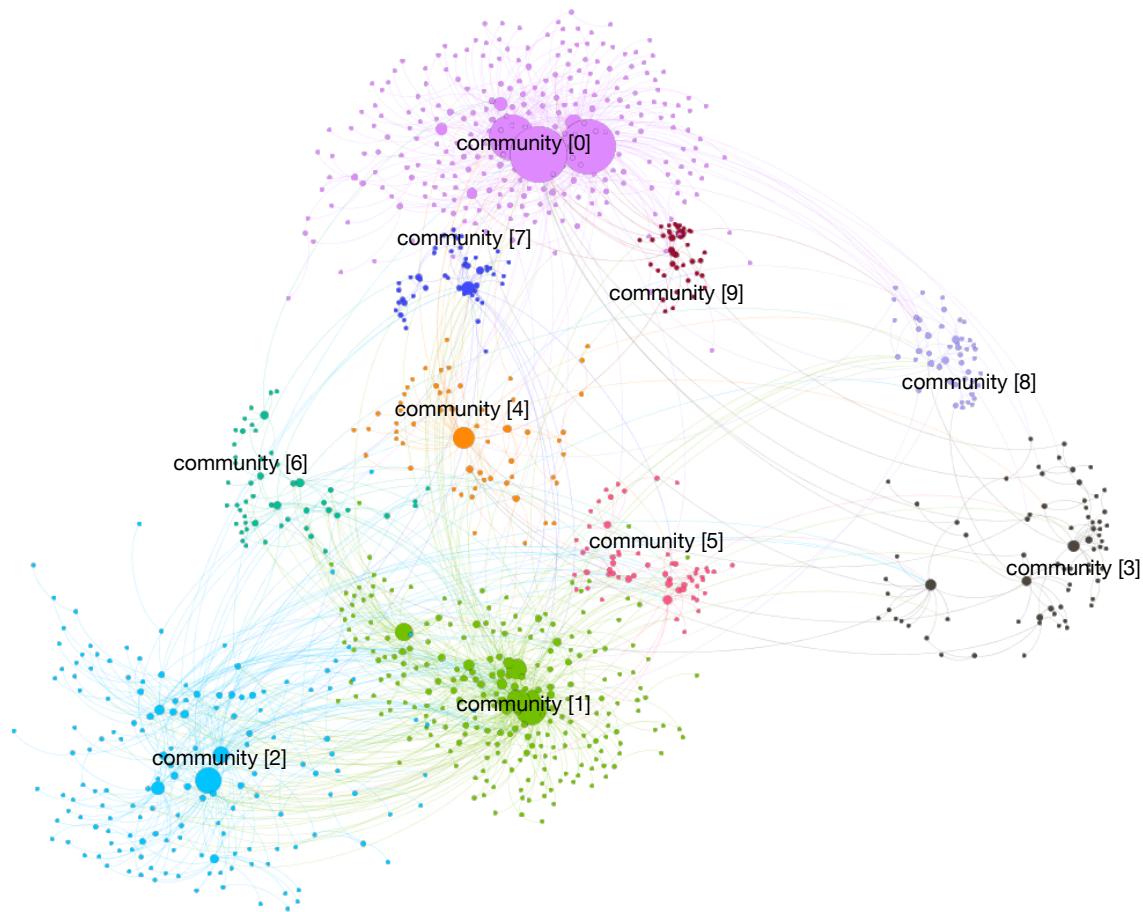
Πίνακας 3.12.1. Πίνακας με τα πιο χρησιμοποιημένο hashtag σε κάθε community του δικτύου της Εικόνας 3.12

Όνομα Λογαριασμού	Community
Blonde070121	0
pepe_g12	2
yianderm	0
Anarxos_2nd	0
GTrapeziotis	0
NtinaMpatzia	0
dimitrafotiadou	0
masterchrp	0
staposto	1
gavaises	2

Πίνακας 3.12.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.12 και του community στο οποίο ανήκει ο καθένας

Μάρτιος 2020

neademokratia



Εικόνα 3.13. Δίκτυο retweet γύρω από το λογαριασμό *syriza_gr* για τον μήνα Μάρτιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	vouli	277
Community [1]	IStandWithGreece	232
Community [2]	IStandWithGreece	157
Community [3]	AgriesMelisses	71
Community [4]	σεισμος	57
Community [5]	Evros	57
Community [6]	EOKA	51
Community [7]	cyprus	49
Community [8]	μενουμεστοσπιτι	48
Community [9]	κλειστε_τα_σχολεια_τωρα	39

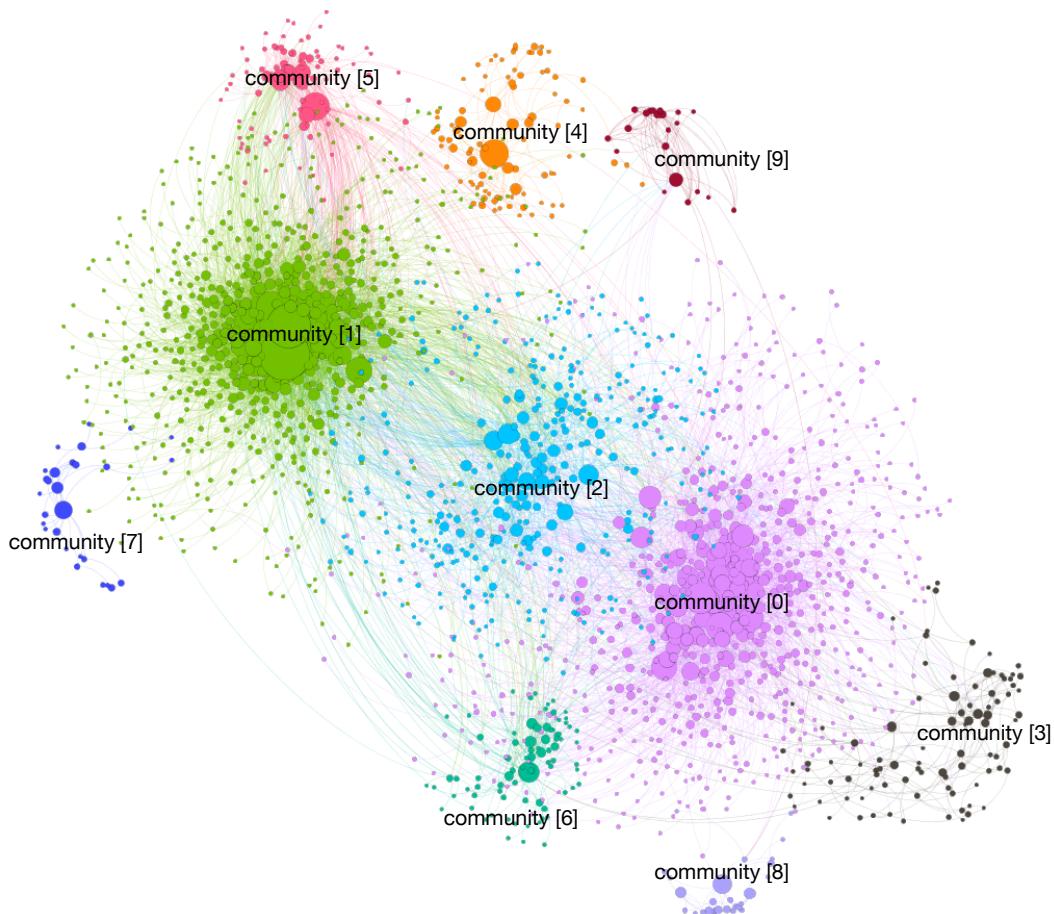
Πίνακας 3.13.1. Πίνακας με τα πιο χρησιμοποιημένα hashtag σε κάθε community του δικτύου της Εικόνας 3.13

Όνομα Λογαριασμού	Community
ellinaspiratis	0
NikosKotzias	0
kostasbarkas	0
HelenaDaZeus	1
Kyvernitiparat1	2
XenakisSotiris	1
Dachtus	4
THEOCHAROUSE	1
ipanagiotaros	1
HMetanastria	2

Πίνακας 3.13.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.13 και του community στο οποίο ανήκει ο καθένας

Μάρτιος

Και για τους δύο λογαριασμούς



Εικόνα 3.14. Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Μάρτιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	IStandWithGreece	694
Community [1]	κυβερνηση_τσιρκο	674
Community [2]	AgriesMelisses	354
Community [3]	Cyprus	92
Community [4]	Thüringen	92
Community [5]	κορονοιος	88
Community [6]	κορωνοιος	82
Community [7]	China	28
Community [8]	ελλήνωνσυνέλευσις	27
Community [9]	Hercai	21

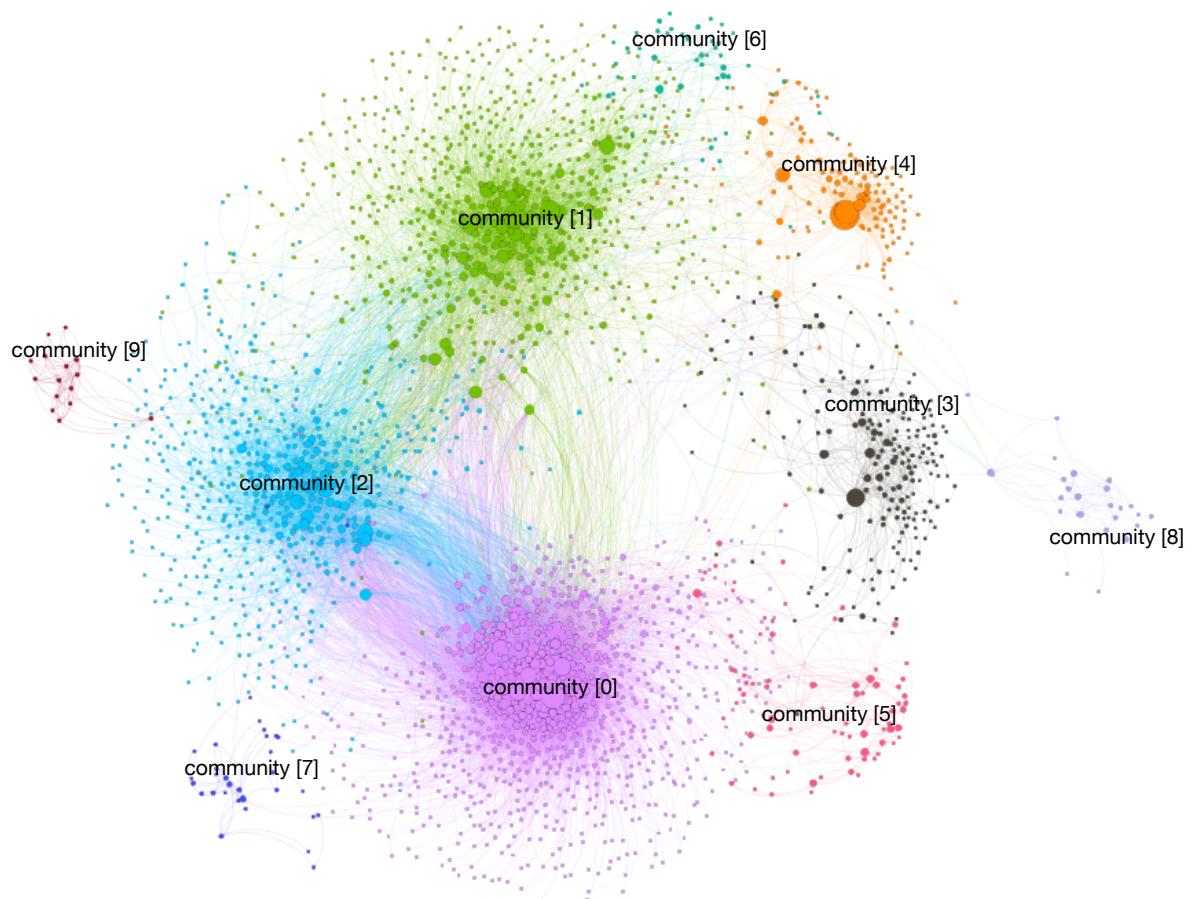
Πίνακας 3.14.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.14

Όνομα Λογαριασμού	Community
Blonde070121	1
pepe_g12	1
yianerm	1
ellinaspiratis	1
NikosKotzias	1
Anarxos_2nd	1
GTrapeziotis	1
HelenaDaZeus	0
kostasbarkas	1
KyvernitiParat1	0

Πίνακας 3.14.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.14 και του community στο οποίο ανήκει ο καθένας

Απρίλιος 2020

syriza_gr



Εικόνα 3.15. Δίκτυο retweet γύρω από το λογαριασμό syriza_gr για τον μήνα Απρίλιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	Σκοιλ_Ελικικου	918
Community [1]	ΝΔ_Ξεφτιλες	684
Community [2]	MasterChefGR	520
Community [3]	Corona	199
Community [4]	onnedtalks	127
Community [5]	TeamGabs	79
Community [6]	Cyprus	51
Community [7]	China	32
Community [8]	ελλήνωνσυνέλευσις	26
Community [9]	AkinAkinözü	15

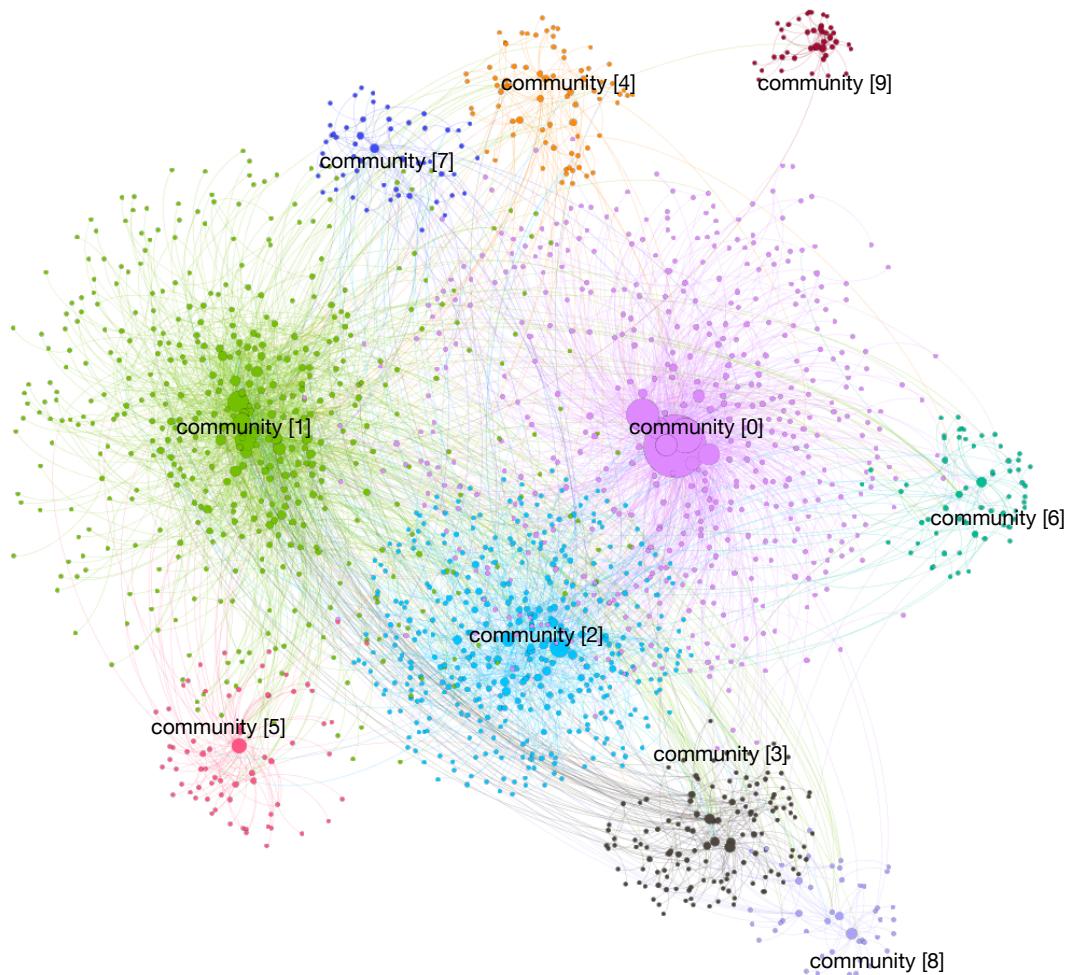
Πίνακας 3.15.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.15

Όνομα Λογαριασμού	Community
Blonde070121	0
karasarininisandr	4
yianderm	0
eirini_chris	0
K_Tsiagklitis	0
giorgosptk	4
LeoKosmas	1
simoritis	0
prodaster	0
athinaxa79	0

Πίνακας 3.15.2. Κατάταξη των πιο σημαντικών hashtags στο δίκτυο της Εικόνας 3.15 και του community στο οποίο ανήκει ο καθένας

Απρίλιος 2020

neademokratia



Εικόνα 3.16. Δίκτυο retweet γύρω από το λογαριασμό neademokratia για τον μήνα Απρίλιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	Σκοιλ_Ελικικου	484
Community [1]	Ρομά	455
Community [2]	radio_arkady	372
Community [3]	onnedtalks	133
Community [4]	Cyprus	75
Community [5]	μενουμε_σπιτι	67
Community [6]	syriza_xeftiles	63
Community [7]	shoppingstar	53
Community [8]	kinima_elpidas	46
Community [9]	αρτέμηςσώρρας	42

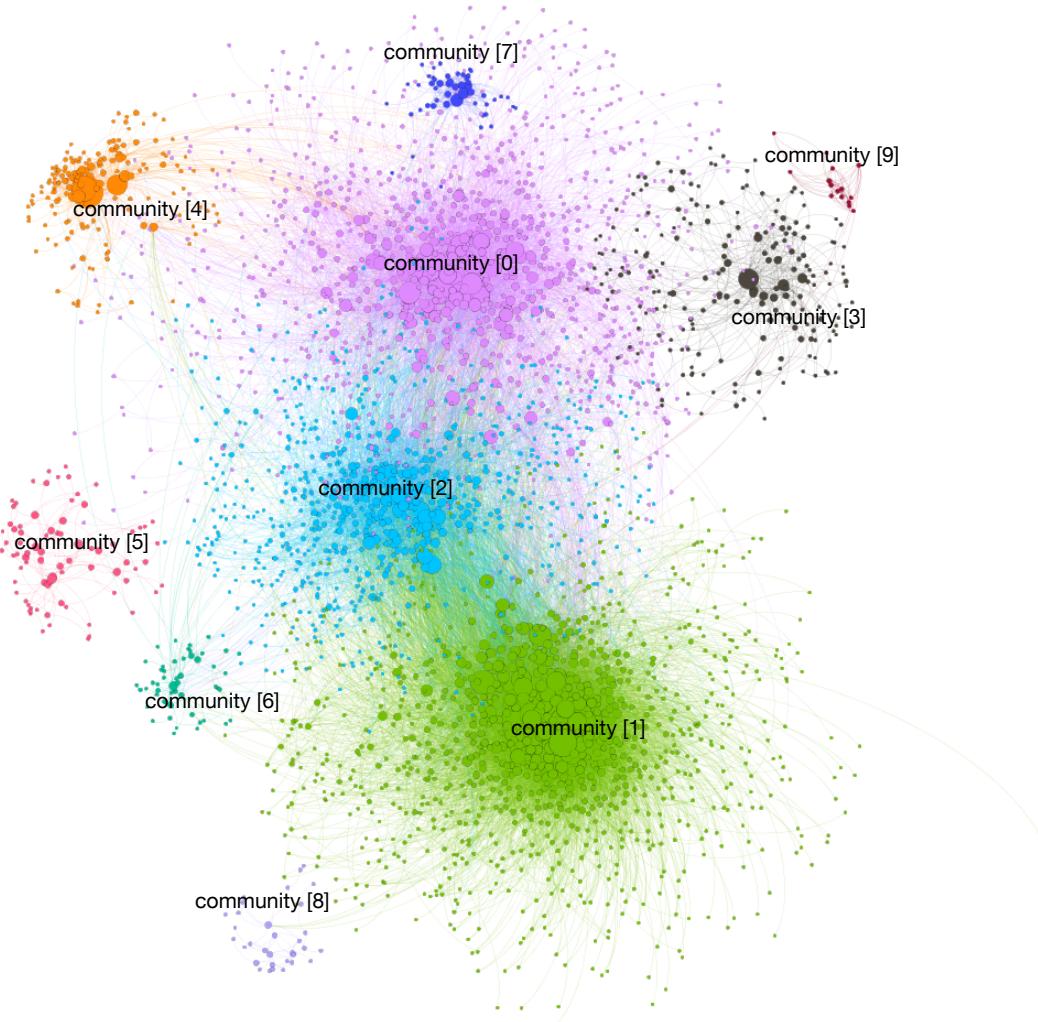
Πίνακας 3.16.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.16

Όνομα Λογαριασμού	Community
ellinaspiratis	0
kostasbarkas	0
NikosKotzias	0
PGfgjyjMeNtf7L2N	0
ax_axristos	0
aaterammos	0
PatisetoKim	1
HelenaDaZeus	1
meli7_meli	2
loukalex	1

Πίνακας 3.16.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.16 και του community στο οποίο ανήκει ο καθένας

Απρίλιος 2020

Και για τους δύο λογαριασμούς



Εικόνα 3.17. Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Απρίλιο

	Most Used hashtag	Σύνολο κόμβων
Community [0]	ΝΔ	1017
Community [1]	Σκοιλ_Ελικικου	1001
Community [2]	MasterChefGR	706
Community [3]	Corona	205
Community [4]	onnectedtalks	164
Community [5]	TeamGabs	79
Community [6]	Cyprus	52
Community [7]	αρτέμηςσώρρας	52
Community [8]	China	32
Community [9]	AkinAkinözü	15

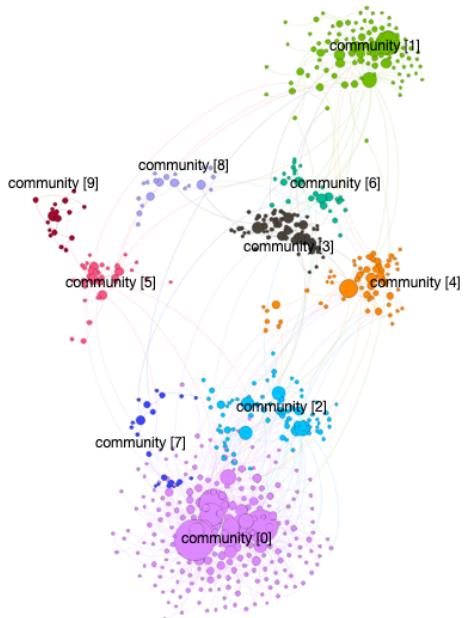
Πίνακας 3.17.1. Πίνακας με τα πιο χρησιμοποιημένα hashtags σε κάθε community του δικτύου της Εικόνας 3.17

Όνομα Λογαριασμού	Community
Blonde070121	1
elinaspiratis	1
karasarinisandr	4
masterchrp	1
eirini_chris	1
yianderm	1
K_Tsagkliotis	4
giorgosptk	0
LeoKosmas	1
simoritis	1

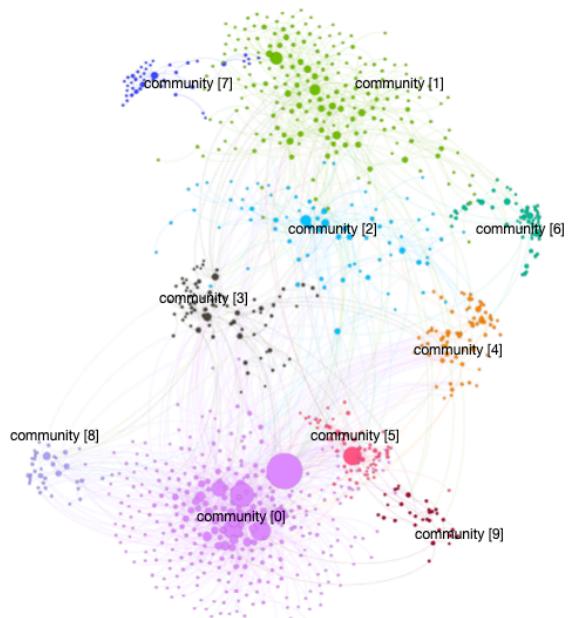
Πίνακας 3.17.2. Κατάταξη των πιο σημαντικών χρηστών στο δίκτυο της Εικόνας 3.17 και του community στο οποίο ανήκει ο καθένας

3.6 Εντοπισμός δυναμικών communities

Στην προηγούμενη ενότητα εντοπίσαμε τα communities στους γράφους που αντιστοιχούν στις χρονικές στιγμές t_0, t_2, \dots, t_4 , όπου η χρονική στιγμή t_0 αντιστοιχεί στον μήνα Ιανουάριο, η χρονική στιγμή t_1 στον μήνα Φεβρουάριο κ.ο.κ. Γνωρίζουμε έτσι, για κάθε μια από τις παραπάνω χρονικές στιγμές τον τρόπο με τον οποίο είναι οργανωμένα τα communities στους γράφους που αντιστοιχούν στον λοαγαριασμό neademokratia, στον λογαριασμό syriza_gr και τέλος και στους δύο λογαριασμούς. Παρόλα αυτά δεν γνωρίζουμε την συσχέτιση που υπάρχει μεταξύ των communities δύο γράφων σε διαδοχικές χρονικές στιγμές. Για παράδειγμα στις εικόνες 3.18.1, 3.18.2 που φαίνονται παρακάτω έχουμε βρεί τα communities για την χρονική στιγμή t_0 αλλά δεν γνωρίζουμε πως αυτά αντιστοιχίζονται με τα communities που έχουμε εντοπίσει την χρονική στιγμή t_1 , δηλαδή το community[0] την χρονική στιγμή t_0 αποτελείται από τους ίδιους κόμβους που αποτελείται και το community[0] την χρονική στιγμή t_1 ;



Εικόνα 3.18.1 Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Ιανουάριο (χρονική στιγμή t_1)



Εικόνα 3.18.2 Δίκτυο retweet γύρω και από τους δύο λογαριασμούς για τον μήνα Φεβρουάριο (χρονική στιγμή t_2)

Για να απαντήσουμε στο παραπάνω ερώτημα δεν θα εντοπίσουμε απλά τα στατικά step communities που εμφανίζονται σε κάθε χρονική στιγμή ξεχωριστά, αλλά θα μελετήσουμε τις μεταβολές που παρουσιάζουν τα communities από την μία χρονική στιγμή στην άλλη. Για να πραγματοποιήσουμε την παραπάνω ανάλυση βασιστήκαμε στην εργασία [3] των Derek Greene, Donal Doyle, Padraig Cunningham οι οποίοι περιέγραψαν έναν αλγόριθμο για τον εντοπισμό δυναμικών communities στον χρόνο.

Όπως αναφέραμε, για να υπολογίσουμε τα δυναμικά communities πρέπει πρώτα να εντοπίσουμε τις μεταβολές που δέχεται κάθε community από την μία χρονική στιγμή στην επόμενη. Παρακάτω

παρουσιάζουμε τις διάφορες κατηγορίες μεταβολών που μπορεί να προκύψουν σε ένα community μεταξύ δύο διαδοχικών χρονικών στιγμών βάση της εργασίας [3].

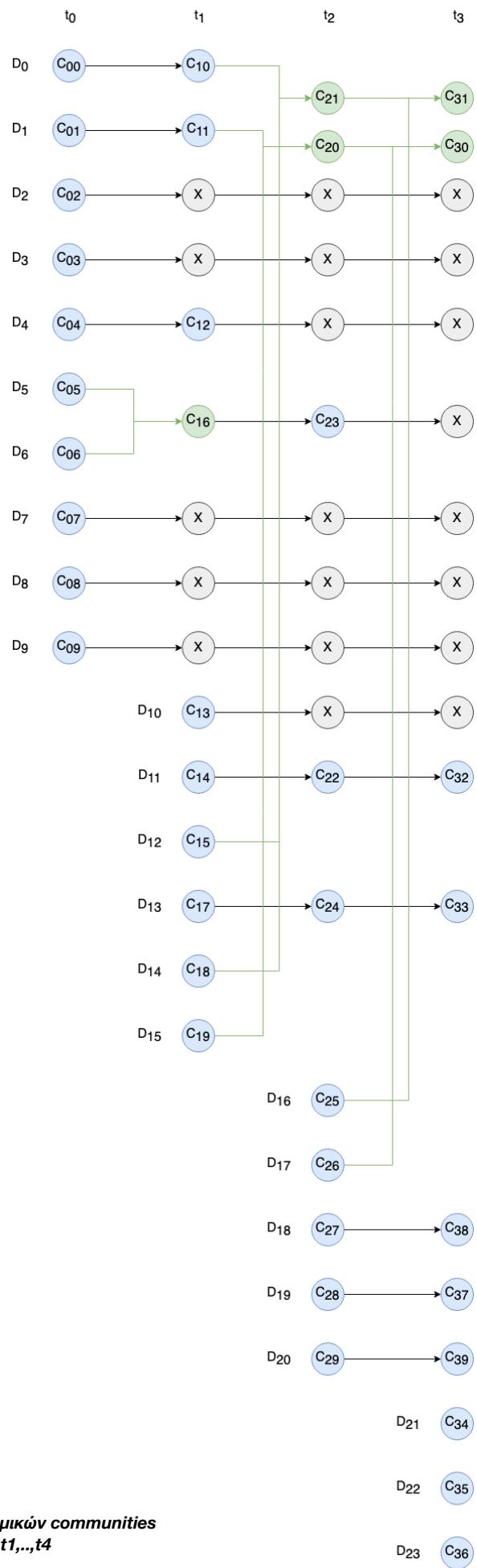
- Birth: Η εμφάνιση ενός step community C_{tj} την χρονική στιγμή t για το οποίο δεν υπήρχε αντίστοιχο δυναμικό community. Δημιουργείται έτσι ένα καινούργιο δυναμικό community που περιέχει το C_{tj} .
- Death: Η εξαφάνιση ενός δυναμικού community συμβαίνει όταν αυτό δεν έχει παρατηρηθεί στο γράφο για δυο συνεχόμενες χρονικές στιγμές.
- Merging: Συμβαίνει όταν δύο ξεχωριστά δυναμικά communities D_i, D_j που παρατηρούνται την χρονική στιγμή $t-1$ συγχωνεύονται σε ένα step community την χρονική στιγμή t .
- Splitting: Συμβαίνει όταν ένα δυναμικό community που παρατηρείται την χρονική στιγμή $t-1$, αντιστοιχίζεται σε δύο διαφορετικά step communities την χρονική στιγμή t .

Έχοντας ορίσει τις κατηγορίες των μεταβολών που μπορεί να προκύψουν, εφαρμόζουμε τον αλγόριθμο που περιγράφεται στο Fig. 4 [3] για την εύρεση των δυναμικών communities. Κεντρική ιδέα του αλγορίθμου είναι ότι για $g1, g2, \dots, gn$ γράφους που αντστοιχούν στις χρονικές στιγμές $t1, t2, \dots, tn$, ξεκινώντας από τον γράφο $g1$ υπολογίζουμε τα στατικά communities με την χρήση οποιουδήποτε αλγόριθμου εντοπίζει communities σε ένα γράφο. Σε κάθε ένα από τα communities που υπολογίζαμε αντστοιχούμε ένα καινούργιο δυναμικό community. Υστερα υπολογίζουμε τα στατικά communities στον γράφο $g2$ και τα συγκρίνουμε με κάθε δυναμικό community που έχουμε μέχρι στιγμής. Η σύσκριση γίνεται με τη χρήση του Jaccard coefficient. Αν η τιμή Jaccard μεταξύ δύο community είναι μεγαλύτερη μιας τιμής θ τότε τα δύο communities ταιριάζουν. Στην περίπτωση που ένα στατικό community ταιριάζει σε ένα δυναμικό community τότε αντιστοιχίζουμε σε αυτό το δυναμικό community το στατικό community. Αν ένα στατικό community δεν ταιριάζει με κανένα δυναμικό community τότε δημιουργούμε ένα καινούργιο δυναμικό community και το αντιστοιχίζουμε με το στατικό. Τέλος, αν ένα δυναμικό community αντιστοιχιστεί σε περισσότερα από ένα στατικά communities τότε υπάρχει splitting, δημιουργούμε έτσι ένα καινούργιο δυναμικό community. Η παραπάνω διαδικασία συνεχίζεται μέχρι να επεξεργαστούν όλοι οι $g1, g2, \dots, gn$ γράφοι.

Ακολουθεί η περιγραφή του αλγόριθμου σε ψευδοκώδικα:

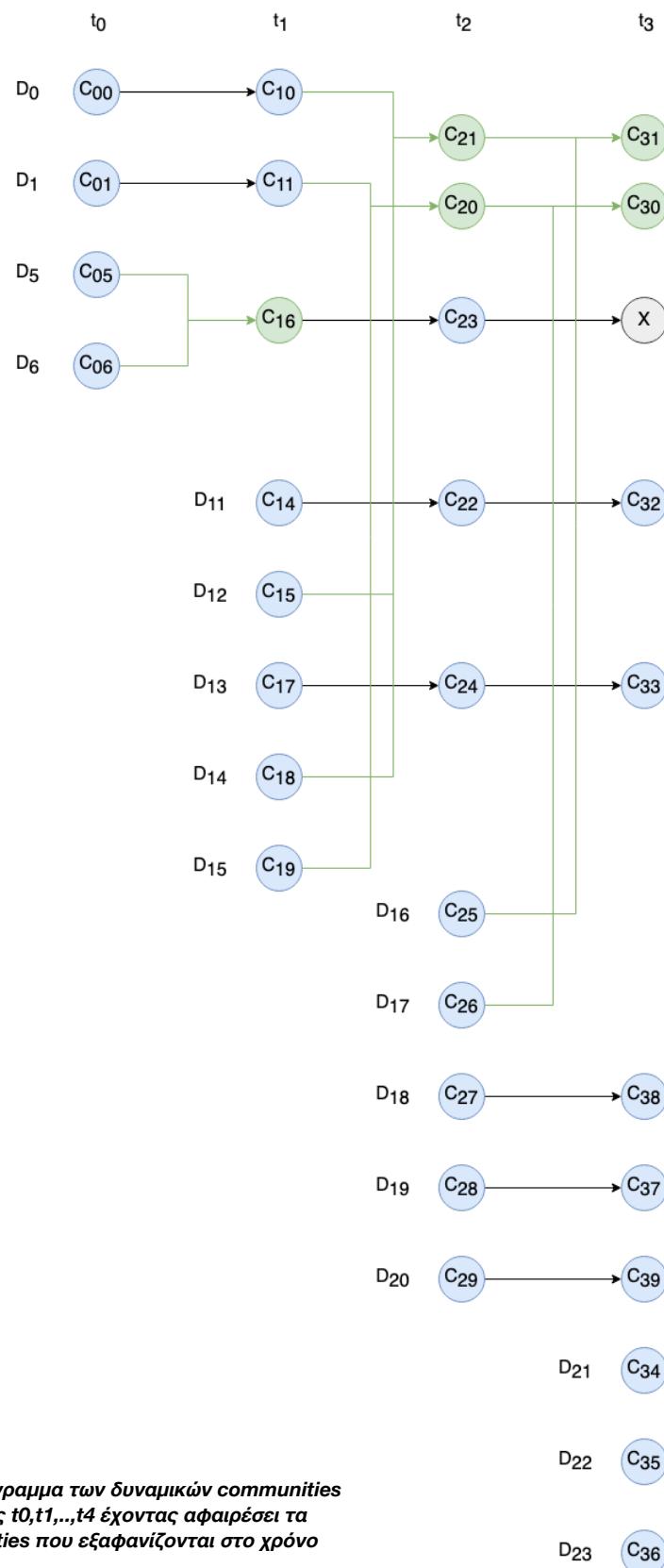
1. Βρες τα communities στον γράφο g_1 και βρες το σύνολο από communities C_1 . Αρχικοποίησε το σύνολο D δημιουργώντας ένα καινούργιο δυναμικό community για κάθε στοιχείο C_{1i} που ανήκει στο C_1 .
2. Για κάθε επόμενο βήμα $t > 1$, βρες το σύνολο από communities C_t από το g_r
3. Για κάθε C_{ta} που ανήκει στο C_t όπως φαίνεται παρακάτω:
 - 3.1. Αντιστοίχισε όλα τα δυναμικά communities D_i για τα οποία $sim(C_{ta}, F_i) > \theta$
 - 3.2. Αν δεν υπάρχει καμία αντιστοίχιση, δημιουργησε ένα καινούργιο δυναμικό community που περιέχει το C_{ta} .
 - 3.3. Άλλιώς πρόσθεσε το C_{ta} σε κάθε δυναμικό community που έχει αντιστοιχιστεί.
4. Ενημέρωσε το σύνολο που κρατάει ποιό είναι το τελευταίο step community που αντιστοιχεί σε κάθε δυναμικό community. Για κάθε περίπτωση όπου ένα υπάρχων δυναμικό community έχει αντιστοιχιστεί σε 2 ή περισσότερα step communities, δημιουργησε ένα καινούργιο split δυναμικό community.
5. Επανέλαβε την παραπάνω διαδικασία από το βήμα 2 μέχρι όλα τα γραφήματα να έχουν προσπελαθεί.

Παρακάτω στην Εικόνα 3.19 φαίνονται τα αποτελέσματα της εκτέλεσης του αλγορίθμου για τους γράφους $g0, g2, …, g3$ που αντιστιχούν στο δίκτυο retweet γύρω και από τους δύο λογαριασμούς (syriza_gr, neademokratia) για τις χρονικές στιγμές t0: Ιανουάριος, t1:Φεβρουάριος, … , t3: Απρίλιος και τιμή $\theta=0.35$.



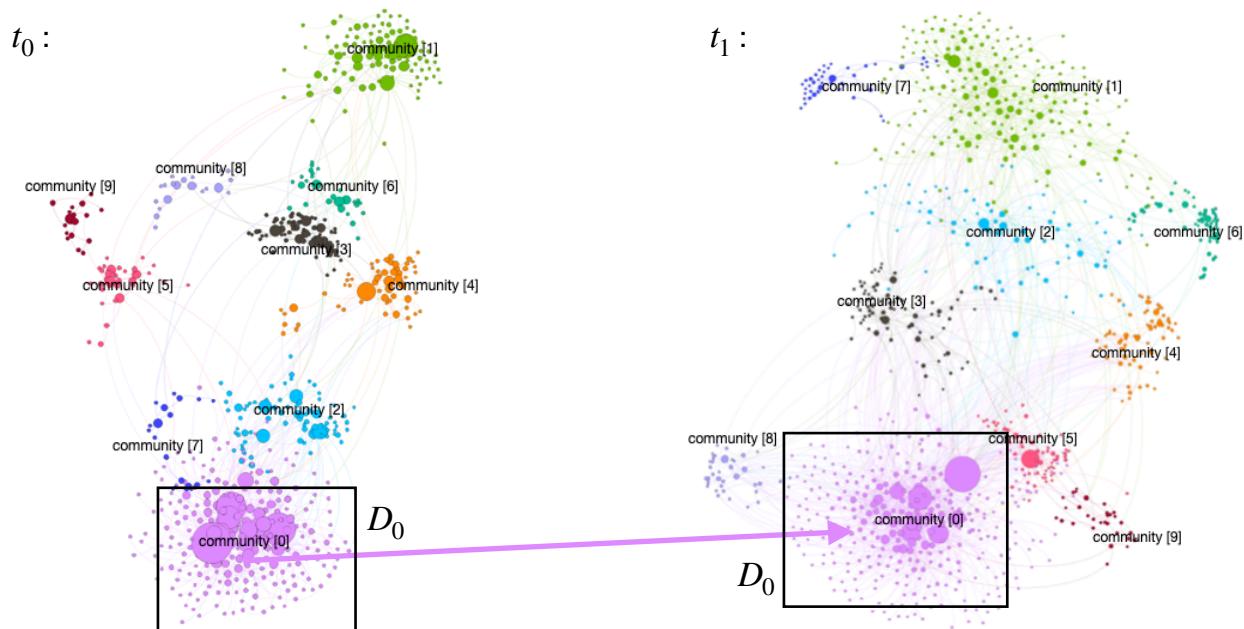
Εικόνα 3.19 Σχεδιάγραμμα των δυναμικών communities
 D_1, D_2, \dots, D_{23} τις χρονικές στιγμές t_0, t_1, \dots, t_4

Στην Εικόνα 3.19 παρατηρούμε πως μερικά δυναμικά communities μετά τη δημιουργία τους (birth) δεν εντοπίζονται σε επόμενες χρονικές στιγμές. Τα δυναμικά communities που δεν έχουν εντοπιστεί για δύο η παραπάνω χρονικές στιγμές θεωρούμε πως έχουν εξαφανιστεί (death). Στην εικόνα 3.20 βλέπουμε τα δυναμικά communities που εντοπίσαμε τις στιγμές t_0, t_1, \dots, t_4 χωρίς τα δυναμικά communities που τελικά εξαφανίζονται.



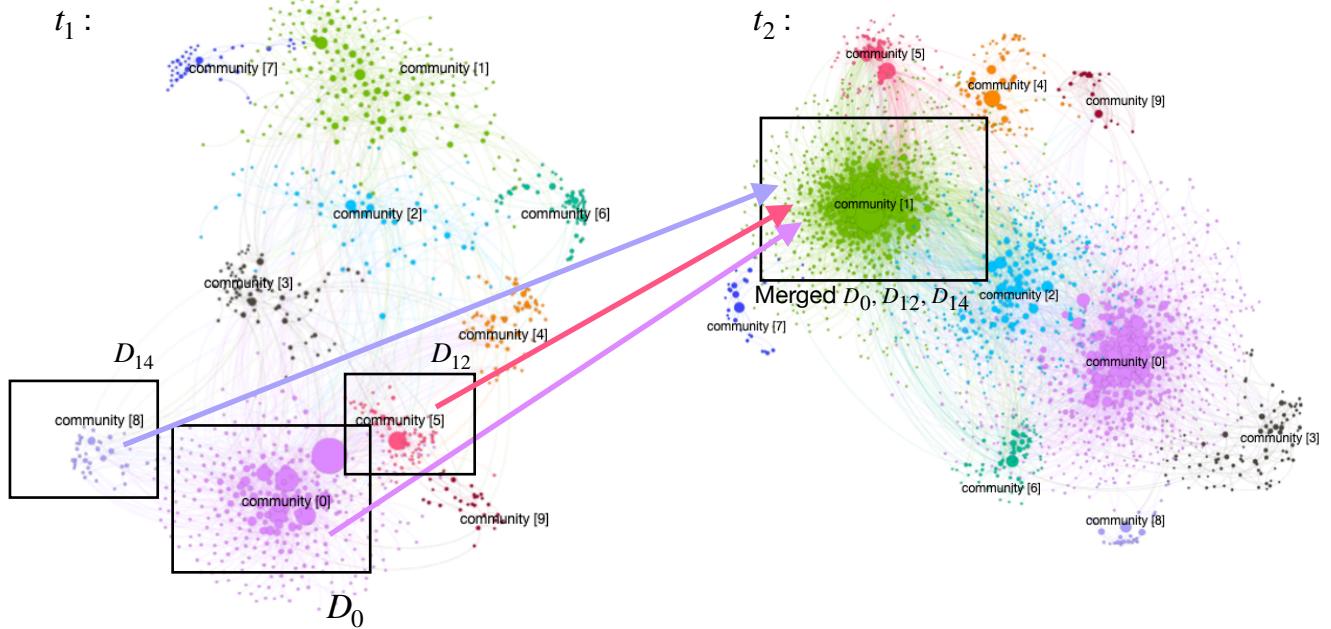
Εικόνα 3.20 Σχεδιάγραμμα των δυναμικών communities τις χρονικές στιγμές t_0, t_1, \dots, t_4 εχοντας αφαιρέσει τα δυναμικά communities που εξαφανίζονται στο χρόνο

Με την παραπάνω ανάλυση έχουμε πλέον πληροφορία μέσα από τα δυναμικά communities για την σχέση που έχουν τα στατικά communities μιας χρονικής στιγμής με τα στατικά communities μιας επόμενης χρονικής στιγμής. Για παράδειγμα, μέσα από το δυναμικό community D_0 διαπιστώνουμε ότι το στατικό community C_{00} (community[0] στον γράφο retweet γύρω και από τους δύο πολιτικούς λογαριασμούς τη χρονική t0: Ιανουάριος) αντιστοιχίζεται στο στατικό community C_{10} (community[0] στον γράφο retweet γύρω και από τους δύο πολιτικούς λογαριασμούς τη χρονική t1: Φεβρουάριος). Συμπεραίνουμε άρα ότι οι κόμβοι που αποτελούν το community[0] τον μήνα Ιανουάριο αποτελούν και το community[0] το μήνα Φεβρουάριο.



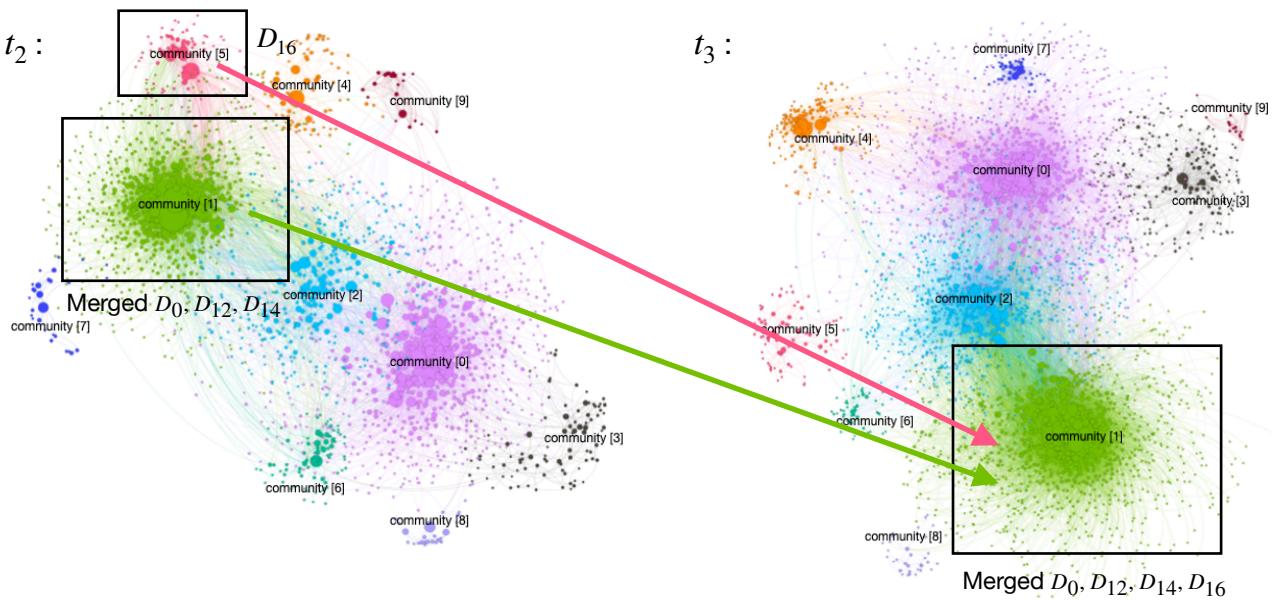
Εικόνα 3.21 Αντιστοιχίση των στατικών communities με τη βοήθεια του δυναμικού community D_0 . Αριστερά ο γράφος retweet την χρονική στιγμή t0. Δεξιά ο γράφος retweet την χρονική στιγμή t1.

Αντίστοιχα, την χρονική στιγμή t2 παρατηρούμε ότι υπάρχει συγχώνευση μεταξύ των δυναμικών communities D_0, D_{12}, D_{14} , έτσι οι κόμβοι που αποτελούν τα δυναμικά communities D_0, D_{12}, D_{14} τη χρονική στιγμή t1, αντιστοιχίζονται στους χρήστες του στατικού community C_{21} τη χρονική στιγμή t2 όπως φαίνεται στην Εικόνα 3.22.



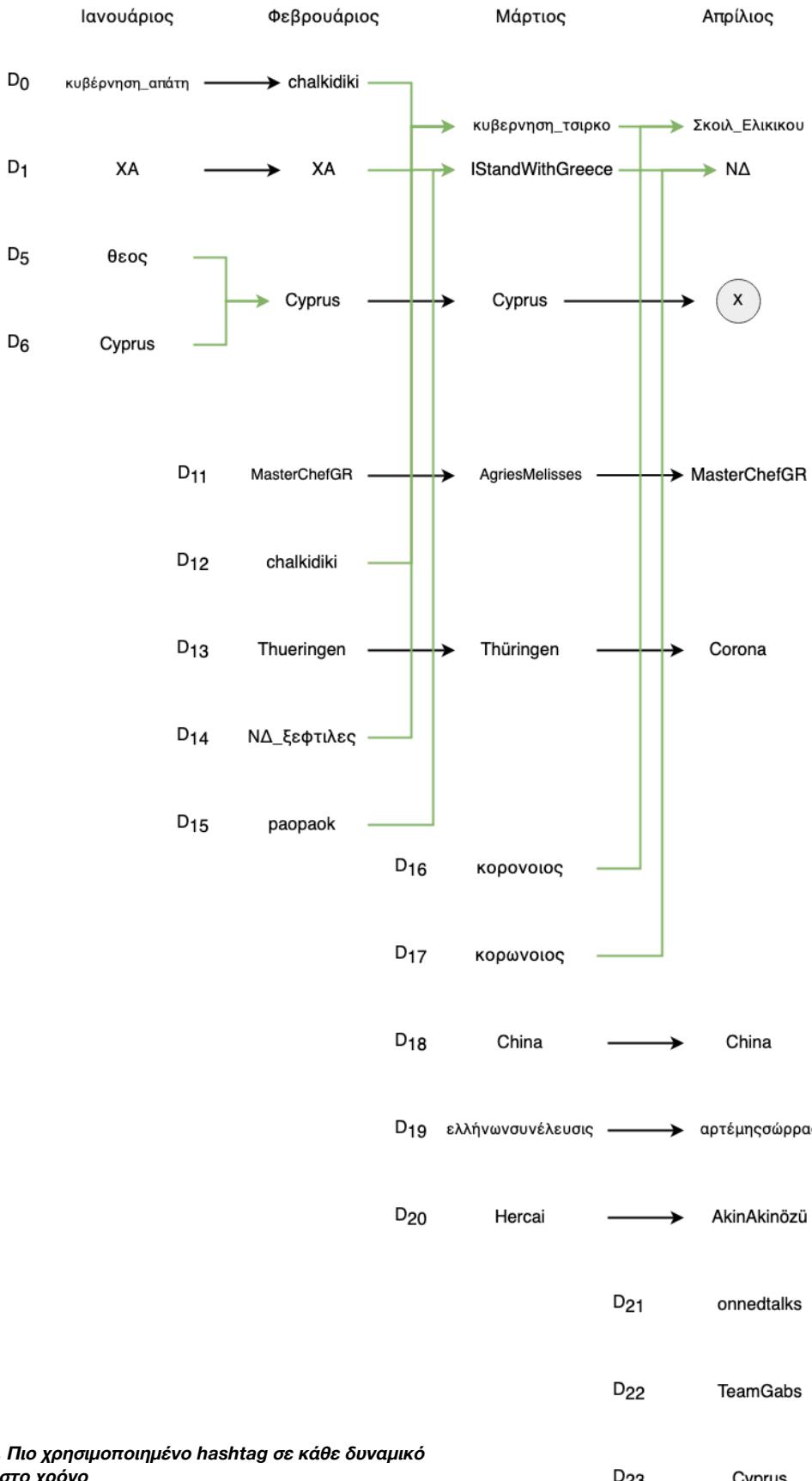
Εικόνα 3.22 Αντιστοίχιση των στατικών communities με τη βοήθεια του δυναμικού community D_0, D_{12}, D_{14} . Αριστερά ο γράφος retweet την χρονική στιγμή t1. Δεξιά ο γράφος retweet την χρονική στιγμή t2.

Την επόμενη χρονική στιγμή t3 παρατηρούμε ότι υπάρχει συγχώνευση μεταξύ των δυναμικών communities $D_0, D_{12}, D_{14}, D_{16}$, έτσι οι κόμβοι που αποτελούν τα δυναμικά communities $D_0, D_{12}, D_{14}, D_{16}$ τη χρονική στιγμή t2, αντιστοιχίζονται στους χρήστες του στατικού community C_{31} τη χρονική στιγμή t3 όπως φαίνεται στην Εικόνα 3.23.



Εικόνα 3.23 Αντιστοίχιση των στατικών communities με τη βοήθεια του δυναμικού community $D_0, D_{12}, D_{14}, D_{16}$. Αριστερά ο γράφος retweet την χρονική στιγμή t2. Δεξιά ο γράφος retweet την χρονική στιγμή t3.

Τέλος, έχοντας εντοπίσει τα δυναμικά communities βάση των στατικών communities και με χρήση της ανάλυσης των ποιό διάσημων hashtags που πραγματοποιήσαμε στην Ενότητα 3.5 μπορούμε να μελετήσουμε ποιό είναι το πιο χρησιμοποιημένο hashtag σε κάθε δυναμικό community στο χρόνο (Εικόνα 3.24).



Εικόνα 3.24. Πιο χρησιμοποιημένο hashtag σε κάθε δυναμικό community στο χρόνο

D23 Cyprus

4. Επίλογος

4.1 Σύνοψη

Στην συγκεκριμένη διπλωματική εργασία είδαμε πως μπορούμε να κατασκευάσουμε ένα εργαλείο για την συλλογή δεδομένων γύρω από έναν λογαριασμό στο Twitter. Το εργαλείο αυτό χρησιμοποιεί ένα σύνολο από αποδοτικές multi-threaded τεχνικές για να συλλέξει δεδομένα από το TwitterAPI, όπως λογαριασμούς χρηστών, tweets, retweets, hashtags. Με αυτό τον τρόπο καταφέραμε να μειώσουμε σημαντικά τον χρόνο συλλογής των δεδομένων. Επίσης, είδαμε πως μπορούμε να αποθηκεύσουμε αυτά τα δεδομένα σε μια σχεσιακή βάση, διευκολύνοντας έτσι την επεξεργασία και την προσπέλαση τους.

Είδαμε πως μπορούμε να κατασκευάσουμε τους γράφους που μας ενδιαφέρουν με τη χρήση του module NetworkX και ύστερα πως μπορούμε να τους οπτικοποιήσουμε με τη χρήση του εργαλείου Gephi.

Πάνω στους γράφους που κατασκευάσαμε είδαμε πως μπορούμε να εντοπίσουμε τα communities που σχηματίζονται μεταξύ των χρηστών με χρήση του modularity αλγόριθμου. Πιο συγκεκριμένα, εντοπίσαμε τα communities στον Retweet γράφο για διακριτές χρονικές στιγμές και για κάθε community βρήκαμε τα hashtags που χρησιμοποίησαν περισσότερο οι χρήστες.

Τέλος, μελετήσαμε πως τα communities στον Retweet γράφο εξελίσσονται στον χρόνο αντιστοιχίζοντας τα στατικά communities που υπολογίσαμε για κάθε χρονική στιγμή σε δυναμικά communities. Κάνοντας χρήση των hashtags που έχουν χρησιμοποιήσει οι χρήστες πιο πολύ προσδιορίσαμε το περιεχόμενο των δυναμικών communities και είδαμε πως αυτό μεταβάλεται στο χρόνο.

4.2 Μελλοντικές επεκτάσεις

Βελτίωση της απόδοσης του εργαλείου συλλογής δεδομένων. Πιο συγκεκριμένα θα μπορούσαμε να σχεδιάσουμε έναν αλγόριθμο ο οποίος κάνει μια εκτίμηση του φόρτου που αντιστοιχεί σε κάθε νήμα και να διαμοιράζει την εργασία στα νήματα με πιο δίκαιο τρόπο.

Προσδιορισμός του περιεχομένου των δυναμικών communities με βάση το περιεχόμενο των tweets. Όπως είδαμε στην ενότητα 3.6 μελετήσαμε την εξέλιξη των communities του Retweet γράφου στο χρόνο και προσδιορίσαμε το περιεχόμενο του κάθε community με βάση τα hashtags που χρησιμοποίησαν οι χρήστες. Μια ιδέα που θα μπορούσαμε να πραγματοποιήσουμε στο μέλλον είναι ο

προσδιορισμός του περιεχομένου των communities με βάση το περιεχόμενο των tweets που έχουν χρησιμοποιήσει οι χρήστες. Τέλος, θα μπορούσαμε να συγκρίνουμε πως εξελίσσεται το περιεχόμενο των δυναμικών communities στο χρόνο, όταν το ορίζουμε με βάση τα hashtags σε αντίθεση με όταν το ορίζουμε με βάση τα tweets.

5. Βιβλιογραφία

- [1] Clauset, A., Newman, M. E., & Moore, C. "Finding community structure in very large networks." *Physical Review E* 70(6), 2004.
- [2] M. E. J Newman 'Networks: An Introduction', page 224 Oxford University Press 2011.
- [3] Derek Greene, Do'nal Doyle, Padraig Cunningham: *Tracking the Evolution of Communities in Dynamic Social Networks*.