Ατομική

Διπλωματική Εργασία

ΕΞΑΓΩΓΗ ΣΥΝΑΙΣΘΗΜΑΤΟΣ ΚΑΙ ΚΑΤΗΓΟΡΟΠΟΙΗΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ FACEBOOK

Άγγελος Σάββα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΞΑΓΩΓΗ ΣΥΝΑΙΣΘΗΜΑΤΟΣ ΚΑΙ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ FACEBOOK

Άγγελος Σάββα

Επιβλέπων Καθηγητής

Μάριος Δικαιάκος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέπων καθηγητή μου, Δρ. Μάριο Δικαιάκο για το ενδιαφέρον που έδειξε σε όλη τη διάρκεια του χρόνου, τις συμβουλές του και τη στήριξη του όσο αφορά αυτή την διπλωματική εργασία.

Θα ήθελα επίσης να ευχαριστήσω τον Δημήτρη Πασχαλίδη για τη σημαντική βοήθεια που μου έδωσε όσο αφορά πρακτικά θέματα για την υλοποίηση αυτής της εργασίας.

Θα ήθελα να ευχαριστήσω όλους τους καθηγητές μου στο Πανεπιστήμιο Κύπρου που μου πρόσφεραν τις γνώσεις και τις δεξιότητες που απέκτησα σε αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την έμπρακτη υποστήριξη που μου πρόσφερε στα φοιτητικά μου χρόνια ώστε να φέρω εις πέρας όλους τους ακαδημαϊκούς μου στόχους.

Περίληψη

Ζούμε σε μία εποχή όπου η χρήση κοινωνικών δικτύων αποτελεί μέρος της καθημερινότητας μας, με 60 % των ανθρώπων παγκοσμίως να τα χρησιμοποιούν και συγκεκριμένα το Facebook να έχει σχεδόν 2.5 δισεκατομμύρια ενεργούς χρήστες κάθε μήνα.

Επίσης η διάρκεια χρήσης των κοινωνικών αυτών δικτύων από τους χρήστες συνεχώς αυξάνεται και ολοένα οι χρήστες βρίσκονται αντιμέτωποι με περισσότερο περιεχόμενο που μοιράζονται μεταξύ τους αλλά και άρθρα διάφορων ειδήσεων.

Είναι σημαντικό να αντιληφθούμε την βαρύτητα που μπορούν να έχουν τα κοινωνικά αυτά δίκτυα στην ψυχολογία μας και ακόμη στη διαμόρφωση του χαρακτήρα μας.

Για αυτό το λόγο σε αυτή την διπλωματική εργασία έχουμε βάλει στόχο να δημιουργήσουμε ένα πρόγραμμα plug-in το οποίο θα έχει την δυνατότητα με την εγκατάσταση του να βελτιώσει την εμπειρία του χρήστη και να τον προστατέψει από κακόβουλο περιεχόμενο.

Το πρόγραμμα που δημιουργήσαμε έχει την δυνατότητα να εξάγει την πληροφορία που βρίσκετε στο προφίλ του χρήστη και με χρήση έξυπνων αλγορίθμων να υπολογίσει την συναισθηματική τιμή που περιέχουν με στόχο να ενημερώσει τον χρήστη.

Ακόμη το πρόγραμμα μας μπορεί να εντοπίσει τα άρθρα που βρίσκονται στο προφίλ του χρήστη και να ελέγξει κατά πόσο αυτά είναι ασφαλής και όχι στην κατηγορία των fake news. Ανάλογα αν το πρόγραμμα μας εντοπίσει fake news θα μπορέσει να ενημερώσει τον χρήστη.

Αυτές οι λειτουργίες τρέχουν ζωντανά και ενημερώνουν τον χρήστη προτού φτάσει να είναι αντιμέτωπος με το συγκεκριμένο περιεχόμενο.

Το πρόγραμμα έχει επίσης την δυνατότητα να αποθηκεύσει όλα αυτά τα δεδομένα του χρήστη και με το πάτημα ενός κουμπιού να ενημερώσει των χρήστη με μια στατιστική αναφορά για το συναίσθημα που εντοπίστηκε στο προφίλ του από την στιγμή που ξεκίνησε να χρησιμοποιεί το πρόγραμμα μας.

Για να πραγματοποιηθούν όλες οι λειτουργείες του προγράμματος έχουν αναπτυχθεί 2 προγράμματα τα οποία λειτουργούν μεταξύ τους ως ένα, το plug-in (browser side) και πλατφόρμα Node Js(server side). Λεπτομερείς ανάπτυξη των λειτουργειών και των δύο προγραμμάτων με αναφορές από κομμάτια κώδικα, παρουσιάζονται στα παρακάτω κεφάλαια.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Το κεφάλαιο αυτό περιέχει τα εισαγωγικά θέματα της διπλωματικής εργασίας, οι στόχοι που τέθηκαν για την εργασία και μια σύντομη περιγραφή της δομής αυτής της αναφοράς και το περιεχόμενων των κεφαλαίων της.

1.1 Υποκίνηση της Εργασίας

Η χρήση των κοινωνικών δικτιών στη σήμερον ημέρα αποτελεί μέρος της καθημερινότητας μας. Με 60% των ανθρώπων παγκοσμίως να χρησιμοποιούν κοινωνικά δίκτυα δεν είναι τυχαίο το γεγονός ότι η πληροφορία είναι ο πιο πολύτιμος πόρος στην σύγχρονη εποχή.

Το κοινωνικό δίκτυο της Facebook το οποίο αυτή η διπλωματική εργασία έχει στόχο να αξιοποιήσει έχει σχεδόν 2.5 δισεκατομμύρια ενεργούς χρήστες κάθε μήνα. Με ένα τόσο μεγάλο αριθμό χρηστών είναι αναμενόμενο ότι και το περιεχόμενο το οποίο οι χρήστες του βρίσκονται αντιμέτωποι θα είναι ανάλογο σε όγκο.

Οι άνθρωποι που χρησιμοποιούν τα κοινωνικά δίκτια συνεχώς αυξάνουν τις ώρες τις οποίες είναι ενεργοί και εκτεθειμένοι στο περιεχόμενο τους. Αυτό οδηγεί αναπόφευκτα στη διαδικασία να επηρεάζουν βραχυπρόθεσμα την ψυχολογική τους κατάσταση αλλά και μακροπρόθεσμα να διαμορφώνουν απόψεις, αντιλήψεις σε διάφορα θέματα και χαρακτηριστικά της προσωπικότητας τους.

Είναι σημαντικό να αντιληφθούμε πόσο σημαντικό ρόλο έχουν στην σήμερον ημέρα τα κοινωνικά δίκτια στην ψυχολογία μας και στην προσωπικότητα μας. Η έκθεση μας σε κάποιο συγκεκριμένο θέμα η συναίσθημα επανειλημμένα μπορεί μόνιμα να μας επηρεάσει θετικά η και αρνητικά.

Επίσης είναι σημαντικό να τονίσουμε πως τα κοινωνικά δίκτια δεν έχουν βοηθήσει μόνο στην διευκόλυνση διακίνησης όλων αυτόν των απόψεων φορτισμένες με διάφορα συναισθήματα μεταξύ χρηστών, αλλά και στη διευκόλυνση διακίνησης ειδήσεων.

Με την ευκολία στην διακίνηση ειδήσεων σε τέτοιες ισχυρές πλατφόρμας με τεράστια νούμερα χρηστών στην διάθεση τους, παρατηρήθηκε και η αύξηση στις ψεύτικές ειδήσεις. Ειδήσεις οι οποίες δεν μπορούν να έχουν καλές προθέσεις αφού το περιεχόμενο τους δεν είναι αληθές.

Για αυτούς τους λόγους κρίθηκε σημαντικό να δημιουργηθεί κάποιο πρόγραμμα σε αυτήν την διπλωματική εργασία το οποίο θα μπορούσε να εντοπίσει όλα αυτά, που μπορεί συχνά στο ανθρώπινο μάτι να παίρνουν απαρατήρητα, λόγο ελλιπές κριτικής σκέψη ή γνώσεις στο συγκεκριμένο θέμα.

1.2 Στόχοι της εργασίας

Στόχοι αυτή της διπλωματικής εργασίας είναι δημιουργία ενός προγράμματος το οποίο μπορεί να ελέγξει το περιεχόμενο της πλατφόρμας του Facebook το οποίο ο χρήστης θα βρισκόταν αντιμέτωπος και πρόωρα να τον ενημερώσει ανάλογα.

Πιο συγκεκριμένα η διπλωματικοί εργασία έχει στόχο να δημιουργήσει μέσω JavaScript, ένα web scraper plug-in το οποίο θα μας δώσει πρόσβαση στο περιεχόμενο του χρήστη. Μέσω έξυπνων αλγόριθμων να μπορέσουμε να εξάγουμε από το περιεχόμενο αυτό το συναίσθημα το οποίο περιέχει αλλά και να ελέγξουμε την εγκυρότητα των άρθρων τα οποία παρουσιάζονται. Αφού οι έξυπνοι αλγόριθμοι θα έχουν εξάγει τα συμπεράσματα τους, να ενημερώσουμε τους χρήστες μας για αυτήν την πληροφορία ώστε αυτοί θα μπορούν να έχουν

κάποιο έλεγχο πλέων σε όλοι αυτήν την πληροφορία που συνεχώς τους βομβαρδίζει, να μπορούν να επιλέξουν κατά πόσο θέλουν να την διαβάσουν.

Αυτά όλα θα πραγματοποιούνται σε πραγματικό χρόνο άρα και το πρόγραμμα θα πρέπει να είναι γρήγορο, με ταχύτητα τουλάχιστο που θα μπορεί να ελέγξει και να εξάγει αυτά τα συμπεράσματα για κάθε δημοσίευση πριν προλάβει ο χρήστης να είναι αντιμέτωπος της.

Το πρόγραμμα επίσης θα μπορέσει να πραγματοποιήσει όλα τα παραπάνω χωρίς να υπάρχει καθόλου επίβλεψη από κάποιο χρήστη.

1.3 Περίγραμμα της εργασίας

Το πρώτο κεφάλαιο περιέχει τον σκοπό, τον λόγο υποκίνησης και τους πολλαπλούς στόχους που έχει αυτή η διπλωματική εργασία.

Το δεύτερο κεφάλαιο περιέχει το θεωρητικό υπόβαθρο το οποίο είναι απαραίτητο για να μπορέσει οποιοσδήποτε να κατανοήσει τα ακόλουθα κεφάλαια. Περιέχει βασικές έννοιες και τεχνολογίας που χρησιμοποιήθηκαν στην ανάπτυξη του προγράμματος αυτής της διπλωματικής εργασίας.

Το τρίτο κεφάλαιο περιέχει την περιγραφή και αναλύει όλα τα βήματα του προγράμματος από την συλλογή των δεδομένων μέχρι την παρουσίαση των αποτελεσμάτων στον χρήστη. Αυτή η ανάλυση των βημάτων επιτρέπει στον χρήστη να δημιουργήσει μία εικόνα στο μυαλό τους για την ροή των δεδομένων, πως το πρόγραμμα χειρίζεται τα δεδομένα αυτά σε κάθε στιγμή και πως αντιδρά σε συγκεκριμένα δεδομένα.

Το τέταρτο κεφάλαιο περιέχει αναλυτικά όλες τις τεχνικές λεπτομέρειες για την υλοποίηση αυτού του προγράμματος από την εξαγωγή των δεδομένων μέχρι την παρουσίαση αποτελεσμάτων στον χρήστη. Ονομαστικά οι τεχνολογίες που χρησιμοποιήθηκαν σε κάθε βήμα μαζί με παραδείγματα από τον κώδικα.

Το πέμπτο κεφάλαιο περιέχει την αξιολόγηση του συστήματος. Κυρίως περιέχει την αξιολόγηση για τα 3 πιο σημαντικά βήματα του προγράμματος τα οποία είναι η εξαγωγή των δεδομένων, τα φίλτρα το οποία χρησιμοποιήθηκαν για να προετοιμάσουν αυτά τα δεδομένα, και η εξαγωγή συναισθήματος.

Το έκτο κεφάλαιο περιέχει τα συμπεράσματα που έχουμε εξάγει με την δημιουργία αυτού του προγράμματος και την εκτενής χρήση του, αλλά και ποιες θα ήταν οι μελλοντικές εργασίες που θα μπορούσαν να πραγματοποιηθούν με στόχο να βελτιώσουν αυτή την διπλωματική εργασία.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό παρατίθενται κάποιες βασικές θεωρητικές έννοιες που είναι απαραίτητες για την μετέπειτα κατανόηση του κειμένου.

Αρχικά αναφέρονται κάποιες βασικές πληροφορίες για το γνωστό κοινωνικό δίκτυο της Facebook και έπειτα αναλύονται χρήσιμες τεχνικές και εργαλεία. Αυτή η βασική θεμελίωση είναι ιδιαίτερα σημαντική καθώς τα μετέπειτα κεφάλαια δείχνουν πώς προσαρμόζονται και πώς αξιοποιούνται αυτές οι τεχνικές στο πρόγραμμα αυτής της διπλωματικής εργασίας

2.1 Facebook

Το Facebook είναι το πιο δημοφιλές κοινωνικό δίκτυο στον κόσμο το οποίο δημιουργήθηκε από τον Mark Zuckerberg και τον Edward Saverin το 2004 στο Harvard University. Το Facebook βρίσκεται στην πρώτη θέση για τις λειτουργίες που προσφέρει στους χρήστες του, οι βασικές των οποίων περιλαμβάνουν :

● Διατήρηση και διαμόρφωση μιας λίστα φίλων και την επιλογή προσαρμογής ρυθμίσεων στο ποιοι θα μπορούν να βλέπουν το περιεχόμενο του προφίλ σου.

● Μεταφόρτωση φωτογραφιών και ομαδοποίηση τους σε άλμπουμ τα οποία μπορεί ο χρήστης να μοιραστεί μαζί με φίλους.

● Διαδραστική online συνομιλία και δυνατότητα να αφήσει ο χρήστης σχόλιο σε σελίδα ή σε κάποια δημοσίευση των φίλων σου

● Ομάδες και Σελίδες στις οποίες ο χρήστης μπορεί να κάνει like και να μένει ενήμερος για το περιεχόμενο που θα ανεβάζουν

● Stream live videos

Το περιεχόμενο που μοιράζεται ο χρήστης μπορεί να είναι προσβάσιμο από όλους, ή από καθορισμένη ομάδα φίλων, ή ακόμη και από ένα μόνο πρόσωπο.

Οι κύριες σελίδες του Facebook είναι:

● Η **αρχική σελίδα (homepage / News Feed)** η οποία σελίδα παρουσιάζει όλες τις τελευταίες δημοσιεύσεις από τους φίλους του χρήστη και τα σχόλια που δέχτηκαν, άρθρα και νέα από τις σελίδες και τις ομάδες στις οποίες ο χρήστης είναι μέλος, αλλά και sponsored δημοσιεύσεις οι οποίες αποτελούνται από διαφημίσεις προϊόντων και υπηρεσιών.

● Το **προφίλ** (**timeline / profile page)** η οποία σελίδα παρουσιάζει τις προσωπικές πληροφορίες που εισήχθησαν από τον κάθε χρήστη στο Facebook για τον εαυτό του, φωτογραφίες τις οποίες ανέβασε αλλά και δημοσιεύσεις από φίλους οι οποίες σχετίζονται απευθείας με τον χρήστη.

● Η **σελίδα** (**page)** είναι προφίλ προσβάσιμο από όλους τους χρήστες με κάποιο συγκεκριμένο θέμα όπως μάρκα, διάσημο πρόσωπο, επιχείρηση ή ακόμη και κάποιο στόχο. Χρήστες του Facebook μπορούν να κάνουν like στη σελίδα για να γίνουν μέλη έτσι ώστε να είναι ενημερωμένοι και να λαμβάνουν ειδοποιήσεις όσον αφορά το περιεχόμενο της.

● Η **ομάδα** (**group)** είναι σελίδα με περιεχόμενο κάποιο στόχο, ενδιαφέρον ή και άποψη για κάποιο θέμα, όπου αν κάποιος χρήστης ενδιαφέρεται μπορεί να γίνει μέλος αυτής της ομάδας έτσι ώστε και αυτός να μπορεί να αλληλοεπιδρά με τα άλλα μέλη της ομάδας και να βλέπει η/και να ανεβάζει περιεχόμενο.

Η Facebook μετά από το περιστατικό που είχε προκύψει με την Cambridge Analytica η οποία με την ευκολία του Graph API μπόρεσαν να αποκτήσουν κρίσιμα προσωπικά στοιχεία των χρηστών και μετέπειτα να παραπλανήσουν τους χρήστες χρησιμοποιώντας αυτά τα στοιχεία, έθεσε σε εφαρμογή πολλαπλά περιοριστικά μέτρα τα οποία θα μειώσουν την πιθανότητα ένα τέτοιο περιστατικό να ξανασυμβεί. Ένα από αυτά τα περιοριστικά μέτρα είναι και η διαγραφή του News Feed API. Αυτό μας ανάγκασε να εξερευνήσουμε εναλλακτικές λύσεις με τις οποίες θα μπορέσουμε να έχουμε πρόσβαση σε αυτά τα δεδομένα που χρειαζόμαστε να αποκτήσουμε για σκοπούς της διπλωματικής αυτής εργασίας.

2.2 Χρήσιμες τεχνικές και εργαλεία

2.2.1 HTML

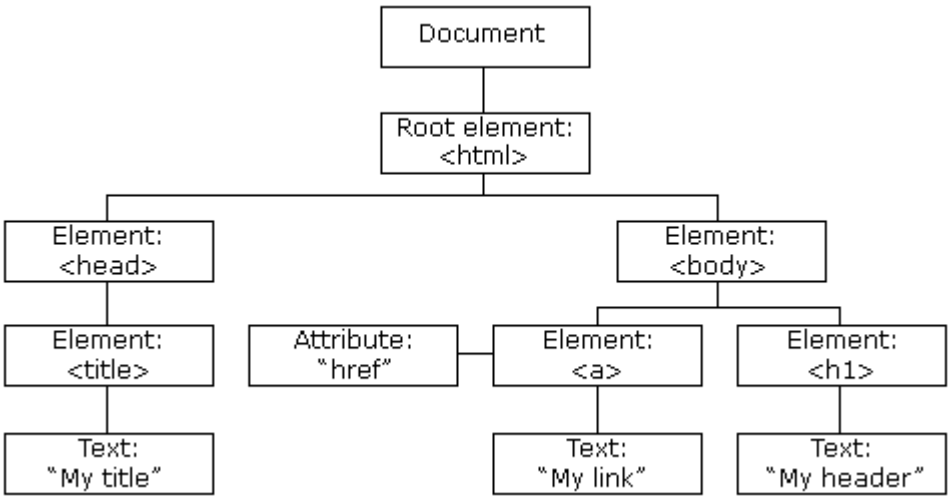
Hyper Text Markup Language (HTML) είναι η γλώσσα που χρησιμοποιείται για όλες τις ιστοσελίδες. Η απλή δομή της επιτρέπει ακόμα και σε αρχάριους χρήστες να εξοικειωθούν μαζί της σε πολλή μικρό χρονικό διάστημα, πράγμα που τη βοήθησε στη αύξηση της δημοτικότητας της σε σημείο που θεωρείται πλέον ως επίσημο πρότυπου ιστού.

Οι σελίδες HTML αποτελούνται από πολλαπλά tags γνωστά και ως elements των οποίων μέσω του πότε ανοίγουν και το πότε κλείνουν δημιουργείται και η έννοια της ιεραρχίας. Ένα HTML tag ανοίγει με την εντολή και κλείνει αντίστοιχα με την εντολή **<tag>** η **</tag>**. Με το άνοιγμα του tag αυτό μπορεί να περιέχει κείμενο, σύνδεσμο ή ακόμη και άλλα tags, που μέσω αυτού υποδηλώνετε και η ιεραρχία τους. Κάθε tag έχει δημιουργηθεί με τον δικό του σκοπό για να περιέχει κάποιου συγκεκριμένου είδους περιεχόμενο ή απλά για να διασφαλίσει κάποιο χώρο στην σελίδα μέσω από τα κληρονομικά του χαρακτηριστικά. Δύο κύρια είδους tag για διασφάλιση χώρου είναι τα block-level και τα inline-block. Τα block-level διασφαλίζουν όλο το διαθέσιμο χώρο της γραμμής και αναγκάζουν άλλα elements με ίδιο βαθμό ιεραρχίας να ξεκινήσουν στην επόμενη γραμμή τής σελίδας, σε αντίθεση με τα inline-block elements τα οποία μοιράζονται των διαθέσιμο χώρο της γραμμής μεταξύ τους.

2.2.2 DOM

Το Document Object Model (DOM) είναι ένα αντικειμενοστεφής λογικό πρότυπο της δομής που χρησιμοποιείτε για να καθορίσει κάποιες βασικές αρχές στα HTML και XML έγγραφα. Μέσω αυτή της λογικής δομής προσφέρεται η δυνατότητα στο χρήστη να δημιουργεί , τροποποιεί , διαγράφει , προσθέτει και να αποκτά πρόσβαση στα στοιχεία και το περιεχόμενο του έγγραφου.

Το Object Model έχει δομή τέτοια που παρομοιάζεται με αυτή του δέντρου όπου στη προγραμματιστική διατύπωση υπάρχουν στοιχεία που έχουν τον ρόλο του πατέρα, των παιδιών η και τα δύο. Ακολουθώντας τα στοιχεία αυτά θα οδηγηθούμε σε ακρινά στοιχεία που δεν έχουν παιδία, γνωστά ως τα φύλλα του δέντρου όπου συνήθως περιέχουν το περιεχόμενο που εμφανίζεται στο έγγραφο(π.χ. κείμενο, εικόνα, σύνδεσμος). Σε ένα έγγραφο μπορούν να υπάρχουν πολλά τέτοιου είδους δέντρα.



2.2.3 JavaScript

Η JavaScript είναι γλώσσα προγραμματισμού που κυρίως χρησιμοποιείται για ανάπτυξη εφαρμογών ιστού. Είναι client-side γλώσσα, που σημαίνει ότι τρέχει στον φυλλομετρητή του χρήστη και όχι στο Server. Αυτό δίνει την δυνατότητα στη γλώσσα να τρέχει διάφορες λειτουργείες και να εξάγει αποτελέσματα χωρίς όμως να υπάρχει κάποια διασταύρωσή δεδομένων με των Server. Με αυτό των τρόπο η JavaScript μπορεί δυναμικά να αποκτήσει και να τροποποιήσει δεδομένα που βρίσκονται στη σελίδα.

Ακόμη στην JavaScript μπορούν να υλοποιηθούν λειτουργείς γνωστές ως Event Listeners οι οποίες λειτουργείς περιμένουν να εντοπίσουν από την σελίδα κάποια συγκεκριμένη ενέργεια του χρήστη για να εκτελέσουν της επιθυμητές ενέργειες.

Γνωστές τέτοιες ενέργειες από την πλευρά του χρήστη είναι :

- Το κλικ του ποντικιού
- Το φτερούγισμα (hover)
- Η μεταβολή της κατάστασης ενός αντικειμένου (onChange)

Παρόλα αυτά, η JavaScript δεν είναι περιορισμένη μόνο σε λειτουργείς οι οποίες δεν αφορούν την επικοινωνία με κάποιο server. Η αποστολή και η λήψη δεδομένων με server είναι δυνατή και η χρήση τους είναι συνηθισμένη.

Γνωστές τέτοιες ενέργειες είναι :

- Με την event listeners για την δυναμική ανανέωση του περιεχομένου.
- Αναμονή για κάποιο μήνυμα (onMessage)
- Χρήση AJAX

Η AJAX, που σημαίνει ασύγχρονη JavaScript και XML είναι η γνωστή τεχνική όπου ένα JavaScript πρόγραμμα μπορεί να στείλει και να λάβει δεδομένα ασύγχρονα από ένα server. Οι ασύγχρονές αυτές λειτουργείες επιτρέπουν αφού πραγματοποιηθεί κάποιο αίτημα από το πρόγραμμα στο server να συνεχιστεί κανονικά η ροή του κώδικα χωρίς να σημαίνει όμως ότι αυτό το αίτημα έχει ολοκληρωθεί. Το πρόγραμμα λαμβάνει αυτά τα δεδομένα αυτόματα αφού ολοκληρωθεί το αίτημα και αυτό έχει ως αποτέλεσμα να μπορούν να ολοκληρώνονται πολλαπλά αιτήματα ταυτόχρονα σε μικρότερο χρονικό διάστημα. Η πιο γνωστή επιλογή μορφή για μεταφορά δεδομένων στα AJAX είναι η JSON.

JSON (JavaScript Object Notation) είναι μια εύκολη και κατανοητή μορφή για μεταφορά δεδομένων που η μορφή τους αποτελείται από στοιχεία με όνομα και τιμή.

```
{
    hey: "guy",
    anumber: 243,
```

Είναι πιθανό όμως μερικές φορές να χρειαζόμαστε αυτά τα δεδομένα από το αίτημα που έχουμε κάνει για της πιο παρακάτω λειτουργείες μας. Σε αυτές της περιπτώσεις υπάρχουν πολλαπλοί τρόποι και τεχνικές για να μετατρέψουμε αυτά τα ασύγχρονα αιτήματα σε πιο απλά και γνωστά με σειριακή συμπεριφορά.

Κάποιες τεχνικές είναι η χρήση λειτουργειών :

- Async/Await (Λειτουργείες των οποίων η ολοκλήρωση είναι αναγκαία για να προχωρήσει η ροή του προγράμματος).
- Promises (Λειτουργείες των οποίων η επιστροφή τιμής είναι αναγκαία για να προχώρηση η ροή του προγράμματος).
- Set timeout (Παγοποίηση του κώδικα για κάποιο επιθυμητό χρόνο).

2.2.4 Node Js

Η Node Js είναι server-side πλατφόρμα η οποία είναι χτισμένη επάνω στη μηχανή τής Google Chrome’s JavaScript και είναι κυρίως γραμμένη σε JavaScript. Το σπουδαίο χαρακτηριστικό στης δικτυακές εφαρμογές που είναι γραμμένες σε Node Js είναι ότι η επικοινωνία μεταξύ server-side και client-side, δεν είναι πλέον αναγκαία απαραίτητη, το αίτημα να εκτελείται από την πλευρά του χρήστη.

Με την Node Js και τα server-side χαρακτηριστικά της μπορούμε να αποστέλλουμε δεδομένα μεταξύ του browser-side plug-in μας και ακόμη να έχουμε πρόσβαση σε αποθηκευτικό χώρο πέραν της μηχανής του χρήστη.

Επίσης, μέσω της Node Js έχουμε πρόσβαση σε μια από της πιο μεγάλες βιβλιοθήκες με πακέτα(NPM) τα όποια αποτελούν το σημαντικότερο κομμάτι για την προ επεξεργασία δεδομένων και υπολογισμών για το συναίσθημα που περιέχουν.

2.2.5 NPM

Η NPM (Node Package Manager) είναι η μεγαλύτερη βιβλιοθήκη με πακέτα με αριθμό που ξεπερνά της 800 χιλιάδες. Είναι ανοικτής πηγής και η χρήση της είναι δωρεάν. Η εγκατάσταση της NPM γίνεται με αυτή της Node Js αφού αρχικά είχε κατασκευαστεί για να προσφέρει καλύτερη διαχείριση στις βιβλιοθήκες της Node Js, για έλεγχο εξαρτήσεων και έκδοσης.

2.2.6 jQuery

jQuery είναι βιβλιοθήκη στη JavaScript η οποία μετατρέπει πολλαπλές γραμμές κώδικα γραμμένες χωρίς αυτήν, σε μικρότερες απλές μεθόδους. Απλοποιώντας την διαδικασία και την πολυπλοκότητα του κώδικα, λειτουργείες όπως η εξόρυξη και η μετατροπή δεδομένων στο φυλλομετρητή μπορούν να εφαρμοστούν με μια μόνο γραμμή κώδικα. Όχι μόνο, αλλά η jQuery μπορεί ακόμη να απλοποιήσει λειτουργείες για την ανταλλαγή δεδομένων με AJAX calls.

2.2.7 NoSQL

NoSQL είναι βάσεις όπου σε αντίθεση της παραδοσιακές βάσεις SQL

- Δεν είναι απαραίτητο κάποιο βασικό μοντέλο
- Επιτρέπει να αποθηκευτούν δεδομένα χωρίς να υπάρχει καθορισμένο ορισμός τύπου
- Επιτρέπει να αποθηκευτούν επιπρόσθετα δεδομένα χωρίς να αλλάξει το σχήμα της βάσης
- Μπορούν να αυξήσουν την ταχύτητα και χωρητικότητα εξυπηρέτησης πολύ πιο εύκολα.

Αυτό σημαίνει ότι η μετατροπή του σχήματος για τα δεδομένα που θα αποθηκεύονται στην βάση μπορεί να πραγματοποιηθεί οποιαδήποτε στιγμή χωρίς καμία δυσκολία, αλλά ακόμη και ότι οι NoSQL βάσεις είναι πιο ανθεκτικές στην κλιμάκωση. Για το λόγο ότι απλά προσθέτουν περισσότερους server στη βάση σε σχέση με SQL βάσεις για τις οποίες η αναβάθμιση των εξαρτημάτων του server είναι απαραίτητη.

2.2.8 Συλλογή δεδομένων – Web Scraping

Web Scraping είναι η τεχνική που εφαρμόζεται συχνά σε όλο το διαδίκτυο για συλλογή δεδομένων και μετέπειτα αποθήκευση των δεδομένων σε κάποιο τοπικό αρχείο ή κάποια βάση.

Αφού τα δεδομένα οποιασδήποτε ιστοσελίδας φορτωθούν στο φυλλομετρητή αυτό σημαίνει πως τα δεδομένα βρίσκονται εντός ενός ή πολλών καθορισμένων μοντέλων DOM. Με το καθορισμένο πλέον μοντέλο, η τεχνική web scrapping χρησιμοποιώντας την jQuery θα μπορέσει να κάνει exploit τα elements από το DOM τα οποία μας ενδιαφέρουν και να ζητήσουμε από αυτά το επιθυμητό περιεχόμενο τους.

Οι δυνατότητα της τεχνικής του web scraping στο να προσαρμόζεται εύκολα στο ξεχωριστό DOM μοντέλο της κάθε σελίδας και η εξαγωγή των δεδομένων από τα στοιχεία που μας ενδιαφέρουν, χωρίς την χρήση έτοιμων λογισμικών API, είναι ο λόγος που λήφθηκε η απόφαση για τη χρήση της σε αυτή την ατομική διπλωματική εργασία.

Κεφάλαιο 3

Περιγραφή - Ανάλυση Συστήματος

Σε κεφάλαιο αυτό αναφέρονται περιληπτικά όλες οι διαδικασίες που θα εκτελέσει το πρόγραμμα για κάθε διαθέσιμη ανάρτηση στο Facebook, από τη συλλογή των δεδομένων μέχρι και την εξαγωγή των διαφόρων αποτελεσμάτων του.

3.1 Συλλογή δεδομένων

3.1.1 Ανάλυση Facebook DOM

Αρχικά, για να μπορέσουμε να δημιουργήσουμε το web scraper μας είναι σημαντικό να αξιολογήσουμε και να ελέγξουμε πως είναι διαχωρισμένα τα δεδομένα στη σελίδα του Facebook μέσω από το DOM της.

Αν και με τη πρώτη ματιά οι αναρτήσεις στο Facebook φαίνονται ότι είναι ίδιες σε δομή, αξιολογώντας τις παρατηρούμε ότι η Facebook έχει αντιθέτως κατασκευάσει πολλαπλές διαφορετικές δομές DOM. Αυτό συμβαίνει κυρίως γιατί υπάρχουν πολλαπλοί τρόποι να μοιραστεί κάποιος χρήστης μια πληροφορία με τους φίλους του. Όχι μόνο κάποιος χρήστης μπορεί να αναρτήσει υλικό που περιέχει κείμενο, φωτογραφίες η και βίντεο αλλά μπορεί επίσης να μοιραστεί υλικό από άλλες πλατφόρμες σε αυτήν του Facebook. Όλα αυτά και άλλα πολλά που αναφέρθηκαν πιο πάνω ως προς της δυνατότης που παρέχει η Facebook προς τους χρήστες τις, παίζουν τεράστια σημασία για τη δομή DOM της κάθε ανάρτησής.

Για να μπορέσουμε να διαχειριστούμε όσες περισσότερες αναρτήσεις γίνεται, εντοπίσαμε κάποια βασικά elements τα οποία είναι κοινά σχεδόν σε κάθε ανάρτηση. Αξιοποιώντας έτσι αυτά τα κοινά elements ήταν δυνατό να σχεδιάσουμε ένα web scraper ο οποίος μπορεί αποτελεσματικά να εξορύξει τα επιθυμητά δεδομένα από σχεδόν οποιαδήποτε ανάρτηση σε οποιαδήποτε από τις σελίδες που παρέχει η Facebook.

3.1.2 Web Scraping

Με την ανάλυση του DOM που έχουμε δημιουργήσει στη JavaScript το plug-in μας το οποίο όταν ο χρήστης κάνει scroll και καινούργιές αναρτήσεις φορτώνονται από την Facebook ο web scraper μας πραγματοποιεί την εξόρυξη στα επιθυμητά δεδομένα.

Κάθε ανάρτηση που είναι δυνατό για τον web scraper να την εντοπίσει θα εξορισθεί μόνο μία φορά και τα δεδομένα της αποθηκεύονται σε τοπικές μεταβλητές για την μελλοντική επεξεργασία τους.

Τα επιθυμητά δεδομένα από κάθε ανάρτηση που στοχεύει ο web scraper να εξάγει είναι :

- Ποιος έχει ανεβάσει την συγκεκριμένη ανάρτηση
- Το κείμενο, αν έχει γράψει κάτι ο κάτοχος του
- Τον αριθμό των αντιδράσεων και likes
- Αν υπάρχουν σχόλια, ποιος έχει αφήσει το σχόλιο και τι γράφει
- Πότε πραγματοποιήθηκε η ανάρτηση
- Αν η ανάρτηση είναι χορηγούμενη

- Αν είναι χορηγούμενη τότε από ποιον
- Αν περιέχει κάποιο άρθρο το URL του

Κύριοι τρόποι και τεχνικές web scraping για τη συλλογή των δεδομένων χρησιμοποιώντας jQuery:

$("element")[0].innerText;

Με την πιο πάνω εντολή jQuery ζητάμε το περιεχόμενο (innerText) κάποιου element (class, id, tag). Ανάλογα με αυτό που χρειαζόμαστε από το element μπορούμε να αλλάξουμε το ζητούμενο. Κάποια συχνά στοιχεία που ζητάμε από το element είναι τα: .title, .value, .innerHTML.

$("element").each(function (i){ });

Με την πιο πάνω εντολή jQuery ζητάμε να μαζέψουμε όλα τα ίδια αντικείμενα στο DOM της σελίδας που περιέχουν το συγκεκριμένο element που αναζητούμε. Η function αυτή συμπεριφέρεται σαν επαναληπτικός βρόγχος όπου σε κάθε κύκλο το κρατούμενο element που έχουμε στη κατοχή μας είναι το ανάλογο με το αντικείμενο που βρίσκεται στο DOM της σελίδας. Δηλαδή στο δεύτερο κύκλο στην κατοχή μας θα έχουμε το δεύτερο αντικείμενο που βρίσκεται στο DOM της σελίδας με το χαρακτηριστικό "element" ξεκινώντας πάντα από την αρχή του DOM, ασχέτως με το πόσες φορές θα τρέξουμε την ίδια εντολή.

$(this)[0].innerText;

Με την πιο πάνω εντολή μπορούμε να αξιοποιήσουμε το αντικείμενο στην κατοχή σε κάθε κύκλο του βρόγχου. Το χαρακτηριστικό που ζητάμε μπορεί να αλλάξει ανάλογα με αυτό που επιθυμούμε.

$(this).find("element")[0];

Με την πιο πάνω εντολή μπορούμε να βρούμε το παιδί του αντικειμένου που έχουμε στην κατοχή μας. Με αυτό τον τρόπο μετακινούμαστε στη δεντρική δομή DOM της σελίδας, και εκμεταλλευόμαστε τις πληροφορίες που μπορούν να μας παρέχουν.

3.2 Ανάλυση δεδομένων σε πραγματικό χρόνο

3.2.1 Φίλτρο γλώσσας

Αφού έχει πραγματοποιηθεί η εξόρυξη και αποθήκευση των δεδομένων σε τοπικές μεταβλητές, προχωρούμε στην προ επεξεργασία τους ώστε να μπορέσουμε αργότερα να εξάγουμε από αυτά τα σημαντικά τους χαρακτηριστικά.

Για να πραγματοποιήσουμε αυτή την προ-επεξεργασία αποστέλλουμε τα δεδομένα από το client-side plug-in μας στη Node Js πλατφόρμα μας. Εκεί χρησιμοποιώντας διάφορες τεχνικές και πακέτα που έχουμε στην κατοχή μας από την NPM, απαλλάσσουμε από όλα τα κείμενα τα διάφορα σύμβολα τα οποία μπορεί να περιέχουν, αφαιρώντας έτσι των πιθανό θόρυβο που πολύ πιθανό θα προκαλούσαν στην μετάφραση και εξαγωγή συναισθήματος τους.

Χρησιμοποιώντας το NPM πακέτο για Google translate μπορούμε να αναγνωρίσουμε σε τι γλώσσα είναι γραμμένα τα δεδομένα μας και ανάλογα να τα μεταφράσουμε στα αγγλικά.

3.2.2 Εξαγωγή συναισθήματος

Έχοντας απαλλάξει τα δεδομένα από πιθανό θόρυβο και αφού έχουν μεταφραστεί στην αγγλική γλώσσα, το πρόγραμμα μεταφέρνει τα δεδομένα στη επόμενη φάση για την εξαγωγή συναισθήματος.

Χρησιμοποιώντας τη βιβλιοθήκη Natural από την NPM έχουμε την δυνατότητα να εξάγουμε από το κείμενο μας το συναίσθημα. Η βιβλιοθήκη επιστρέφει ως αποτέλεσμα ένα πραγματικό αριθμό στο διάστημα (-3 , 3). Όσο πιο χαμηλή είναι η τιμή που επιστρέφει τόσο και πιο αρνητικό είναι το συναίσθημα που περιέχει η πρότασης, και αντίθετα, όσο πιο μεγάλη τιμή τόσο πιο θετικό το συναίσθημα της πρόταση.

Αυτή η διαδικασία εφαρμόζεται για το κείμενο της ανάρτησης αλλά και για τα σχόλια, και θεωρούμε το μέσο όρο τους ως το συναίσθημα που έχει ολόκληρη η ανάρτηση μαζί.

3.2.3 Έλεγχος εγκυρότητας άρθρου

Πέραν από την συναισθηματική αξιολόγηση, το πρόγραμμα μας σε περίπτωση που εντοπίσει ότι μία ανάρτηση είναι χορηγούμενη και έχει εξαχθεί το από που προέρχεται, εξετάζει την εγκυρότητα της πηγής του. Αυτό πραγματοποιείτε αφού υπάρχει μια μαύρη λίστα στο

πρόγραμμα που έχει καταχωρημένα πολλαπλά Domain μαζί με την περιγραφή που τα χαρακτηρίζει. Το πρόγραμμα αφού έχει τώρα στην κατοχή του την προέλευση της είδησης μπορεί να την διασταυρώσει με αυτές που είναι γνωστός ο βαθμός αξιοπιστίας τους για τυχών αντιστοιχία.

Υπάρχουν πολλαπλές ετικέτες για αυτά τα Domain που βρίσκονται καταχωρημένα στην μαύρη λίστα του προγράμματος και κάποια από αυτά είναι :

- Fake
- Conspiracy
- Biased
- Questionable

3.3 Δομή και αποθήκευση δεδομένων

Αφού έχει πραγματοποιηθεί η εξόρυξη, η προ-επεξεργασία και εξαγωγή των διάφορων συμπερασμάτων όπως την συναισθηματική τιμή και την εγκυρότητα του άρθρου, το πρόγραμμα προχωράει στον να τα αποθηκεύσει όλα αυτά τα δεδομένα και αποτελέσματα σε μια βάση δεδομένων.

Η δομή με την οποία τα δεδομένα αποθηκεύονται στη βάση είναι :

- Ποιος είναι ο δημιουργός της ανάρτησης
- Το κείμενο (αν έχει γράψει κάτι ο κάτοχος του)
- Τον αριθμό των αντιδράσεων και likes
- Αν υπάρχουν σχόλια: ποιος έχει αφήσει το σχόλιο και τι γράφει
- Πότε πραγματοποιήθηκε η ανάρτηση
- Αν η ανάρτηση είναι χορηγούμενη
- Αν είναι χορηγούμενη τότε από ποιον
- Η συναισθηματική τιμή του κείμενου (αν έχει γράψει κάτι ο κάτοχος του)
- Η μέση συναισθηματική τιμή από όλα τα σχόλια
- Η μέση συναισθηματική τιμή για την ανάρτηση
- Η ετικέτα που χαρακτηρίζει την πηγή της είδησης

3.4 Αλληλεπίδραση με χρήστη

3.4.1 Ενημέρωση για συναίσθημα

Το πρόγραμμα μας διαθέτει δυο τρόπους για να επικοινωνήσει με το χρήστη για το συναίσθημα που ανιχνεύει στην αναρτήσεις.

Ο ένας τρόπος πραγματοποιείται ζωντανά εάν το πρόγραμμα εντοπίσει ότι υπάρχει κάποια ανάρτηση που είναι φορτωμένη στο περιεχόμενο που παρακολουθεί ο χρήστης η οποία ξεπερνά κάποιες τιμές. Οι τιμές είναι όταν ο ολικός μέσος όρος της δημοσίευσης ξεπερνά το 1.3 η είναι χαμηλότερος από -1.3.Ο τρόπος που το πρόγραμμα αλληλοεπιδρά με τον χρήστη στη προκειμένη περίπτωση είναι με το να δημιουργήσει ένα χρωματιστό περίγραμμα γύρο από την συγκεκριμένη δημοσίευση . Πράσινο χρώμα για τα θετικά και κόκκινο για τα αρνητικά.

Ο δεύτερος τρόπος που μπορεί να ενημερωθεί ο χρήστης είναι με το να πατήσει το κουμπί “check status” το οποίο θα καλέσει να φορτωθούν από την βάση όλες οι αναρτήσεις. Από αυτές τις αναρτήσεις το πρόγραμμα θα υπολογίσει το ποσοστό του τι ποσοστό τους ήταν θετικές και τι ποσοστό τους ήταν αρνητικές, και μέσω αναδυόμενης ειδοποίησης, ενημερώνει τον χρήστη για αυτές.

3.4.2 Ενημέρωση για fake news

Πέραν της ενημέρωσης για το συναίσθημα, το πρόγραμμα ενημερώνει τον χρήστη για τυχών εντοπισμό άρθρου που βρίσκεται στην κατηγορία των Fake news. Η κατηγορία Fake news αντιπροσωπεύει οποιοδήποτε domain βρίσκεται στην blacklist κατηγορία.

Το πρόγραμμα, σε περίπτωση που διαπιστώσει τέτοιο είδος άρθρα, μέσω χρήσης αναδυόμενης ειδοποίησης ενημερώνει τον χρήστη. Επίσης για να επισημάνει στον χρήστη πια είναι αυτή η συγκεκριμένη ανάρτηση, τροποποιεί το χρώμα του παρασκήνιου από άσπρο σε κόκκινο.

Κεφάλαιο 4

Λεπτομέρειες Υλοποίησης

Αυτό το κεφάλαιο περιέχει στιγμιότυπα του προγράμματος και αναλυτική περιγραφή όλων των διαδικασιών που θα εκτελέσει το πρόγραμμα για κάθε διαθέσιμη ανάρτηση, από τη συλλογή των δεδομένων μέχρι και την εξαγωγή των διαφόρων αποτελεσμάτων του.

4.1 Ανάλυση Facebook DOM

Κάνοντας inspect τη σελίδα ή οποία μας ενδιαφέρει μπορούμε να μελετήσουμε το DOM της και να συγκεντρώσουμε όλες τις απαραίτητες πληροφορίες για την εξόρυξη των δεδομένων που αναζητούμε. Σε αυτή τη διπλωματική εργασία εστιαζόμαστε στη εξόρυξη δεδομένων του Facebook και στις παρακάτω εικόνες θα αναλύσουμε κάποια από τα κύρια DOM elements που χρησιμοποιούνται από το Facebook για να παρουσιάσουν διάφορα είδη δημοσιεύσεων στο News Feed άλλα και σε άλλες σελίδες.

- Η πιο κύρια δομή DOM που χρησιμοποιείται είναι αυτή της απλής δημοσίευσης, από οποιονδήποτε χρήστη η σελίδα, ή οποία αυτή δημοσίευση είναι μοναδική δηλαδή δεν είναι κοινή και από άλλους χρήστες.

Η κλάση “userContentWrapper” απευθύνεται στο div element το οποίο είναι μια ολοκληρωμένη δημοσίευση κάποιου χρήστη. Το div αυτό είναι γονέας για άλλα δύο div. Ακολουθώντας το πρώτο child div με κλάση “_1dwg _1w_n _q7o” θα μπορέσουμε να βρούμε τα δικά του παιδιά που θα μπορέσουν να μας παρέχουν διάφορες πληροφορίες για τη δημοσίευση όπως: ποιος , πότε , πού , πώς νιώθει και το περιεχόμενο της δημοσίευσης (κείμενο, εικόνα, βίντεο, και το συνδυασμό τους) . Ενώ το δεύτερο child div περιέχει ένα element form με κλάση “commentable_item” το οποίο μέσω από τα δικά του παιδιά μπορεί να μας παρέχει διάφορες πληροφορίες για την αλληλεπίδραση της δημοσίευσης από άλλους χρήστες όπως likes και σχόλια.

- Δομή DOM δημοσίευσης η οποία έχει δημοσιευτεί από περισσότερο από ένα χρήστη. Αυτή η δημοσίευση εμφανίζεται στο News Feed όταν πολλαπλοί χρήστες δημοσιεύουν το ίδιο περιεχόμενο (π.χ. κάποιο άρθρο).

Η κλάση “_4-u2 nbm _4mrt _5v3q 7cqq _4-u8” είναι ο γονέας της δημοσίευσης και χωρίζει το περιεχόμενο του στα εγγόνια του που βρίσκονται εντός του παιδιού του με κλάση “_5pcr clearfix”. Το εγγόνι με κλάση “_3-8j” περιέχει την κοινή δημοσίευση και τα χαρακτηριστικά όπως ποιοι από τους φίλους του χρήστη έχουν κάνει τη δημοσίευση , τι είναι αυτή η δημοσίευση, το περιεχόμενο της, η ημερομηνία ανάρτησης της και ποιος είναι ο δημιουργός της. Ενώ το δεύτερο εγγόνι με κλάση “uiListuiCollapsedList...” μπορούμε να παρατηρήσουμε ότι είναι λίστα και αποτελείται από πολλά divs. Τα divs αυτά μπορούμε να παρατηρήσουμε ότι είναι τα ίδια με αυτά της απλής δημοσίευσης που αναλύσαμε παραπάνω.

- Δημοσιεύσεις οι οποίες είναι χορηγούμενες (sponsored) και έχουν likes από φίλους του χρήστη αλλά και δημοσιεύσεις που έχουν ετικέτα κάποιου φίλου του χρήστη, έχουν διαφορετική δομή DOM από τις απλές δημοσιεύσεις

Σε αυτή τη δομή DOM μπορούμε να παρατηρήσουμε ότι ο γονέας είναι το div με κλάση “userContentWrapper” ή οποία κλάση είναι η ίδια με αυτή των απλών δημοσιεύσεων. Το πρόβλημα εδώ είναι ότι αυτός ο γονέας δεν περιέχει ο ίδιος το περιεχόμενο, αλλά έχει ένα παιδί το οποίο έχει ως περιεχόμενο ποιοι φίλοι έχουν κάνει like ή βρίσκονται σε ετικέτα, ένα δεύτερο παιδί το οποίο έχει την ίδια κλάση με των γονέα του και με τον γονέα στις απλές δημοσιεύσεις, και περιέχει όντως όλο το περιεχόμενο της δημοσίευσης. Αυτό μας

προβληματίζει για χρειάζεται να προσθέσουμε κώδικα με των οποίο θα κάνουμε έλεγχο αν το αντικείμενο με κλάση “userContentWrapper” είναι γονέας.

4.2 Web Scraping

Όταν το chrome plug-in που δημιουργήσαμε είναι ενεργοποιημένο, οι Listeners είναι σε ετοιμότητα για να καλέσουν την function sentimentPost για συλλογή δεδομένων της σελίδας του Facebook με web scraper, σε οποιαδήποτε από τις δύο περιπτώσεις:

1. Όταν ο χρήστης συνδεθεί στο Facebook
2. Κάθε φορά που ο χρήστης κάνει scroll το ποντίκι του για να φορτώσει περισσότερο περιεχόμενο

Όταν η λειτουργεία sentimentPost καλεστεί, χρησιμοποιώντας την jQuery μαζεύουμε όλα τα posts που είναι φορτωμένα την παρούσα στιγμή στο Facebook του χρήστη με την παρακάτω εντολή και αποθηκεύονται σε πίνακα με όνομα $post .

```
function get_post_set (){
    return new Promise((resolve,reject)=>{

        $post = [];
        $('.userContentWrapper').each(function(i){
            $post[i] = $(this);
        });

         resolve($post);
    });
}
```

Χρησιμοποιώντας μια καθολική μεταβλητή που ενημερώνεται συνεχώς με το πλήθος των διαθέσιμων post μπορούμε άμεσα να επικεντρωθούμε στα καινούργια post που έχουμε αποκτήσει.

Συνεχίζουμε για να μαζέψουμε όλα τα στοιχεία από το post με το να καλέσουμε όλες τις λειτουργείες που φαίνονται πιο κάτω, όπου και αυτές με την σειρά τους, χρησιμοποιώντας την jQuery εξάγουν το ανάλογο περιεχόμενο από την δημοσίευση την οποία αναζητάμε.

```
post_who_posted[i] = await get_post_who_posted(post[i]);
post_body[i] = await get_post_body(post[i]);
post_likes[i] = await get_post_likes(post[i]);
post_who_comments[i] = await get_post_who_comments(post[i]);
post_comments[i] = await get_post_comments(post[i])
var temp = await get_post_timestamp(post[i]);
```

Η κάθε λειτουργία καλεί ανάλογα το αντικείμενο με βάση την κλάση η τις κλάσεις που έχει, και κυρίως ζητάμε από αυτά το innerText, με εξαίρεση την λειτουργία που αναζητάμε το URL του άρθρου. Στην περίπτωση που έχουμε εντοπίσει ότι κάποια δημοσίευση περιέχει κάποιο άρθρο, τότε αναζητάμε από αυτό το URL μέσω της λειτουργείας get_post_url(post[i]) η οποία σε αντίθεση με τις άλλες λειτουργείες που αναζητούν για το innerText, αυτή αναζητά για το attributes[0].nodeValue.

```
function get_post_timestamp(post){
    return new Promise((resolve ,reject)=>{

        var timestamp = "";

        try{
           timestamp = $(post).find("._5pcp._5le1._2jyu._232_")[0].innerText;
           // console.log(timestamp);
        }catch (error){
        }
        resolve(timestamp)
    });
}

function get_post_url(post){
    return new Promise((resolve , reject)=>{

        try{
            if($(post).find("._6ks")[0]){
                var url = $($(post).find("._6ks")[0]).find("a")[0].attributes[0].nodeValue;

                url = url.replace(/^(https\:\/\/l\.facebook\.com\/l\.php\?u\=)/, "");
                url = url.replace("https%3A%2F%2F", "");
                url = url.replace("http%3A%2F%2F", "");
                url = url.substring(0 , url.indexOf("%2"));

                resolve(url);
            }
        }catch (error){
            resolve(" ");
        }
    });
}
```

Από τίς πιο πάνω εικόνες μπορούμε να παρατηρήσουμε ότι η συλλογή και εξόρυξη των δεδομένων γίνετε σειριακά χρησιμοποιώντας τεχνικές όπως είναι το async/await και promise.

Η σειριακή εκτέλεση του κώδικα στην συλλογή των δεδομένων είναι απαραίτητη για τον λόγο ότι κάποια αντικείμενα τα οποία προσπαθούμε να αποκτήσουμε έχουν το ρόλο του παιδιού και άλλα του γονέα. Γνωρίζοντας το αυτό και ότι το DOM παρομοιάζετε με δέντρο,

ξέρουμε ότι είναι απαραίτητο να έχουμε σίγουρα εξασφαλίσει τα αντικείμενα τα οποία έχουν το ρόλο του γονέα προτού προχωρήσουμε στο να αναζητήσουμε αντικείμενα που είναι παιδιά τους.

Το πιο σημαντικό παράδειγμα αυτής της περίπτωσης είναι το κάλεσμα της λειτουργίας που μαζεύει όλες τις δημοσιεύσεις και τις αποθηκεύει σε πίνακα. Αν αυτή δεν είχε ολοκληρωθεί, τότε δεν θα μπορούσαμε να προχωρήσουμε στην αναζήτηση των υπόλοιπων στοιχείων.

Επίσης από τις εικόνες μπορούμε να παρατηρήσουμε ότι χρησιμοποιούμε την τεχνική try/catch για την συλλογή δεδομένων. Εντός των αγκυλών του try/catch επιχειρούμε να αποκτήσουμε τα δεδομένα και σε περίπτωση αποτυχίας επιστρέφουμε πίσω στο πρόγραμμα κενό (“ “). Είναι απαραίτητο αυτό γιατί η χρήση των τεχνικών async/wait και promise αποτρέπουν το πρόγραμμα από το να συνεχίσει την ομαλή λειτουργεία του αν αυτά δεν επιστρέψουν κάποια τιμή και για αυτό τον λόγο έχουμε προσθέσει την επιστροφή του κενού.

Το πρόγραμμα μπορεί να αποτύχει για διάφορες περιπτώσεις να συλλέξει τα δεδομένα τα οποία του ζητάμε. Ένας σημαντικός λόγος για τον οποίο το πρόγραμμα μπορεί να αποτύχει είναι όταν τυχών βρεθούμε αντιμέτωποι με καινούργια δομή DOM για την οποία το πρόγραμμα μας δεν είναι σε ετοιμότητα να εξάγει τις πληροφορίες του σωστά, αφού όπως αναφέραμε προηγουμένως το Facebook έχει πολλαπλές δομές DOM για να παρουσιάσει τα δεδομένα του. Έτσι, παρόλο που η τεχνική που χρησιμοποιούμε μπορεί να αντιμετωπίσει το μεγαλύτερο κομμάτι από δημοσιεύσεις, υπάρχουν ακόμη δημοσιεύσεις για τις οποίες το πρόγραμμα μας αδυνατεί να εκτελεστεί.

4.3 Ανάλυση δεδομένων

4.3.1 Φίλτρο γλώσσας

Αφού έχουμε αποκτήσει τα δεδομένα προχωράμε στην επεξεργασία τους με στόχο μετέπειτα να μπορέσουμε να εξάγουμε από αυτά το συναίσθημα αν είναι κείμενο, ή αν είναι URL κάποιου άρθρου να μπορέσουμε να κρίνουμε κατά πόσο αυτό είναι αξιόπιστο ή αναξιόπιστο.

Στην περίπτωση όπου έχουμε εξάγει το URL κάποιου άρθρου αυτό πρέπει να περάσει από μια σειρά αλλαγών οι οποίες φαίνονται πιο στην εικόνα πιο κάτω.

```
url = url.replace(/^(https\:\/\/l\.facebook\.com\/l\.php\?u\=)/, "");
url = url.replace("https%3A%2F%2F", "");
url = url.replace("http%3A%2F%2F", "");
url = url.substring(0 , url.indexOf("%2"));
```

Αρχικά το URL περιέχει διάφορα σύμβολα και κείμενο τα οποία κρύβουν και μας αποτρέπουν την άμεση πρόσβαση στο Domain, αλλά με τις παραπάνω εντολές αφαιρούμε το μη επιθυμητά κομμάτια και αφήνουμε πίσω μόνο αυτό που χρειαζόμαστε για τον έλεγχο εγκυρότητας.

Για το κείμενο όμως που εξάγουμε από τις δημοσιεύσεις και τα σχόλια της κάθε δημοσίευσης, χρειάζεται μεγαλύτερη επεξεργασία λόγω του ότι πέραν του ότι και τα σχόλια μπορούν να περιέχουν σύμβολα ή μπορούν να είναι γραμμένα σε οποιαδήποτε γλώσσα ή ακόμη και σε γλώσσα η οποία δεν είναι πραγματική, όπως είναι και τα Greeklish. Για να μπορέσουμε να χειριστούμε αυτές τις πιο περίπλοκές διαδικασίες τις οποίες θα χρειαστούμε να εκτελέσουμε για την μετατροπή του κειμένου το αποστέλλουμε στην Node Js πλατφόρμα που έχουμε δημιουργήσει. Στην Node Js με την χρήση της τεράστια βιβλιοθήκης με πακέτα NPM που είχαμε αναφέρει στο δεύτερο κεφάλαιο έχουμε την δυνατότητα για να απλοποιήσουμε το κείμενο μας.

Αρχικά απαλλάσσουμε το κείμενο από τα διάφορα σύμβολα το οποία μπορεί να περιέχει με την ακόλουθο κώδικα που φαίνεται στην εικόνα.

```
//If the body value is not empty
if(body != ''){

    //Remove all the symbols from the text
    body_modified = body.replace(/[&!\/\\#,+()$~%.'":*?<>{}]/g, '');
```

Και αργότερα προχωρούμε στην αναγνώριση της γλώσσας και μετάφραση του περιεχομένου χρησιμοποιώντας την βιβλιοθήκη **@vitalets/google-translate-api**. Μέσω αυτή βιβλιοθήκης μπορούμε να αποστέλλουμε το κείμενο μας στην google και να χρησιμοποιήσουμε μία από τις πιο δημοφιλές λειτουργείες της, το google translate. Το google translate δεν μας επιτρέπει μόνο να μεταφράσουμε κείμενο από μία γλώσσα που είδη γνωρίζουμε σε μια άλλη γλώσσα της επιλογής μας, αλλά μπορεί μέσω των τεχνολογιών της και το deep learning NMT δίκτυο της, να αναγνωρίσει την γλώσσα του περιεχομένου μας από μόνο του και μετά να το μεταφράσει στην επιθυμητή γλώσσα που επιθυμούμε. Αυτό μας επιτρέπει πολύ απλά να εκτελέσουμε την παρακάτω εντολή χωρίς να γνωρίζουμε πραγματικά σε ποια γλώσσα είναι το κείμενο μας.

```
//Translate the text no matter the language in english
translate(body_modified, {to: 'en'}).then(res => {
```

Τα παραπάνω βήματα επαναλαμβάνονται για όλα τα σχόλια που πιθανό να υπάρχουν ξεχωριστά.

Το τελικό βήμα πριν την εξαγωγή του συναισθήματος είναι η χρήση tokenizer για την μετατροπή του κειμένου μας από μία μεταβλητή σε μορφή string σε πίνακα όπου σε κάθε

κελί του, αποθηκεύεται και μία από τις λέξεις. Αυτό γίνεται γίνετε για το λόγο ότι η μέθοδος που χρησιμοποιούμε για την εξαγωγή του συναισθήματος παίρνει σαν είσοδο ένα τέτοιο πίνακα.

4.3.2 Εξαγωγή συναισθήματος

Για την εξαγωγή του συναισθήματος από το κείμενο αφού έχουμε ολοκληρώσει όλη την απαραίτητη προ επεξεργασία τους, χρησιμοποιούμε την βιβλιοθήκη Natural. Για την χρήση της βιβλιοθήκης αρχικά εκτελούμε τις παρακάτω εντολές για την αρχικοποίηση των αντικειμένων τα οποία θα χρησιμοποιήσουμε.

```
var Analyzer = natural.SentimentAnalyzer;
var stemmer = natural.PorterStemmer;
var analyzer = new Analyzer("English", stemmer, "afinn");
```

Ο Stemmer έχει αρχικοποιηθεί με τον PorterStemmer τον οποίο η βιβλιοθήκη natural μας παρέχει. Η χρήση του είναι για να επιστρέφει τις λέξεις που θα το δοθούν στην κανονική τους μορφή, για παράδειγμα αν στην πρόταση μας είχαμε την λέξη “was” ο Stemmer θα μας επιστρέψει πίσω την λέξη “be”, η αν είχαμε την λέξη “words” θα μας επιστρέψει την λέξη “word”.

Η παράμετρος “afinn” που δίνουμε για την αρχικοποίηση του analyzer μας είναι λεξικό το οποίο μας παρέχει η βιβλιοθήκη natural το οποίο έχει αποθηκευμένες αγγλικές λέξεις μαζί με την συναισθηματική τους τιμή.

Έτσι ο Stemmer μαζί με την επιλογή της αγγλικής γλώσσας και το “afinn” εφαρμόζονται στην αρχικοποίηση του Analyzer τον οποίου analyzer θα χρησιμοποιήσουμε για να εξάγουμε το συναίσθημα από της προτάσεις μας. Μέσω του analyzer έχουμε πρόσβαση στην λειτουργία getSentiment η οποία επεξεργάζεται όλες τις λέξεις που αποτελούν την πρόταση μας και εξάγει την συναισθηματική τιμή η οποία την αντιπροσωπεύει. Η χρήση του analyzer παρουσιάζετε στην παρακάτω εικόνα.

```
//Get sentiment value for the body text
sentimentBody = analyzer.getSentiment(tokenizer.tokenize(res.text));
```

Τα παραπάνω βήμα για την εξόρυξή του συναισθήματος επαναλαμβάνεται για όλα τα σχόλια που πιθανό να υπάρχουν ξεχωριστά και αποθηκεύονται σε μονοδιάστατό πίνακα στην αντιπροσωπευτική τους θέση.

Έχοντας ολοκληρώσει την εξαγωγή του συναισθήματος για το κείμενο και από τα πιθανά σχόλια που μπορεί να έχει μια δημοσίευση υπολογίζουμε τον μέσο όρο όλων τους και αποθηκεύεται σε μια νέα μεταβλητή η οποία αντιπροσωπεύει το ολικό συναίσθημα της συγκεκριμένης δημοσίευσης. Αυτή η αντιπροσωπευτική τιμή για το συναίσθημα της δημοσίευσης επιστέφεται πίσω στο plug-in μας, όπου με βάση την τιμή αυτή αλληλοεπιδρά με τον χρήστη ανάλογα.

4.3.3 Εγκυρότητα άρθρου

Για να κριθεί αν κάποιο άρθρο είναι έγκυρο η όχι έχουμε δημιουργήσει μια Json μεταβλητή (Blacklist) η οποία είναι σε άμεση πρόσβαση από το plug-in μας. Η Blacklist μεταβλητή περιέχει περισσότερο από 1500 Domains με τις ανάλογες ετικέτες που τα χαρακτηρίζουν. Τα 1500 Domains και οι ετικέτες όμως είναι μόνο για κακόβουλο περιεχόμενο, έτσι όταν κάνουμε αναζήτηση για κάποιο συγκεκριμένο άρθρο αν υπάρχει το URL προέλευσης τους εντός της Blacklist μεταβλητής και αποτύχουμε τότε αυτόματα θεωρούμε το άρθρο ως ασφαλές.

Είναι αφελής από εμάς να πιστεύουμε ότι έχουμε μαζέψει όλα τα Domains τα οποία είναι γνωστά για το κακόβουλο περιεχόμενο τους έτσι και γνωρίζουμε ότι η μέθοδος που χρησιμοποιούμε θε επιστέφει σε κάποιες περιπτώσεις false positives το οποίο σημαίνει ότι λανθασμένα θα τα θεωρεί ότι είναι αληθές ενώ στην πραγματικότητα δεν είναι. Αυτό δεν είναι κάτι που μας επηρεάζει όμως, για το λόγο ότι το πλήθος των fake news ειδήσεων είναι πολύ μικρότερος ο αριθμός τους από τις πραγματικές ειδήσεις πόσο μάλλον αυτές που είναι fake και δεν περιλαμβάνονται στην δική μας λίστα.

Για να αναζητήσουμε την Json μεταβλητή και να ελέγξουμε κατά πόσο είναι fake το συγκεκριμένο άρθρο, πραγματοποιείτε με των ακόλουθο κώδικα.

```
if(blacklist[post_url[i]]!=null){

    post_url_label[i] = blacklist[post_url[i]];
```

Αφού είναι Json η μεταβλητή που περιέχει τα URL και τις ετικέτες μπορούμε να αναζητήσουμε την ετικέτα του συγκεκριμένου άρθρου χρησιμοποιώντας το URL του ως την θέση του Json πίνακα και λαμβάνοντας πίσω ως αποτέλεσμα την ετικέτα. Στην παραπάνω εικόνα μπορούμε να παρατηρήσουμε ότι προτού προσπαθούμε να αποθηκεύσουν την ετικέτα στα χαρακτηριστικά της δημοσίευσης μας, πραγματοποιούμε έλεγχο για το κατά πόσο υπάρχει αυτό το URL εντός της λίστας. Αυτό συμβαίνει για να αποτρέψουμε μελλοντικά προβλήματα στο πρόγραμμα με λάθος μεταχείρισή null μεταβλητών αφού αν δεν υπάρχει το URL του άρθρου στην Blacklist τότε αυτό θα μας επιστρέψει null.

4.4 Αποθήκευση δεδομένων

Αφού έχουμε ολοκληρώσει τις διαδικασίες για την εξαγωγή του συναισθήματος και τον έλεγχο για την εγκυρότητα του άρθρου, προχωράμε στην αποθήκευση τον στοιχείων και των αποτελεσμάτων στην βάση μας.

Για την αποθήκευση των δεδομένων έχουμε επιλέξει την χρήση της NoSQL βάσης MongoDB. Η επιλογή έγινε για τα πλεονεκτήματα που μπορεί μια NoSQL βάση να μας παρέχει όπως είναι η αποθήκευση επιπρόσθετων στοιχείων χωρίς να αλλάξει το σχήμα.

Η δημιουργία του δικού μας σχήματος για την αποθήκευση των δεδομένων έχει πραγματοποιηθεί με την χρήση της βιβλιοθήκης mongoose από την NPM. Έχει δημιουργηθεί JavaScript αρχείο "submission.js" το οποίο αξιοποιεί αυτή την βιβλιοθήκη και αποτελεί αντικείμενο για την δημιουργία submissions μέσω της αναφοράς του από την Node Js πλατφόρμα μας. Η μορφή του αρχείου παρουσιάζεται στην πιο κάτω εικόνα.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const SubmissionSchema = new Schema({
    who_posted: {
        type: String,
        required: false
    },
    body: {
        type: String,
        required: false
    },
    likes: {
        type: String,
        required: false
    },
    who_commented:[{
        type: String,
        required: false
     }],
```

Το αντικείμενο αποτελείται από πολλαπλές δημόσιες μεταβλητές όπως και αυτές που παρουσιάζονται στην πιο πάνω εικόνα οι οποίες μέσω της δημιουργίας ενός submissions αντικειμένου είναι προσπελάσιμες.

Η Node Js πλατφόρμα μας δημιουργεί μεταβλητές τύπου "submission" και μέσω αυτού γεμίζει όλες τις μεταβλητές με την ανάλογη τιμή που έχουμε ήδη υπολογίσει. Στην παρακάτω εικόνα μπορούμε να δούμε όλα τα στοιχεία με ξεκάθαρη ονομασία που θα αποθηκευτούν στην βάση και το τι θα περιέχουν.

```
//Prepare to store the data in database
var submission = new Submission();
submission.who_posted = who_posted;
submission.body = body;
submission.likes = likes;
submission.sponsored = sponsored;
submission.timestamp = timestamp;
submission.total_sentiment = sentimentTotal.toString();
submission.body_sentiment = sentimentBody.toString();
submission.total_comments_sentiment = sentimentCommentsTotal.toString();
submission.url = url;
submission.label = label;

if(who_commented != null)
    for(var j=0; j<who_commented.length; j++)
        submission.who_commented.push(who_commented[i]);

if(comments != null)
    for(var j=0; j<comments.length; j++)
        submission.comments.push(comments[i]);

//Store the post
submission.save(function (err) {});
```

Σε κάθε περίπτωση όπου δεν υπήρχε στην συγκεκριμένη δημοσίευση κάποια συγκεκριμένη μεταβλητή τότε στην θέση του στον πίνακα θα πάρει την τιμή του κενού (“ “). Με την εξαίρεση για την μεταβλητή sponsored που αποθηκεύει αν μια δημοσίευση είναι χορηγούμενη και παίρνει τιμή αληθές ή ψευδής (true / false).

Επιπρόσθετα πρέπει να αναφέρουμε ότι κάποιες τιμές όπως είναι η τιμή για το συναίσθημα εξαρτώνται από λειτουργείες οι οποίες χρειάζονται κάποιο μικρό χρονικό διάστημα για να υπολογιστούν. Αυτό αποτελεί πρόβλημα το οποίο πρέπει να διαχειριστούμε, αφού γνωρίζουμε ότι η Node Js λειτουργεί ασύγχρονα. Με την ασύγχρονη λειτουργία της Node Js αν δεν υπάρχει κάποια καθυστέρηση πριν την αποθήκευση των δεδομένων, αυτά που εξαρτώνται από μεταβλητές που χρειάζονται κάποιο χρόνο να υπολογιστούν θα παίρνουν τις περισσότερες φορές την τιμή του κενού. Για αυτόν τον λόγο όπως φαίνεται και πιο κάτω στη εικόνα, πριν την αποθήκευση των δεδομένων χρησιμοποιώντας την τεχνική του “setTimeout”, παγοποιώντας το πρόγραμμα για 4 δευτερόλεπτα έτσι ώστε να είμαστε σίγουροι ότι όλες οι μεταβλητές έχουν πάρει την σωστή τιμή τους.

```
setTimeout(function(){
}, 4000);
```

4.5 Αλληλεπίδραση με χρήστη

Η αλληλεπίδραση με τον χρήστη όπως αναφέραμε προηγουμένως χωρίζεται σε 2 τομείς, στην ενημέρωση του χρήστη για το συναίσθημα το οποίο οι δημοσιεύσεις περιέχουν και για εντοπισμό fake news.

Η ενημέρωση του χρήστη για το συναίσθημα το οποίο περιέχουν οι δημοσιεύσεις γίνεται μέσω της τροποποίησης του στυλ για την συγκεκριμένη δημοσίευση. Αφού έχουμε εξάγει και επεξεργαστεί το περιεχόμενο μίας δημοσίευσης, έχουμε ως αποτέλεσμα την τιμή για το συναίσθημα, η οποία κυμαίνεται μεταξύ -3 και 3. Εφόσον η τιμή δεν είναι κοντά στο 0, δηλαδή ουδέτερο συναίσθημα αλλά αντιθέτως είναι μικρότερη του -1.3 η μεγαλύτερη του 1.3 τότε αλληλοεπιδράμε με τον χρήστη με το να προσθέσουμε χρωματιστό περίγραμμα γύρο από την δημοσίευση. Στην περίπτωση που το συναίσθημα είναι θετικό το χρώμα είναι πράσινο αλλιώς κόκκινο.

Η υλοποίηση αυτή γίνετε μέσω της χρήσης της jQuery και τις λειτουργείες που μας παρέχει. Για το συγκεκριμένο στόχο στοχεύουμε να τροποποιήσουμε το CSS το οποίο είναι υπεύθυνο για το στυλ της σελίδας και των δημοσιεύσεων. Παρακάτω παρουσιάζεται εικόνα με κομμάτι κώδικα για το πώς αυτό είναι εφικτό.

```
if(sentiment[i] >= 1.3)
    post[i].css('border', '3px solid green');
else
    if(sentiment[i] <= -1.3)
        post[i].css('border', '3px solid red');
```

Με αυτό τον τρόπο είναι εφικτό να πραγματοποιήσουμε την αλλαγή στο CSS στην δημοσίευση και να ενημερώσουμε τον χρήστη με το ανάλογο χρώμα εάν η δημοσίευση περιέχει ευχάριστο ή δυσάρεστο κείμενο.

Πέραν από το χρωματιστό πλαίσιο, ο χρήστη μπορεί μέσω του πατήματος ενός κουμπιού να ενημερωθεί κατά πόσο, στατιστικά οι δημοσιεύσεις οι οποίες εμφανίζονται στην οθόνη του περιέχουν αρνητικό συναίσθημα. Όταν ο χρήστης πατήσει το κουμπί “check_status” το οποίο παρουσιάζεται στην εικόνα πιο κάτω το plug-in μέσω ajax call ζητάει από την Node Js πλατφόρμα αυτήν την στατιστική ενημέρωση.

Η Node Js πλατφόρμα έχει πρόσβαση στην βάση με όλα τα δεδομένα και αυτό ζητά από αυτήν όλο το περιεχόμενο που έχει μαζέψει για τον συγκεκριμένο χρήστη. Όταν αποκτήσει όλο το περιεχόμενο κάνει έλεγχο ποια από αυτά έχουν τιμή μικρότερη από το -1.3 και βάση αυτού του αριθμού και το ολικό πλήθος των αποθηκευμένων δημοσιεύσεων η πλατφόρμα ετοιμάζει το κείμενο το οποίο περιέχει αυτό το ποσοστό και το αποστέλλει πίσω στο plug-in μας όπως παρουσιάζεται και στην παρακάτω εικόνα.

```
db.collection('submissions').find({}, function (findErr, result) {

  if (findErr) throw findErr;
  gotem = result;

  result.forEach(function(item){

      if(item.body_sentiment<=-1.5)
          total_negative_posts++;
      total_posts++;
  });

  setTimeout(function(){
      percentage = total_negative_posts*100/total_posts;
      percentage = percentage.toString();
      total_posts = total_posts.toString();
      total_negative_posts = total_negative_posts.toString();
      message = "Out of the "+total_posts+" of posts scanned "+total_negative_posts+" were negative, a "+percentage+"%";
      res.end(message);
  },5000)
```

Όταν το plug-in έχει το κείμενο αυτό μέσω της χρήσης του alert της JavaScript παρουσιάζουμε το μήνυμα αυτό στον χρήστη.

Όσο για την ενημέρωση του χρήστη για εντοπισμό fake news η στρατηγική που χρησιμοποιούμε για να ενημερώσουμε των χρήστη είναι παρόμοια. Όταν το plug-in εντοπίσει το URL κάποιου άρθρου ότι βρίσκεται εντός του Blacklist αυτόματα το πρόγραμμα ενημερώνει των χρήστη μέσω της χρήσης του alert με μήνυμα ότι εντοπίστηκε fake news. Επιπρόσθετα ενημερώνει των χρήστη ότι η συγκεκριμένη δημοσίευση με το fake news άρθρο τροποποιήθηκε με κόκκινο χρώμα ως φόντο. Αυτό πραγματοποιείται όπως και προηγουμένως με την χρήση της jQuery για να τροποποιήσουμε το CSS της συγκεκριμένης δημοσίευσης.

Κεφάλαιο 5

Αξιολόγηση Συστήματος

Σε αυτό το κεφάλαιο υπάρχει μια αξιολόγηση των μεθόδων που χρησιμοποιήθηκαν για τη συλλογή δεδομένων, επεξεργασία δεδομένων και εξαγωγή συναισθήματος ως προς την αξιοπιστία, την ταχύτητα και την αποδοτικότητά τους.

5.1 Αξιολόγηση Web Scraping

Το Web Scraping plug-in μας φαίνεται να επιτυγχάνει το βασικό σκοπό του στο να εξάγει την πληροφορία από τις δημοσιεύσεις και να την προωθεί στου έξυπνους αλγόριθμους για να εξάγουν το συναίσθημα που περιέχουν.

Με την δημιουργία αυτού του scraper έχουμε την δυνατότητα να ενημερώσουμε το χρήστη για το συναίσθημα του περιεχομένου, και την εγκυρότητα των άρθρων που του παρουσιάζονται. Ακόμη έχουμε την δυνατότητα με την συνεχή συλλογή των δεδομένων του χρήστη να εξάγουμε στατιστικά συμπεράσματα τα οποία θα αντιπροσωπεύουν το είδος των αναρτήσεων για τα οποία ο συγκεκριμένος χρήστης συνεχώς ενημερώνεται.

Επειδή όμως αυτή η ενέργεια συλλογής δεδομένων ενεργοποιείται με το scrolling του ποντικιού και το φόρτωμα καινούργιων δημοσιεύσεων, η συλλογή δεδομένων είναι αργή και εξαρτάται απόλυτα από τον αριθμό των χρηστών του plug-in αλλά και την επιθυμία τους να εξερευνήσουν καινούργιο περιεχόμενο.

Ο μικρός αυτός όγκος δεδομένων που έχουμε στην κατοχή μας, μας αποτρέπει να εξάγουμε έγκυρα στατιστικά συμπεράσματα για το συναίσθημα των δημοσιεύσεων και την εγκυρότητα των άρθρων για όλη την πλατφόρμα του Facebook.

Αν ένας μεγάλος αριθμός χρηστών της πλατφόρμας του Facebook χρησιμοποιούσε και το plug-in που δημιουργήσαμε σε αυτή την ατομική διπλωματική εργασία τότε αυτό δεν θα ήταν πλέον πρόβλημα. Με τον μεγάλο αριθμό χρηστών θα μπορούσαμε να εξάγουμε αντιπροσωπευτικά συμπεράσματα για την πλατφόρμα του Facebook, για το συναίσθημα των αναρτήσεων και την εγκυρότητα των άρθρων της.

5.2 Αξιολόγηση φίλτρων

Για να εξάγουμε την πληροφορία από τις δημοσιεύσεις υπάρχουν διάφορα φίλτρα που πρέπει τα στοιχεία αυτά από τις δημοσιεύσεις να περάσουν με σκοπό να μπορούμε να εξάγουμε την ανάλογη πληροφορία τους.

Κάποια από αυτά τα φίλτρα έχουν ως σκοπό να απαλλάξουν το κείμενο και τα σχόλια από σύμβολα, ορθογραφικά λάθη και να μεταφραστούν στην αγγλική γλώσσα αλλά και να μπορέσουμε να εξάγουμε το Domain ενός άρθρου από το ασήμαντο κείμενο που περικυκλώνει το URL του.

Αυτά τα φίλτρα παρόλο που είναι πολύ αποτελεσματικά δεν μπορούν να ανταπεξέλθουν και να ετοιμάσουν πάντοτε με 100% ακρίβεια το περιεχόμενο έτσι ώστε ο αλγόριθμος να μπορεί να εξάγει το συναίσθημα με ακρίβεια. Το αυτόματο φίλτρο που μας προσφέρει η Google για να διορθώσει τα ορθογραφικά λάθη που μπορεί να υπάρχουν στο κείμενο αδυνατεί να ολοκληρωθεί με ακρίβεια. Αυτό συμβαίνει όταν το κείμενο έχει τόσο έντονα ορθογραφικά λάθη που πλέον οι λέξεις δεν είναι αναγνωρίσιμες από την μηχανή.

Στην περίπτωση όπου η σωστή μετάφραση αποτύχει να ολοκληρωθεί με ακρίβεια τότε είναι αναπόφευκτο ότι και ο αλγόριθμος που θα εξάγει το συναίσθημα δεν μπορεί να υπολογίσει με ακρίβεια το συναίσθημα που περιέχει η πρόταση.

Πέραν από το κείμενο και τα σχόλια που περνάνε από διάφορα φίλτρα με στόχο να εξάγουμε το συναίσθημα τους, έχουμε παρατηρήσει πως το στοιχείο που περιέχει τον δημιουργό της ανάρτησης χρειάζεται περισσότερη επεξεργασία μέσω φίλτρων ώστε να μπορεί να μας παρέχει την πληροφορία που χρειαζόμαστε.

Έχουμε παρατηρήσει πως η δομή για την παρουσίαση του χρήστη που έχει δημιουργήσει την ανάρτηση δεν είναι σταθερή. Το κείμενο που περιγράφει ποιος είναι ο δημιουργός μπορεί να περιέχει :

- Όνομα ή Ονοματεπώνυμο χρήστη
- Όνομα χρήστη ή χρηστών με συναισθηματική κατάσταση
- Όνομα χρήστη ή χρηστών με την τοποθεσία τους
- Όνομα χρήστη ή χρηστών που αντέδρασαν, σχολίασαν η τους άρεσε κάποια δημοσίευση
- Όνομα χρήστη και την σελίδα που έχει δημιουργήσει την ανάρτηση
- Όνομα διαφημιστή

Δεν είναι απαραίτητα αρνητικό αυτό, γιατί μέσω της σωστής διαχείρισης του περιεχομένου η πληροφορία που θα μπορούμε να εξάγουμε είναι μεγαλύτερη από αυτή του απλού ονόματος του χρήστη.

Το πρόβλημα είναι πως χρειάζεται να δημιουργηθούν πολλαπλά φίλτρα που θα μπορούν να διαχειριστούν και να ξεχωρίσουν όλες τις παραπάνω περιπτώσεις και άλλες πιθανές που μπορεί να αντιμετωπιστούν προτού προχωρήσουν για αποθήκευση.

5.3 Αξιολόγηση εξαγωγής συναισθήματος

Η εξαγωγή συναισθήματος μέσω της χρήση της βιβλιοθήκης natural αν και είναι σχετικά ακριβής όταν η ορθογραφία του κειμένου είναι σωστή, έχουμε παρατηρήσει πως υπάρχουν κάποιες αδυναμίες.

Αρχικά έχουμε παρατηρήσει πως ο αλγόριθμος έχει την δυνατότητα να εντοπίσει άρνηση στο κείμενο και μας επιστρέφει το ανάλογο αντίστροφο αποτέλεσμα με αυτό χωρίς την άρνηση. Ένα παράδειγμα με την χρήση της άρνησης και τον τρόπο που ο αλγόριθμος το αντιμετωπίζει είναι το παρακάτω :

“**Nice** weather we have today” = **0.6**

“**Not nice** weather we have today” = **-0.5**

Είναι ξεκάθαρο ότι το συναίσθημα των δυο προτάσεων προέρχεται από την λέξη “Nice” με την διαφορά ότι στην δεύτερη περίπτωση ο αλγόριθμος παρατηρεί ότι υπάρχει άρνηση μέσω της λέξης “Not” και μετατρέπει το αποτέλεσμα ανάλογα.

Ωστόσο, ο αλγόριθμος αδυνατεί να εντοπίσει λέξεις όπως “Very” οι οποίες ενισχύουν το συναίσθημα μίας πρότασης όπως παρουσιάζεται παρακάτω.

“**Nice** weather we have today” = **0.6**

“**Very nice** weather we have today” = **0.5**

Παρόλο που υπάρχει η λέξη “Very” για να ενισχύσει το θετικό συναίσθημα που έχει η λέξη “Nice” ο αλγόριθμος αδυνατεί και μας επιστρέφει ένα λιγότερα θετικό συναίσθημα αφού η θετική τιμή από την λέξη “Nice” διαμοιράζετε σε μια πιο μεγάλη πρόταση.

Η σημαντικότερη παρατήρηση που έχουμε όμως, είναι ότι η εξαγωγή συναισθήματος μέσω του αλγορίθμου είναι αποκλειστικά από κείμενο. Αυτό είναι τεράστιο μειονέκτημα, για το λόγο ότι στην σήμερον ημέρα παρατηρείτε εκτενής χρήση emoji και gifs οι οποίες είναι σταθερές η κινούμενες εικόνες που μπορούν να εκφράσουν συναίσθημα, αντικείμενα η κάποιο μήνυμα.

Με το πλήθος των χρηστών που χρησιμοποιούν emoji ή gifs για να εκφράσουν τα σχόλια τους να αυξάνετε ο αλγόριθμος μας αδυνατεί όλο ένα και περισσότερο να εξάγει σωστές αντιπροσωπευτικές τιμές για το συναίσθημα, λόγο του ότι μειώνετε το κείμενο. Σε αυτές τις περιπτώσεις το ανθρώπινο μάτι μπορεί με ευκολία να κρίνει θετικό η αρνητικό συναίσθημα, αλλά ο αλγόριθμος θα εξάγει ουδέτερο συναίσθημα.

Κεφάλαιο 6

Συμπεράσματα - Μελλοντική Εργασία

Σε αυτό το τελευταίο κεφάλαιο γίνεται μια ανασκόπηση του έργου που έγινε σε αυτή την εργασία και των αποτελεσμάτων που εξάχθηκαν. Στο τέλος αναφέρονται οι τρόποι μελλοντικής βελτίωσης αυτού του συστήματος που έχουν αναγνωριστεί, καθώς επίσης και κάποιες σχετικές ερευνητικές εργασίες που θα μπορούσαν να ξεκινήσουν στο μέλλον, οι οποίες έγιναν εμφανείς κατά τη φάση της ανάπτυξης του παρόντος συστήματος.

6.1 Συμπεράσματα

Έχοντας ολοκληρώσει την εγκατάσταση των τεχνολογιών και υλοποιήσει τα 2 προγράμματα αυτό του plug-in και αυτό την Node Js πλατφόρμας έχουμε δημιουργήσει ένα πρόγραμμα το οποίο έχει την δυνατότητα να εκπληρώσει τους στόχους που είχαμε θέσει σε αυτήν την διπλωματική εργασία.

Με την δημιουργία του Web Scraper έχουμε πρόσβαση στις αναρτήσεις οποιαδήποτε σελίδας του Facebook και μέσω της Node Js πλατφόρμας μπορούμε να εξάγουμε το συναίσθημα και να κρίνουμε την εγκυρότητα του άρθρου σε πραγματικό χρόνο.

Με τα αποτελέσματα που εξάγουμε μπορούμε άμεσα με την χρήση της jQuery να ενημερώσουμε τον χρήστη για το συναίσθημα και για την εγκυρότητα του άρθρου με το να τροποποιήσουμε το στυλ (CSS) της ανάρτησης. Μπορούμε επίσης να παρέχουμε μέσω JavaScript alerts με στατιστική ενημέρωση για το γενικό συναίσθημα που περιέχει το προφίλ του χρήστη και προειδοποίηση για αναρτήσεις με άρθρα κατηγορία “fake news”.

Η αξιοπιστία για τη σωστή λειτουργία όλων των λειτουργειών είναι αρκετά ψηλή, με συγκεκριμένες περιπτώσεις για αποτυχία οι οποίες είναι κυρίως εξαρτώμενες από εξωτερικούς παράγοντες. Παράδειγμα τέτοιων εξωτερικών παραγόντων είναι η κακή ορθογραφία του κειμένου.

Μετά από αρκετή χρήση του προγράμματος όμως είχαμε αρχίσει να παρατηρούμε πως παρόλο ότι το πρόγραμμα της διπλωματικής εργασία λειτουργούσε ακριβώς όπως θα έπρεπε, οι τιμές που μας επέστρεφε για το συναίσθημα του περιεχομένου της ανάρτησης δεν είναι πάντα απόλυτα αντιπροσωπευτικές.

Αυτό το φαινόμενο που το πρόγραμμα παρόλο που λειτουργεί σωστά δεν μπορούσε να μας παρέχει πάντα τις πραγματικές συναισθηματικές τιμές ίσως οφείλεται στην αύξηση της χρήσης emoji και gifs. Οι κινούμενες και σταθερές εικόνες που μπορούν να εκφράσουν συναίσθημα , κάποιο αντικείμενο η μέχρι και να μεταφέρουν ένα ολόκληρο μήνυμα με απλά το πάτημα ενός κουμπιού, έχουν κατακτήσει και διαμορφώσει τον τρόπο που επικοινωνούν και εκφράζονται οι άνθρωποι στην σήμερον ημέρα. Έχουμε παρατηρήσει αναρτήσεις οι οποίες δεν περιέχουν καθόλου γραπτό κείμενο αλλά όλα το περιεχόμενο τους αποτελείται μόνο από αυτές τις σταθερές και κινούμενες εικόνες.

Οι λειτουργείες οι οποίες χρησιμοποιούμε για την εξαγωγή του συναισθήματος από την βιβλιοθήκη natural χρησιμοποιούν λεξικά που αντιστοιχούν λέξεις με την συναισθηματική τιμή τους. Το συναίσθημα που περιέχουν αυτές οι εικόνες αν και είναι άμεσο και ξεκάθαρο για το ανθρώπινο μάτι, το πρόγραμμα μας αδυνατεί να μπορεί να επεξεργαστεί αυτήν την πληροφορία αφού αυτές δεν περιέχουν καθόλου γραπτό κείμενο.

6.2 Μελλοντική εργασία

Μετά από εκτενή χρήση του προγράμματος που έχουμε δημιουργήσει για την διπλωματική αυτή εργασία, έχουμε παρατηρήσει αρκετές μελλοντικές εργασίες που θα μπορούσαν να πραγματοποιηθούν για να βελτιώσουν την λειτουργία του προγράμματος σε διάφορους τομείς στους οποίους τώρα αδυνατεί .

Οι διάφοροι τομείς χωρίζονται σε :

- Διανομή του προγράμματος
- Εξαγωγή συναισθήματος
- Αποθήκευση δεδομένων

Διανομή του προγράμματος

Αρχικά η διανομή του προγράμματος με την μορφή που έχει τώρα αποτελεί μεγαλύτερη δυσκολία απ' όσο θα μπορούσε. Το γεγονός ότι το πρόγραμμα αποτελείται από 2 κομμάτια: αυτό του plug-in (browser-side) και αυτό της Node js πλατφόρμας (Server-side) κάνει τη διαδικασία διανομής δυσκολότερη. Ο λόγος είναι ότι το κομμάτι της Node Js πλατφόρμας χρειάζεται να ανεβεί σε server ώστε να λειτουργεί σωστά το plug-in μας που θα εξάγει το περιεχόμενο από τον χρήστη.

Για να διευκολύνουμε την διανομή του προγράμματος θα ήταν ιδανικό να απαλλάξουμε το πρόγραμμα μας από την Node Js πλατφόρμα, διατηρώντας όμως όλες αυτές τις χρήσιμες λειτουργείες που μας παρέχει μέσω της NPM και όλες τις βιβλιοθήκες της. Με την συνεχής ανάπτυξη διαφόρων τεχνικών θα ήταν δυνατό να εγκαταστήσουμε όλες τις βιβλιοθήκες που χρησιμοποιούμε από την NPM στη Node Js πλατφόρμα μας στο plug-in μας ως browser-side βιβλιοθήκες. Διατηρώντας έτσι όλες τις λειτουργείες του προγράμματος μας αλλά μετατρέποντας το σε ένα πιο αυτοτελές plug-in με ευκολότερη διαδικασία διανομής, το οποίο δεν θα χρειάζεται πλέον την εξωτερική πλατφόρμα για να λειτουργήσει.

Εξαγωγή συναισθήματος

Όπως αναλύσαμε και πιο πάνω στα συμπεράσματα, μετά από χρήση του προγράμματος αρχίσαμε να παρατηρούμε πως η χρήση του σταθερών και κινούμενων εικόνων είναι πολύ μεγαλύτερη από αυτή που θα περίμενε κανείς. Αυτή η εκτεταμένη χρήση των εικονιδίων αποτελεί το μεγαλύτερο εμπόδιο του προγράμματος μας να εξάγει με ακρίβεια το συναίσθημα από τις αναρτήσεις. Ακόμη και στις περιπτώσεις όπου μία ανάρτηση δεν αποτελείτε μόνο από τέτοιου είδους εικονίδια η πιθανότητα μην περιέχει καθόλου εικονίδια είναι σχεδόν αδύνατο. Η έντονη αυτή χρήση των εικονιδίων μειώνουν τον όγκο του κειμένου που μπορεί το πρόγραμμα μας να επεξεργαστεί με αποτέλεσμα να εξάγει μια τιμή κοντά στο 0 η οποία υποδηλώνει ουδέτερο συναίσθημα.

Ίσως η πιο σημαντική μελλοντική εργασία για το πρόγραμμα μας θα ήταν η δημιουργία και χρήση αλγορίθμων οι οποίοι μπορούν να αντιμετωπίσουν όλα αυτά τα εικονίδια και να εξάγουν από αυτά την συναισθηματική τιμή που τα αντιπροσωπεύει. Με την συνεργασία των αλγορίθμων εξαγωγής συναισθήματος από το κείμενο και εικόνες θα μπορούσαμε να έχουμε πολύ πιο ακριβείς και αντιπροσωπευτικές τιμές συναισθήματος για το περιεχόμενο μας.

Βιβλιογραφία

https://www.w3.org/TR/WD-DOM/introduction.html

https://www.w3schools.com/js/js_htmldom.asp

https://www.hostinger.com/tutorials/what-is-html

https://techterms.com/definition/javascript

https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX

https://www.json.org/json-en.html

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage

https://www.tutorialspoint.com/nodejs/nodejs_introduction.html

https://www.w3schools.com/whatis/whatis_npm.asp

https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/

https://www.w3schools.com/jquery/jquery_intro.asp