Link Analysis-Page Rank implemtation

במהלך קורס "אחזור מידע" בחרנו לעבוד על פרויקט "Link Analysis" .

אנו רוצים למצוא את הדירוג והאיכות של אתר מסוים, למען מטרה זאת מודל הלמידה שנבחר הינו "Page Rank" , מודל זה עובד בצורה הבאה:

1. יהי אתר X שעליו רצה למצוא את הדירוג, ניקח את כל האתרים ש-X מצביע אליהם.
2. לכל אתר ש-X מצביע עליו נבצע אותה פעולה בשביל למצוא רשת של אתרים המקושרים אחד אל השני דרך אתר X.
3. לאחר מציאת כל האתרים נחשב את ה-Page Rank של אתר X ע"י חישוב רקורסיבי של כל האתרים אליהם הוא מצביע וכך נדע את הדירוג של X.

למען ביצוע פעולות אלה בחרנו בכלים מסוימים לכל שלב בשביל מימוש מתאים.
סביבת העבודה שנבחרה הייתה פייתון, עם מפרש 3.6, שאר הכלים שנעזרנו בהם היו חבילות של הסביבה הנבחרת.

נחלק את פעולות התוכנית לשלבים למען הנוחות:

שלב 1:

1. אחרי שנבחרי אתר מסוים, נשתמש בחבילה של urllib בשביל לפתוח את קוד המקור שלו.
2. ניקח את קוד המקור ונמיר אותו ל-string אשר איתו נוכל לעבוד:
   2.1. נבצע ניקוי של כל מה שאינו חלק מלינק כלשהו.
   2.1. נבצע "תיקון" של כל לינק, הורדת "-, '-, -/-, -//- והוספת http:// לתחילת כל לינק.
   2.2. כל לינק תקין נכנס למילון ראשי אשר ישמור את הנתונים, כמות כניסות/יציאות לאתר, רשימה של כל האתרים הנכנסים/יוצאים אל/מ.. האתר ועדכון מצב הצומת← "אם ביקרנו בה או לא".
   2.3. אם נתקלים בצומת אשר לא מקושרת לאתר הראשי הנבדק, כגון אם האתר הראשי www.sce.ac.il ונתקלנו ב- www.ynet.co.il , לא נסרוק את YNET אלה נייצר צומת אחת של YNET וכל צומת אשר נתקל בהמשך אשר תתחיל עם אותו קישור של YNET נשייך אותה לצומת הראשית של YNET .
3. חזור על פעולות "1" ו-"2" שלוש פעמים, כל פעם ניקח את הצמתים הפנימיות של המילון ובכך נבקר כמעט את כל הקישורים המקושרים אל האתר הספציפי, מאחר וכל איטרציה מגדילה את כמות הקישורים באופן מעריכי נגביל את כמות האיטרציות ל-3 .

בשלב זה אנו משתמשים בחבילה urllib ← "from urllib.request import urlopen" .
ונסיים את השלב עם מילון מוכן שעליו נעבוד בהמשך.

להלן דוגמא של המילון אשר הוא נבנה על האתר http://www.sce.ac.il/ :

המילון מורכב בצורה הבאה←
{ [רשימה כל הקישורים היוצאים] [רשימת כל הקישורים הנכנסים] [מצב ביקור הצומת] [כמות הקישורים היוצאים] [כמות הקישורים הנכנסים] : [שם הקישור← מפתח המילון] }

- בזמן סיום השלב זמן הריצה נע בין 20 דק' ל-4 שעות, תלוי אתר נבדק, לכן בשביל ייעול הזמנים השתמשנו בחבילה "Pickle" אשר תפקידה לשמור אובייקטים לפי בחירת המשתמש, ולכן אנו נשמור את המילון הסופי בתיקייה אשר תיווצר בקובץ הפרויקט תחת השם של האתר, כך שבמקרה ונרצה לרוץ על האתר יהייה לנו כבר שדה נתונים מוכן לשימוש.
אופציה תינתן למשתמש התוכנית בתחילת ההרצה לבחור אם הוא מעוניין לבנות מודל מהתחלה על אתר חדש או לקחת את מילון הנתונים באתר שנבדק מראש.

<u>שלב 2:</u>

בשביל לחשב את ה- Page Rank אנו צריכים ליצור חישוב רקורסיבי אך למען הנוחות אנו נחשב בצורה איטרטיבית מאחר וחישוב רקורסיבי דורש כמות משאבים וזכרון שאנו לא מסוגלים לספק כיום.

ניקח את המילון קישורים שנבנה בתהליך הקודם ונעבור ליצירת מטריצת המעבר ונמשיך לייצר נוסחאות לחישוב.

בחלק זה נבחרה החבילה של numpy (← "imprt numpy") אשר מייעלת את הקריאה ועיבוד המידע במטריצות.

1. ניקח את המילון קישורים ונבנה את המטריצה בגודל nXn כאשר n הוא כמות המפתחות (= הקישורים ללא כפילויות) במילון.
2. נרוץ כל תא במילון בצורה הבאה:
   a. תרוץ על שורה X.
   b. תרוץ על עמודה Y.
   c. על כל תא תסתכל על הקישור בשורה X ועל הקישור המתאים לו בעמודה Y, תסתכל במילון האם Y.link מצביע חזרה אל X.link :
      i. אם לא תשים בתא "0".
      ii. אם כן תשים בתא (כמות כל הקישורים היוצאים מ- Y.link)/1.

דוגמא למטריצה אחרי הריצה הראשונה על הנתונים←

```
[[0.00480769 0.         0.         ... 0.         0.         0.         ]
 [0.         0.00490196 0.         ... 0.         0.         0.         ]
 [0.         0.         0.00473934 ... 0.         0.         0.         ]
 ...
 [0.         0.         0.         ... 0.         0.         0.         ]
 [0.         0.         0.         ... 0.         0.         0.         ]
 [0.         0.         0.         ... 0.         0.         0.         ]]
```

<u>שלב 3:</u>

אחרי הריצה הראשונה על המטריצה אנו צריכים לייצר משוואות אשר בנויות משורות המטריצה בשביל החישוב הערכים באיטרציה הבאה של המטריצה.
כל וקטור כזה יורכב בצורה הבאה:
יהי שורה X, ערכו הבא של קישור של שורה X מורכב מסכום ערכי התא של (X,Yi)*(הערך של קישור Yi).
בשביל לא לאבד נתונים נשמור לפני כל איטרציה וקטור שמורכב מערכי הקישורים בשורות כך שנוכל להשתמש בנתונים על כל המטריצה ללא איבוד/שינוי הנתונים תוך כדי ריצה, לאחר סיום האיטרציה נעדכן את הוקטור שמחזיק את הערכים (וקטור זה לבסוף יכיל את הערכים של הדירוג הסופי) ונחזור על הריצה על המטריצה עד להגעת ערכים המתכנסים למספר סופי.
כמות האיטרציות שנבצע תהייה 50 כי לאחר כ-50 איטרציות הערכים מתכנסים לאותם ערכים.
וקטור הערכים הסופיים לאחר האיטרציות יהוו את ערכי ה- Page Rank של הקישורים.

1. שמור את הערכים בוקטור למען החישובים.
2. חזור על השלבים בחלק הקודם לחישובי בערכים במטריצה.
3. עדכן את ערכי הוקטור.
4. חזור על שלבים 1-3 50 פעמים.
5.

- יכול להיות שצומת במטריצה עם "צומת קצה" (dead end, צומת שלא מצביעה לקישור אחר), או שהיא חלק מ"מלכודת" (spider trap) ולכן בכל איטרציה הערך המחושב יהייה רק 80% מהערך המתעדכן וה-20% הנותרים יהיו "מס" על הצומת מהערך שיתווספו כדי שערך צומת לא יוכל להתאפס.

➔ דוגמא להרצת משוואות על המטריצה עם מיסוי של 20%.

◆Equations $\mathbf{v} = 0.8(M\mathbf{v}) + 0.2$:

$y = 0.8(y/2 + a/2) + 0.2$
$a = 0.8(y/2) + 0.2$
$m = 0.8(a/2 + m) + 0.2$

y		1	1.00	0.84	0.776		7/11
a	=	1	0.60	0.60	0.536	. . .	5/11
m		1	1.40	1.56	1.688		21/11

$$* \; v_n = [v_1, v_2, v_3, v_4] \text{ after n steps}$$

$$v_{n+1} = M * v_n = \begin{Bmatrix} 0 & 0 & 1/3 & 0 \\ ½ & 0 & 1/3 & 0 \\ ½ & 0 & 0 & 1 \\ 0 & 1 & 1/3 & 0 \end{Bmatrix} * \begin{Bmatrix} V1 \\ V2 \\ V3 \\ V4 \end{Bmatrix} = \begin{Bmatrix} V1_{n+1} \\ V2_{n+1} \\ V3_{n+1} \\ V4_{n+1} \end{Bmatrix}$$

שלב 4:

בשלב זה נייצר קובץ אשר יכיל רשימה של הקישורים ולידם את התוצאות המנורמלות (0-1) של המטריצה.
הרשימה תהייה ממוינת עפ"י סדר יורד כך שבראש הרשימה יהיו את הקישורים עם הדירוג הכי גבוה.

1. צור רשימה של הקישורים עם ערך הדירוג שלהם.
2. נמיין את הרשימה בסדר יורד עפ"י mergeSort.
3. נעבור על הרשימה הממוינת ונרשום אותה לקובץ txt אשר נכנס אוטומטית לתיקיית האתר בתוך הפרויקט.

דוגמא לקובץ הטקסט שנוצר בסוף התהליך החישוב על www.ynet.co.il ←

output1.txt - פנקס רשימות

קובץ עריכה עיצוב תצוגה עזרה

```
http://www.ynet.co.il  1.0
http://www.googletagmanager.com  0.5348323111469109
http://www.googletagservices.com  0.5005972352810606
https://z.ynet.co.il  0.49963846093922737
https://  0.49963846093922737
http://www.bigdeal.co.il  0.4978674230723368
http://www.ynetshops.co.il  0.4978674230723368
http://www.winwin.co.il  0.4978674230723368
http://www.mynet.co.il  0.4978674230723368
http://www.ynet.co.il/home/0,7340,L-8,00.html  0.4978674230723368
http://www.facebook.com  0.4975680986898079
http://www.radware.com  0.4636323472064865
http://bit.ly  0.4636323472064865
http://www.alljobs.co.il  0.4636323472064865
http://pplus.ynet.co.il  0.4636323472064865
http://www.blazermagazine.co.il  0.4636323472064865
http://www.calcalist.co.il  0.4636323472064865
http://www.activetrail.co.il  0.4636323472064865
http://www.google.com  0.4636323472064865
http://www.akamai.com  0.4636323472064865
https://m.ynet.co.il  0.4636323472064865
http://www.promisejs.org  0.4636323472064865
http://www.ynetnews.com  0.4636323472064865
https://images1.ynet.co.il  0.4525070619036727
http://hot.ynet.co.il  0.4501924433284628
http://www.pro.co.il  0.4385773380952185
http://z.ynet.co.il  0.4385773380952185
http://www.ynetart.co.il  0.4385773380952185
http://www.tali-rights.co.il  0.4385773380952185
http://www.kick.co.il  0.4385773380952185
http://www.visit.yedioth.co.il  0.4385773380952185
http://www.zikaronet.co.il  0.4385773380952185
http://www.frogi.co.il  0.4385773380952185
http://www.shoppinglaisha.co.il  0.4385773380952185
http://nadlan.winwin.co.il  0.4385773380952185
http://www.xnet.co.il  0.4385773380952185
http://www.kikar.co.il  0.4385773380952185
http://www.yediot.co.il  0.4385773380952185
```

שלב 5:

אחרי המיון ושמירת הנתונים לקובץ נרצה לייצג ויזואלית את הנתונים בגרף דו-מימדי.
לכן השתמשנו בחבילה "networkx" (import network as nx) אשר מיועדת לציור וייצוג גרפים.

נשתמש במילון הראשי , זה שמחזיק את הלינקים הנכנסים והיוצאים של כל עמוד.

נרוץ על המילון :

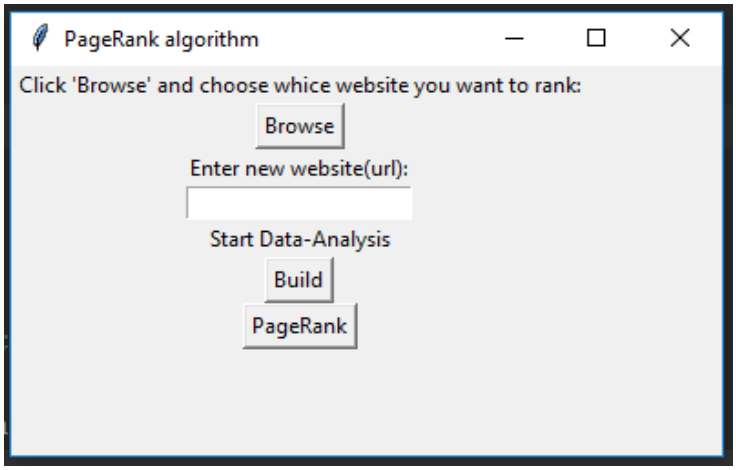
1.אם לא קיים בגרף, נוסיף את המפתח כצומת חדשה.

2.עבור כל לינק יוצא נוסיף קשת ביניהם.

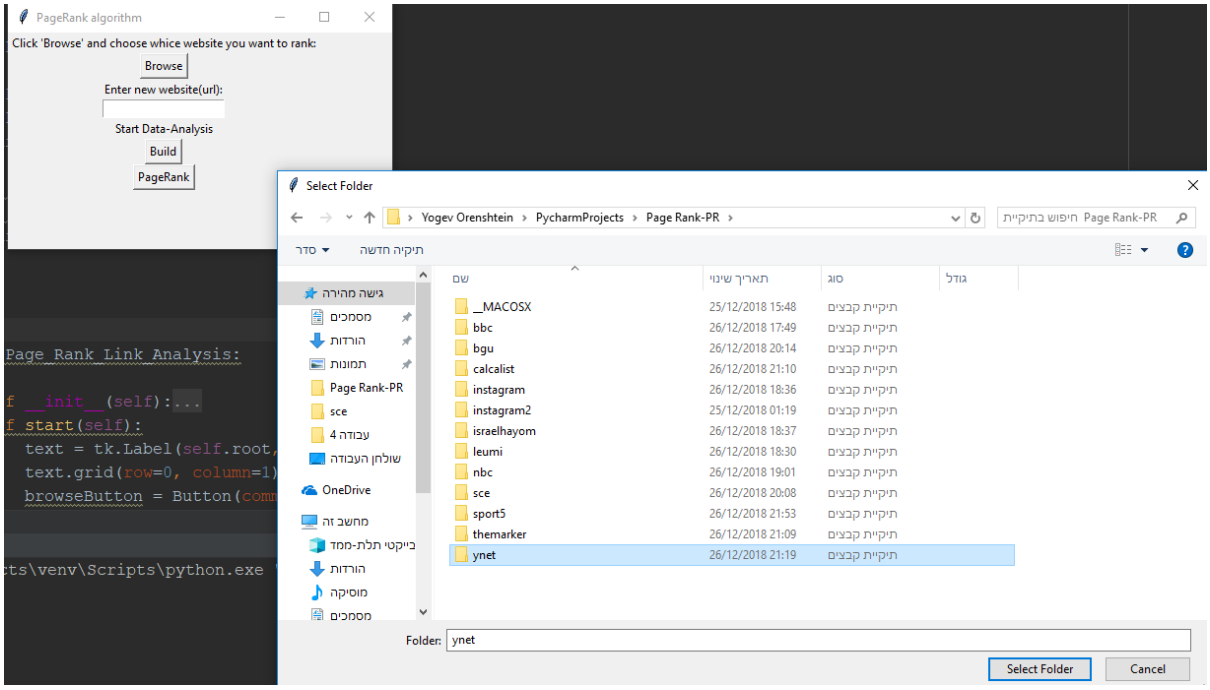
- אחרי כתיבת התוכנית הוספנו GUI בשביל נוחות המשתמש כאשר יש אופציה לבחירה בין מודל אתר שכבר קיים, או לבנות מודל חדש, וחשב את ה-Page Rank ביחד עם הצגתו הויזואלית.
בשביל ה-GUI השתמשנו בחבילה של tkinter אשר מתמחה בלייצר GUI נוח ופשוט.
- עוד מחלקה בתוכנית היא os (import os) אשר השתמשנו בה ליצירת תיקיות וקבצים במיקומים הנכונים כל פעם שמריצים אתר.

דוגמאות להרצה התוכנית:

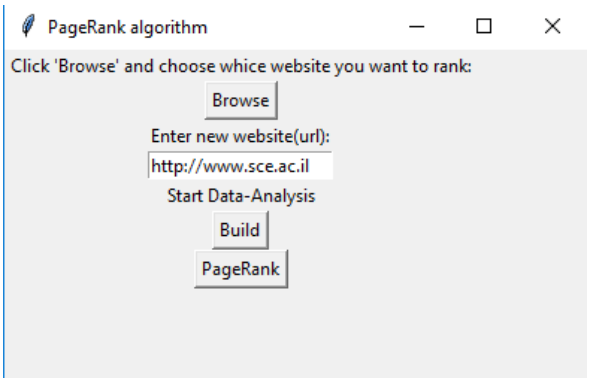
- מסך ראשי של התוכנית



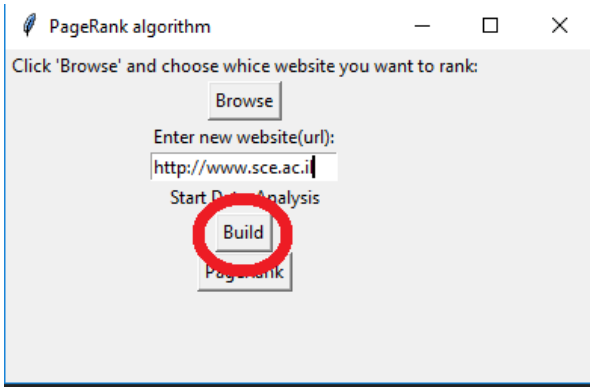
- המשך ההרצה כאשר נבחרה הפעולה "**browse**", יש לבחור מתוך תיקיית הפרוייקט את התיקייה של האתר שרוצים לרוץ עליו.



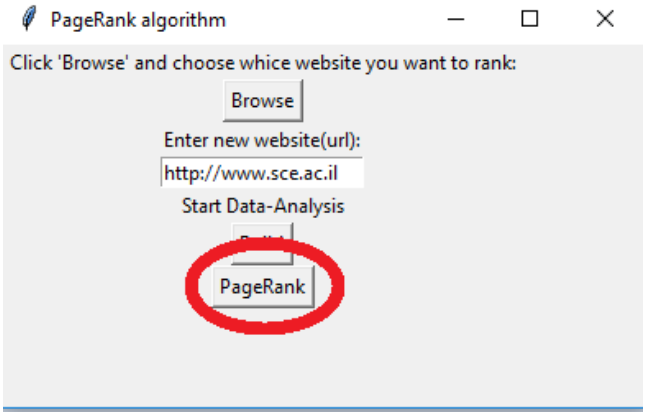
- אופציה נוספת להכניס אתר לבנות עליו את המודל.
- **חשוב!!** יש לכתוב את הכתובת עם http:// בהתחלה על מנת הפעלת התוכנית.



- אחרי כתיבת שם האתר יש ללחוץ על "build".



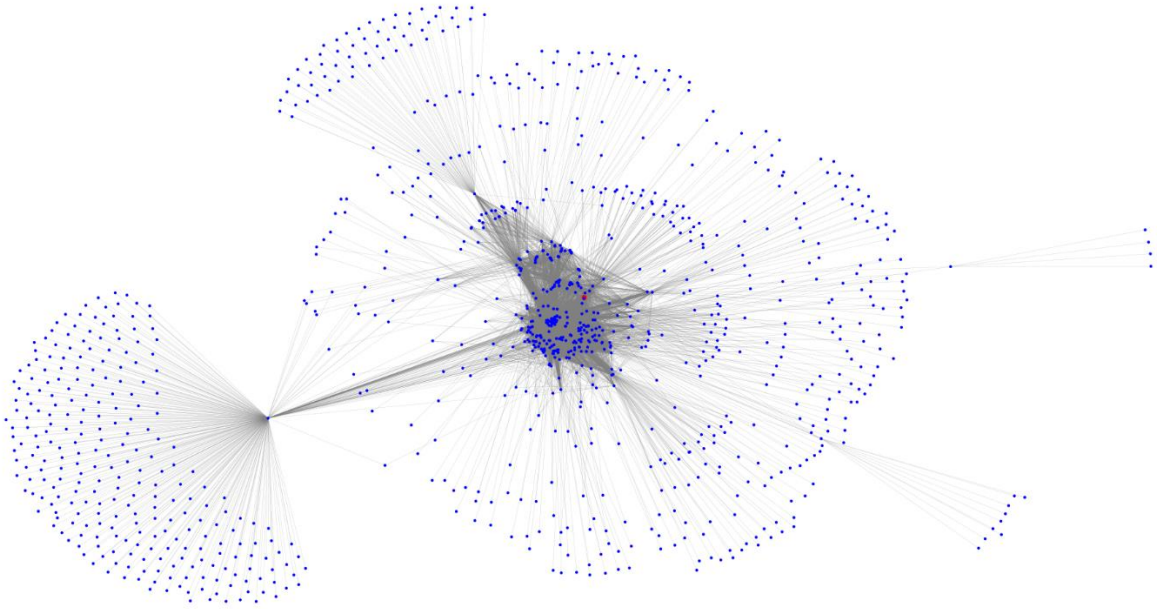
- אחרי בחירת browse/build יש לבחור ב- PageRank על מנת לבצע את החישוב והדפסת הגרף.



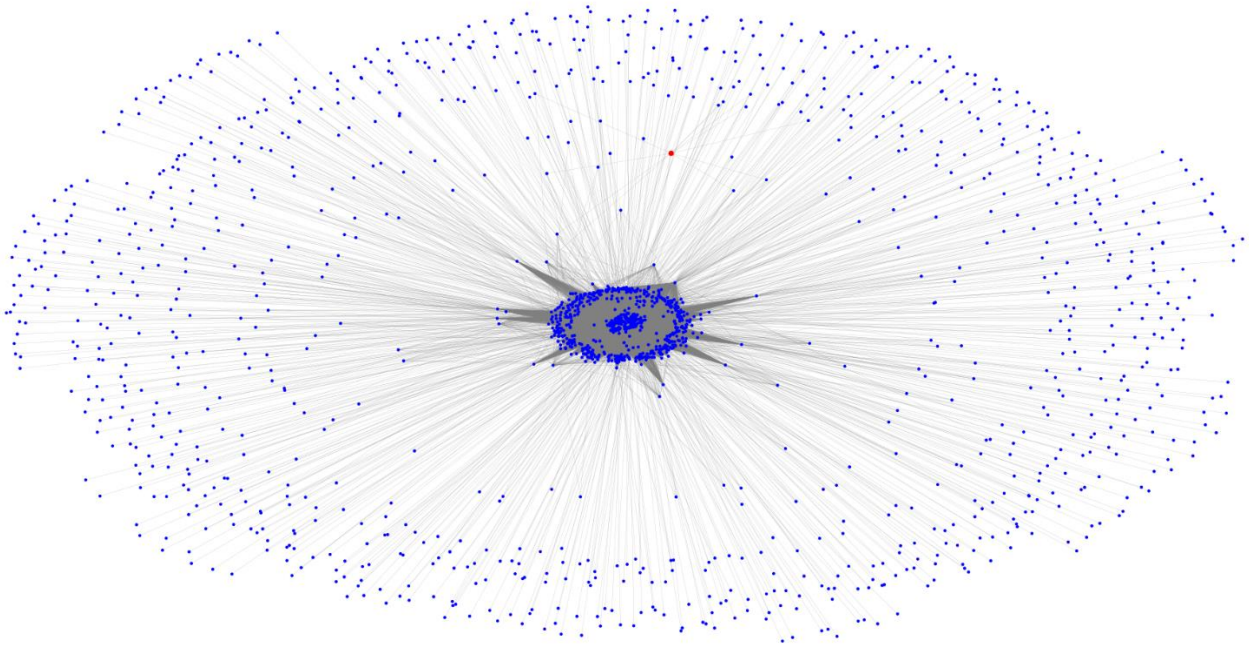
- בשלב זה אחרי המתנה תינתן האופציה לצייר את הגרף, במידה ו"כן" נקבל גרף קודקודים אשר רואים בו את התפלגות הצמתים כך שאפשר לראות מי מקושר למי.
- קודקוד אדום מסמל את הצומת/צמתים עם הדירוג הכי גבוה.

<u>דוגמאות לגרפים:</u>

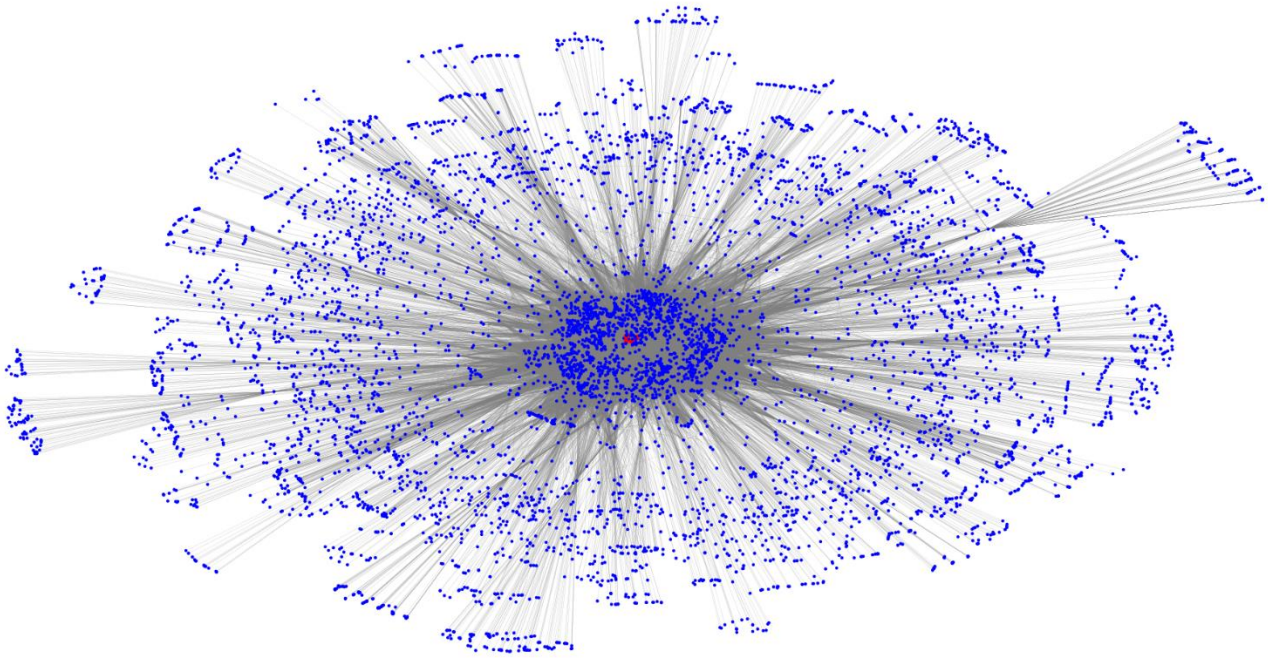
www.ynet.co.il



www.sce.ac.il



In.bgu.ac.il



www.nbc.com