דו"ח תרגיל 4 – עיבוד שפות טבעיות

מגיש: שגיא גוילי, ת.ז. 203638804

תרגיל זה הוא סוג של המשך לתרגיל הקודם רק שהפעם אנחנו נשתמש בשיטה של word embeddings. קיבלנו וקטורי מילים באורכים של 50 ו-300 ובעזרתם סיווגנו את יחידות הסיווג שלנו (שני משפטים בתרגיל הזה) לקורפוס המתאים – אנגלית רגילה או אנגלית פשוטה.

חלק א' – הכנה והתנסות:

בשלב הראשון ביצענו התנסות בוקטורי המילים על ידי פעולות בסיסיות שלמדנו בהרצאה, כמו לבדוק דימיון בין שתי מילים על ידי קוסינוס הזווית בין שני הוקטורים המייצגים של כל מילה (וקטור במרחב דו-מימדי כמובן לא וקטור מילים). ארבעת צמדי המילים שבחרתי לשימוש בפונקציה similarity, תוצאות והסברים:

midfielder – goalkeeper, glove50d result - 0.7324171, glove300d result - 0.601985

בחרתי את צמד המילים הנ"ל בגלל הקירבה הברורה שלהם- עולם הכדורגל. קשר ושוער הם שניהם תפקידים במשחק הכדורגל ולכן ציפיתי שרמת הדמיון בניהם תהיה גבוהה. התוצאות היום בהתאם- 60 עד 75 אחוזי דמיון בין שתי המילים די איששו את ההשערה שלי, אך הייתה לי סוג של הפתעה- שתי המילים הן בהכרח קשורות אך ורק לספורט ובפרט לכדורגל לכן ציפיתי לדמיון אף גבוה יותר ממה שהתקבל, ככל הנראה במקרים לא מעטים פשוט לא נוטים לשים את שני התפקידים האלה באותו משפט (אולי מהטעם ששניהם תפקידים קצת שונים- אחד משחק במרכז המגרש, משתמש בעיקר ברגליים ו"מנהל" את המשחק כאשר השני שומר על ספציפית על השער, משתמש בעיקר בידיים ומשמעותי ברגעים מועטים יותר)

midfielder – learning, glove50d result - 0.13991463, glove300d result - 0.011875518

לעומת הצמד הראשון, בחרתי פה שני מושגים עם עולם תוכן שונה- "למידה" יותר מתקשר לעולם ההשכלה כאשר "קשר" לעולם הכדורגל, על פניו שני מושגים שלא אמור להיות בניהם קשר רב מדי. התוצאות אכן איששו את השערתי, בעיקר בוקטורים באורך 300 בהם היה דמיון בפחות מ-2%. אני מניח שבפעמים בהם כן היה קשר זה בתיאור של התפתחות של שחקן בתפקיד קשר (The midfielder is learning new techniques) וכמו שניתן להבחין גם זה לא קורה יותר מדי.

shallow – deep, glove50d result - 0.7442153, glove300d result - 0.5457015

צמד המילים הנ"ל מתאר שני הפכים- רדוד ועמוק. באופן כללי המחשבה הראשונית שלי הייתה ששני הפכים לרוב לא יאמרו/יכתבו באותו המשפט או אפילו בקירבה כלשהיא האחד לשני, למשל אם ידברו על מעיין שהוא עמוק לא תהיה סיבה להזכיר את המילה רדוד. אך לאחר מחשבה מסוימת הנטייה שלי הייתה לחשוב שישנם מקרים בהם הפכים דווקא כן יגיעו ביחד, למשל כאשר מתארים נחל שמתמלא בתקופת החורף אך מתייבש טיפה בתקופת הקיץ, שני ההפכים יהיו מאוד סמוכים האחד לשני. התוצאות התאימו למחשבה השנייה שלי בסופו של דבר, 55-75 אחוזי דמיון די הפריכו את המחשבה ששני הפכים לא יאמרו בקירבה כלשהיא בדיבור או משפט, ובאיזשהו מקום זה קצת הפתיע אותי- ציפיתי לאחוזים קצת יותר נמוכים.

shallow – dog, glove50d result - 0.32064644, glove300d result - 0.058880158

לעומת הצמד הקודם, בצמד זה לקחתי שתי מילים מעולמות תוכן שונים לחלוטין- כלב הוא לרוב חיה, חיית בית בפרט, כאשר רדוד הוא לרוב שם תואר. המחשבה שלי הייתה שהדימיון יהיה מאוד מאוד קטן, כי כלב לא יכול להיות רדוד ובנוסף כלב הוא חיית בית לכן לרוב הסמיכות שלו לנחלים, מעיינות או מקומות אחרים שניתן להגיד עליהם שהם רדודים אמורה להיות מאוד נמוכה. התוצאות די הפתיעו אותי, אמנם אכן הדימיון קטן, 5-32 אחוזי דמיון, אבל ציפיתי לאחוזי דמיון קטנים אף יותר (כמו בצמד "קשר" ו-"למידה"). אני מתאר לעצמי שהדבר נובע מכך שעל אף האמור להעיל, קורה שכלב מגיע לא מעט למקומות רדודים, לדוגמא בעלי

כלבים לוקחים כלבים לים לא מעט, ולרוב כלבים מפחדים מהים ומהגלים בים בפרט אז אולי ישנה נטייה לכלבים כן להיכנס למים רדודים יותר ולכן זה נאמר במשפט.

לסיכום, ניתן לראות ירידה מתמדת באחוזי הדמיון בין שתי מילים כתלות באורך וקטור המילים- כאשר אורך הוקטור עולה הדימיון יורד, והדבר אכן הגיוני שכן עוד ועוד מילים נוספו ולכן כאשר במקרה שלנו מגדילים את אורך הוקטור פי 6 ישנה עלייה רבה במספר המילים הנוטות להיות קרובות למילה מסוימת מה שמוריד את אחוז הדמיון למילים ספציפיות.

4 מילים לשימוש בפונקציה most_similar:

parrot:

Glove 50d Results

Word	Similarity
lizard	**0.7699268460273743**
parrots	**0.7595348358154297**
songbird	**0.7308381199836731**
parakeet	**0.7296543121337891**
giraffe	**0.727449893951416**
moths	**0.7244707345962524**
owl	**0.7157155275344849**
moth	**0.713994026184082**
dragonfly	**0.7072760462760925**
gecko	**0.7041857838630676**

Glove 300d Results

Word	Similarity
parrots	**0.6657733917236328**
parakeet	**0.5025736093521118**
macaw	**0.486573725938797**
frog	**0.4563598930835724**
birds	**0.44363030791282654**
eclectus	**0.4379664659500122**
squirrel	**0.4336472749710083**
monkey	**0.4326956868171692**
bird	**0.42096829414367676**
pheasant	**0.4154737591743469**

תוכי הוא חיה ממשפחת הציפורים. הנטייה שלי הייתה לחשוב שהדברים הדומים ביותר יהיו זנים של תוכים, ציפורים בכללי וציפורים ספציפיות. הופתעתי לגלות שבשני המודלים (בעיקר ב-50) קיבלתי במקרים רבים חיות אחרות לגמרי, אפילו לא ציפורים, בדימיון גבוה מאוד ולעיתים אף לפני זנים ספציפיים של תוכים לדוגמא ג'ירפה, עש, צפרדע (שלא הייתה בכלל במודל הראשון וקפצה גבוה מאוד במודל השני), קוף ועוד.. קשה לי להסביר את התופעה אך כלל הנראה מדברים על תוכים בהקשר של חיות אלו גם כן . lizard (לטאה) הכי הפתיעה אותי, 77% אחוזי דמיון במודל הראשון, ההשערה היחידה שעלתה לי הייתה שתוכים ניזונים מלטאות, מעבר לכך קשה לי להסביר את התופעה.

politician:

Glove 50d Results

Word	Similarity
businessman	**0.7817543745040894**
mp	**0.7724176645278931**
liberal	**0.7486166954040527**
lawmaker	**0.73398756980896**
jurist	**0.7278751134872437**
parliamentarian	**0.7202852964401245**
conservative	**0.7129016518592834**
elected	**0.7100611925125122**
candidate	**0.7029688954353333**
citizen	**0.700545072555542**

Glove 300d Results

Word	Similarity
businessman	**0.6441149115562439**
jurist	**0.5666126012802124**
lawmaker	**0.5599642992019653**
mp	**0.5299320220947266**
candidate	**0.5184500217437744**
liberal	**0.5171478390693665**
prominent	**0.5160537958145142**
politicians	**0.5150622129440308**
legislator	**0.5093942284584045**
lawyer	**0.5091251134872437**

פוליטיקאי הוא דמות ציבורית שנבחרה לייצג את העם בפרלמנט. רוב המילים הדומות שעלו על ידי most_similair תואמות את הציפיות שלי- מועמד, נבחר, פרלמנטר, עושה חוק וכדומה.

בלטו לי בעין המילים איש עסקים וליברל:

באשר לאיש עסקים אני חושב שהדבר נובע ראשית מכך שפוליטיקאי מצליח הרבה בזכות עסקאות מוצלחות שהוא סוגר, אם זה בהבאת תקציבים או סגירת הסכמים עם פוליטיקאים אחרים, אבל לדעתי בעיקר מראה מגמה של אנשים במחשבה שלהם על פוליטיקאים- במקום להיות אנשים שדואגים לנו לפי המצע אותו הציגו לנו כשבחרנו אותם, הם בעצם אנשי עסקים שדואגים יותר לעצמם ולעסקים האישיים שלהם ולראייה איש עסקים קיבל בשני המודלים את הדמיון הגבוה ביותר לפוליטיקאי.

באשר לליברל, שזו בעצם השקפת עולם, זה פחות נכון לגבי כלל הפוליטיקאים שכן לא כל פוליטיקאי הוא ליברל ולא בטוח שאפילו רובם ליברלים, אבל ככל הנראה במרבית הפעמים כאשר פוליטיקאי נכלל במשפט או הדיבור נעשית השוואה לראות כמה פוליטיקאי ליברל או שאנשים רבים נוטים לראות פוליטיקאי כליברל.

journalist:

Glove 50d Results

Word	Similarity
reporter	**0.8463937044143677**
writer	**0.837344765663147**
photographer	**0.8265179395675659**
editor	**0.765821635723114**
author	**0.7652691602706909**
freelance	**0.7579443454742432**
novelist	**0.7515745162963867**
correspondent	**0.7421873807907104**
citizen	**0.7381494641304016**
translator	**0.7372987270355225**

Glove 300d Results

Word	Similarity
reporter	**0.759986162185669**
writer	**0.6434952616691589**
photographer	**0.6208367347717285**
correspondent	**0.6134044528007507**
freelance	**0.5966933965682983**
journalists	**0.5694776177406311**
editor	**0.5681557655334473**
novelist	**0.5498708486557007**
author	**0.5399045944213867**
activist	**0.5386793613433838**

עיתונאי הוא אדם שמסקר דברים בתקשורת. מילה זו עניינה אותי, בדומה לפוליטיקאי, בגלל הצורה בה אנשים רואים כיום עיתונאים במקרים לא מועטים-מוטים, מכורים ולא אותנטיים. חשבתי שבדומה לפוליטיקאי גם כאן אמצא, לפחות בחלק מהמילים, הקשר לדברים הללו אך הופתעתי לגלות שלא כך הדבר ובשני המודלים אכן המילים הדומות ביותר לעיתונאי הן אכן המילים שבצורה טהורה הייתי חושב לנכון לתאר עיתונאי. אני מניח שהמחשבה שלי נוצרה כתוצאה מלראות תגובות של אנשים על עיתונאים ופחות ממאמרים (עליהם, ביחד עם ויקיפדיה, מתבססים וקטורים המילים) שבהם עדיין ההקשר של עיתונאי הוא בקונטקסט הרגיל שלו ללא הטיות ועל כן התוצאות שניתן לראות בטבלאות הנ"ל.

cop:

Glove 50d Results

Word	Similarity
cops	**0.8096667528152466**
gangster	**0.7510235905647278**
detective	**0.7214533686637878**
kid	**0.7112536430358887**
thug	**0.7076013684272766**
guy	**0.6722906827926636**
bumbling	**0.661464512348175**
crime	**0.6613893508911133**
dumb	**0.657253086566925**
mobster	**0.6571474075317383**

Glove 300d Results

Word	Similarity
cops	**0.7100366950035095**
detective	**0.5643267631530762**
gangster	**0.4631376266479492**
undercover	**0.4631376266479492**
nypd	**0.46138995885849**
thug	**0.4418871998786926**
guy	**0.4381211996078491**
policeman	**0.4292941689491272**
lapd	**0.4275149703025818**
mobster	**0.4166252017021179**

שוטר הוא אוכף חוק מטעם המדינה שאמון על שמירת הסדר הציבורי, cop הוא כינוי לשוטר, לרוב באור שלילי יותר. המחשבה שלי הייתה לראות מונחים בעולם תוכן של משטרה ופשע, ואכן רוב המילים הגיעו מעולם תוכן זה.

בלטו בעיניי המילים kid, bumbling ו-dumb:

- שוטר הוא לא ילד, גם אין נטייה מובהקת של שוטר לטפל במקרים של הפרת החוק על ידי ילדים וכן אין באופן בולט מדי הפרות חוק שקשורות לילדים בלבד. לכן ההשערה שלי היא שמילה זו הגיעה בהקשר (די גבוה חשוב לציין, לפחות במודל של ה-50) למילה שוטר ככל הנראה ממקומות רבים בהם ציינו שוטר כחזק על ילדים יותר מעל פושעים מבוגרים יותר שאיתם הוא מפחד להתעסק ולכן מכנים אותו ילד. מה שמחזק את ההשערה הוא ש-cop, כפי שנאמר להעיל, הוא יותר מילת גנאי מאשר השם הישיר של התפקיד (policeman)
- הפירוש הישיר של bumbling הוא שם תואר שמראה על חוסר יכולת ולבלבול. המחשבה הטבעית היא ששוטר אמור להיות דמות חזקה ומאוד דומיננטית. שם התואר הזה נשמע פחות מתאים

לדמות השוטר אבל ככל הנראה מילה זו נמצאה דומה ל-cop בעיקר בגלל האור השלילי של cop שסביבתה נמצאו ביקורות של אנשים כלפי שוטרים- לעיתים מבולבלים ולא מוכשרים במיוחד.

- Dumb משמעותו טיפש, בדומה ל-bumbling גם מילה זו הופיעה בקשר של cop בגלל האור השלילי ששוטרים מוצגים כאשר הם מכונים cops. באופן טבעי שוטר לא אמור להיות טיפש, אלא להפך.

3 דוגמאות לשימוש ב-Positive ו-Negative בפונקציה most_similar:

$$france = country + (paris - capital)$$

Glove 50d Results

Word	Similarity
france	**0.7531338930130005**
french	**0.7335113883018494**
world	**0.708888053894043**
european	**0.6885150671005249**
tour	**0.682683527469635**
europe	**0.6817781925201416**
open	**0.6476880311965942**
women	**0.6410533785820007**
jean	**0.6342536211013794**
here	**0.6269105672836304**

Glove 300d Results

Word	Similarity
france	**0.58545982837677**
prohertrib	**0.4812527298927307**
europe	**0.48049768805503845**
french	**0.4337702989578247**
world	**0.42908984422683716**
countries	**0.4197865128517151**
britain	**0.41040974855422974**
european	**0.4019140899181366**
nation	**0.39236509799957275**
london	**0.39168328046798706**

צרפת היא מדינה שבירתה פריז. ציפיתי שחיבור בין "מדינה" ו"פריז" פחות "עיר בירה" תניב לי את המילה צרפת כי פריז היא עיר בירה צרפתית, נוריד ממנה את עיר בירה נשאר עם צרפתית ותוספת של מדינה נקבל צרפת. התוצאות הן אכן בהתאם כמו שניתן לראות france קיבלה בשני המודלים את ההתאמה הגדולה ביותר, במודל ב-300 אף בפער רב מהמקום השני. ניתן לראות שבהמשך ישנם מושגים מאותו עולם תוכן- אירופה שמדינה ופריז נמצאות בהן באותו אופן גם עולם, תיירות שכן צרפת ופריז בפרט הן יעד תיירותי ידוע אך ישנן גם מילים פחות קשורות כמו women,jean,here,open ככל הנראה כי מילים אלו נפוצות בהקשר של צרפת, פריז ומדינות בכללי.

$$happier = happy + (louder - loud)$$

Glove 50d Results

Word	Similarity
happier	**0.8177835941314697**
glad	**0.7704576849937439**
definitely	**0.765666127204895**
thrilled	**0.7633472681045532**
everybody	**0.7495272755622864**
'm	**0.748511552810669**
excited	**0.744789719581604**
happily	**0.7429897785186768**
hopefully	**0.7423971891403198**
surely	**0.7238234281539917**

Glove 300d Results

Word	Similarity
happier	**0.633429229259491**
better	**0.4945078492164612**
hopefully	**0.4866147041320801**
glad	**0.4805920720100403**
pleased	**0.465847373008728**
definitely	**0.4588868021965027**
wish	**0.4549599587917328**
quicker	**0.4514312148094177**
sooner	**0.4495793282985687**
'm	**0.4465576410293579**

happier פירושו "יותר שמח", כאשר אני מחבר את happy עם (louder – loud) אני נשאר רק עם התוספת של היותר, "רועש יותר" פחות "רועש", ועל כן אני מצפה לקבל את happier שכפי שניתן להבחין לפי התוצאות אכן מתקבלת התוצאה הרצויה ואף באחוזים גבוהים. בנוסף ניתן לראות מושגים הקשורים לשמחה בכל מיני צורות, בנוסף מילים באנגלית בתוספת העצמה, כלומר תוספת "er", ועוד נטיות שפה של אנגלית.

$$nephew = man + (niece - woman)$$

Glove 50d Results

Word	Similarity
nephew	**0.9121725559234619**
cousin	**0.9092628955841064**
uncle	**0.8890045285224915**
son	**0.8859953880310059**
granddaughter	**0.869355320930481**
brother	**0.8657659888267517**
grandson	**0.854733407497406**
father	**0.827750027179718**
grandfather	**0.8230489492416382**
daughter	**0.8218050599098206**

Glove 300d Results

Word	Similarity
nephew	**0.7602677941322327**
uncle	**0.7041468620300293**
brother	**0.6975055932998657**
cousin	**0.6511509418487549**
son	**0.6252315044403076**
grandson	**0.620060920715332**
nephews	**0.6125978231430054**
grandfather	**0.596366286277771**
father	**0.5789940357208252**
brother-in-law	**0.5766507983207703**

אחיין הוא בן האח או האחות, לכן היה לי סביר שעבור man עם niece (אחיינית) פחות woman אני אקבל אחיין, ואכן ניתן לראות שהדמיון גדול מאוד והאלגוריתם זיהה טוב מאוד (90% במודל הראשון) את הדמיון וההקשר לאחיין. בנוסף התקבלו מילים רבות אחרות שקשורות למשפחה, סבא, דוד, אח, אחות נכד וכדומה גם הם באחוזים גבוהים יחסית.

חלק ב'- סיווג:

בחלק זה חזרתי על משימת הסיווג, כלומר קיבלתי ערכים באנגלית וערכים באנגלית פשוטה, חילקתי ליחידות סיווג שמורכבות משני משפטים, יצרתי מכל יחידת סיווג Feature_Vector , איזנתי באמצעות SMOTE, עירבבתי בניהם (בפונקציה StratifiedKFold שקיבלה True בפרמטר Shuffle), השתמשתי ב-RandomForestClassifier על מנת לסווג את המשפטים לקורפוס האנגלית ולקורפוס האנגלית הפשוטה (מסווג זה מבצע את עבודתו על ידי עצי החלטה כפי שלמדנו בהרצאה) ולסוף בחנתי את ביצועי הסיווג על ידי ארבעה מדדים – Accuracy, Precision, Recall ו-f1 להם היה מימוש על ידי שימוש ב- make_scorer ו-cross_validate. כלל הפונקציות הובאו מהספרייה sklearn.

את יצירת ה-feature_vectors עשיתי ב-3 דרכים שונות, האחת כאשר לוקטור של מילה ביחידת הסיווג ניתן משקול של 1, השנייה כאשר לכל וקטור של מילה ביחידת הסיווג ניתן משקול של מספר רנדומלי בין 0 ל-5 והשלישית על ידי משקול ספציפי שאני הגדרתי לכל משפט (וקטור של כל מילה נילקח מה- word embeddings המתאים, 50 או 300).

אפרט כעת את הדרך בה בחרתי למשקל כל מילה במשקול שאני הגדרתי:

המשקול הראשון שבחרתי ליישם הוא משקל נמוך יותר ל-100 המילים הנפוצות ביותר בקורפוס המאוחד. המחשבה שלי היא שהמילים הנפוצות ביותר באיחוד הקורפוסים יבדילו הכי פחות טוב בין הקורפוסים שהרי אני רוצה לדעת כמה שיותר טוב לאיזה קורפוס מתאימה כל יחידת סיווג וכאשר מופיעות לי מילים שהן נפוצות מאוד שני בקורפוסים יחד הרי שהמסווג ידע פחות טוב לאיזה קורפוס לסווג. כל וקטור של מילה כזו הכפלתי בסקלאר של 0.5. **התוצאות שקיבלתי השתפרו וזמן הריצה היה תקין לכן השארתי את המשקול.**

הניסיון הבא שעשיתי הוא משקול גבוה יותר ל-100,000 המילים ההכי פחות נפוצות בקורפוס המאוחד. הרציונל שלי במשקול זה היה לתת יותר משקל למילים נדירות שבכל הנראה יהווה גורם מסווג טוב יותר שכן הסיכוי של מילה שמופיעה ממש מעט להיות ספציפית לקורפוס אחד הוא יותר גבוה, מה שיעזור למסווג בעבודתו. בחרתי לבחור מבין 100,000 המילים ההכי פחות נפוצות מפני שכפי שלמדנו בהרצאה (הזנב שראינו לפי חוק זיף) ישנן המון מילים שמופיעות פעם או פעמיים בקורפוס והייתי רוצה שיהיה מנעד כמה שיותר רחב של מילים שיהיה להן משקל גבוה יותר אם הן יסווגו לי טוב יותר את היחידות. כל וקטור של מילה כזו הכפלתי בסקלאר של 5. התוצאות היו בערך אותו הדבר, רק זמן הריצה כמעט הוכפל- לכן החלטתי לא להשאיר את המשקול בקוד.

משקול נוסף שניסיתי שעשיתי הוא לתת משקל גבוה יותר למילים ארוכות יותר, שכן באנגלית הרגילה ישנם ביטויים במשלב לשוני גבוה יותר, כאשר הנטייה של ביטויים במשלב גבוה באנגלית להיות ארוכות יותר מה שעשוי להיות מבדיל טוב בין אנגלית הרגילה לאנגלית הפשוטה. **המשקול תרם במקצת ולא צורך יותר מדי זמן חישוב ולכן השארתי אותו-** כל מילה שאורכה יותר מ-6 הכפלתי את הוקטור שלה בסקלאר של 4, מילה קצרה או שווה באורכה ל-6 הכפלתי את הוקטור שלה ב-2.

ניסיון אחר שעשיתי היה לתת משקל רב יותר למילים ראשונות בכל משפט (כל אחד מהמשפטים ממנו מורכבת יחידת סיווג אחת), הדבר לא הניב תוצאות טובות מדי ולכן החלטתי לוותר על המשקול בצורה הזאת.

אחרי שראיתי שבסופו של דבר הדבר שהכי תרם להעלאת הדיוק הוא בעצם המשקל לכן העלתי את המשקל שאני נותן לכל מילה (כלומר סקלאר לוקטור המילה המתאים) וכל כעת כל מילה שהיא באורך מעל 6 אותיות תקבל משקל של 10, מילה מתחת לאורך שכזה תקבל משקל של 5. בנוסף הורדתי את המשקול של מילה שהיא ב-100 המילים הנפוצות ל0.3

ההבדלים בין שיטות המשקול השונות:

שיטות המשקול השונות התבררו כמוצלחות למדי עם יותר מ-90% אחוזי הצלחה ברובן, עבור שני המודלים.

התוצאות להלן:

Arithmetic mean:
word2vec_50 model performance:
Accuracy: 91.9907
Precision: 91.3798
Recall: 92.7329
F1: 92.0491
word2vec_300 model performance:
Accuracy: 91.2285
Precision: 90.9055
Recall: 91.6354
F1: 91.2603

Random weights:
word2vec_50 model performance:
Accuracy: 93.8287
Precision: 94.4278
Recall: 93.1571
F1: 93.7859
word2vec_300 model performance:
Accuracy: 93.1490
Precision: 93.7639
Recall: 92.4515
F1: 93.1005

My weights:
word2vec_50 model performance:
Accuracy: 91.9483
Precision: 91.3045
Recall: 92.7399
F1: 92.0121
word2vec_300 model performance:
Accuracy: 93.8145
Precision: 94.3180
Recall: 93.2489
F1: 93.7784

ניתן לראות שהתוצאות הטובות ביותר התקבלו עבור המשקלים שאני הגדרתי במודל ה-300, אחרי בדיקות רבות הבנתי שמה שהיה חשוב פה זה המשקול הגדול שנתתי למילים רגילות והמשקול הנמוך שנתתי למילים הנפוצות יותר- כאשר המשקול היה עד 5 בשתי השיטות הראשונות התוצאה לא עברה את ה-93, בשיטה שאני הגדרתי הגעתי ל-94+
ברמת ההשוואה בין שיטות המשקול- את התוצאות הטובות בשני המודלים הביאו המשקלים הרנדומליים, בעיקר בהשוואה למודל ה-50 שבו בשתי השיטות האחרות הוא לא עבר את ה-93, במשקלים הרנדומליים הוא קיבל 93+, התופעה לא כל-כך ברורה לי שכן במודל ה-300 העלאת גודל המשקול שנתתי אכן תרמה לדיוק- במודל ה-50 היא לא עזרה יותר מדי ככל הנראה האפקט של מספרים רנדומיים גדולים בלי חיוביים קטנים מ-1 תורמת יותר למודל ה-50.
ברמת ההשוואה בין שני המודלים- התוצאות במודל ה-300 היו ברובן טובות יותר עם אחוזי דיוק של 92-93 בממוצע לעומת ה-50 שהיה 91-92 בממוצע, כך שהפערים בין שני המודלים לא גדולים מדי והניבו שניהם תוצאות מספקות מאוד.