Who Tweeted What?

עמית סגל - 311340350, בר מועלם - 305794505, רון יצחק - 311604938, אוריין חרמוני - 302170204

7 ביוני 2019

Pre-Preprocessing

התחלנו מחילוק 10 אחוז מכל הטוויטים מכל סלבריטי לסט $test$ בו לא נגענו עד סוף ההאקתון. שנית התבוננו ב־$Data$ וניסינו להבין איך ניתן לפרסר את הדאטא לכדי פיצ'רים מועילים. שמנו לב לדברים הבאים:

1. סטטיסטית, הסיכוי שאדם ימנשן ($mention$ עם @) את עצמו גדול מהסיכוי שימנשן מישהו אחר.
2. ציוצים מחדש ללא תגובה נכתבו יחד עם הטקסט של הטוויט המקורי, המידע הזה פחות רלוונטי, יותר רלוונטי את מי הסלבריטי צייץ מחדש, לכן מזה אנו נפתרים. ציוצים שצוייצו מחדש עם תגובה לעומת זאת, נשמרו עם קישור בסוף.
3. שמנו לב שלרוב למילים יש משמעות נוספת אם מוסיפים אליה את המילים שלצידה, לכן ניסינו להשתמש ב־n-grams.
4. ארנולד שוורצנגר עושה Mention לעצמו בכ־35% מהציוצים שלו!
5. כ־30% מהציוצים של קים קרדשיאן ושל ארנולד שוורצנגר הם Retweet של ציוצים קיימים:

user	handle	self mention %	retweet %
0	@realDonaldTrump	6.77%	22.33%
1	@joebiden	0.29%	13.46%
2	@ConanOBrien	0.40%	0.56%
3	@TheEllenShow	0.59%	1.88%
4	@KimKardashian	5.02%	28.99%
5	@KingJames	9.32%	14.02%
6	@ladygaga	6.65%	17.07%
7	@Cristiano	10.16%	5.67%
8	@jimmykimmel	2.97%	6.91%
9	@Schwarzenegger	34.63%	29.49%

עבור ה־Pre Processing, ביצענו מספר כללי החלטה על מנת לעבד את המידע כך שמכיל רכיבים גנריים יותר, מהם ניתן לחלץ פיצ'רים משמעותיים, כגון:

- הסרת לינקים מציוצים, היות שלא מהווים פיצ'רים מועילים.
- החלפה של סמיילים ואמוג'ים ב־Placeholders על מנת שיפורסרו כמילה יחידה.
- החלפה של Retweets ב־Handle של מי המשתמש לו עשו את ה־Retweet בלבד, שכן שאר המידע לא צויץ בפועל על ידי המשתמש.
- החלפה של תווים מיוחדים כגון סימני קריאה כ־Placeholders על מנת שלא יוסרו על ידי ה־Vectorizers מובנים של sklearn, ונוכל להתייחס אליהם כתווים נפרדים כאשר הם באים ברצף, על מנת שיקבלו משקל גדול יותר.

שיטות שניסינו

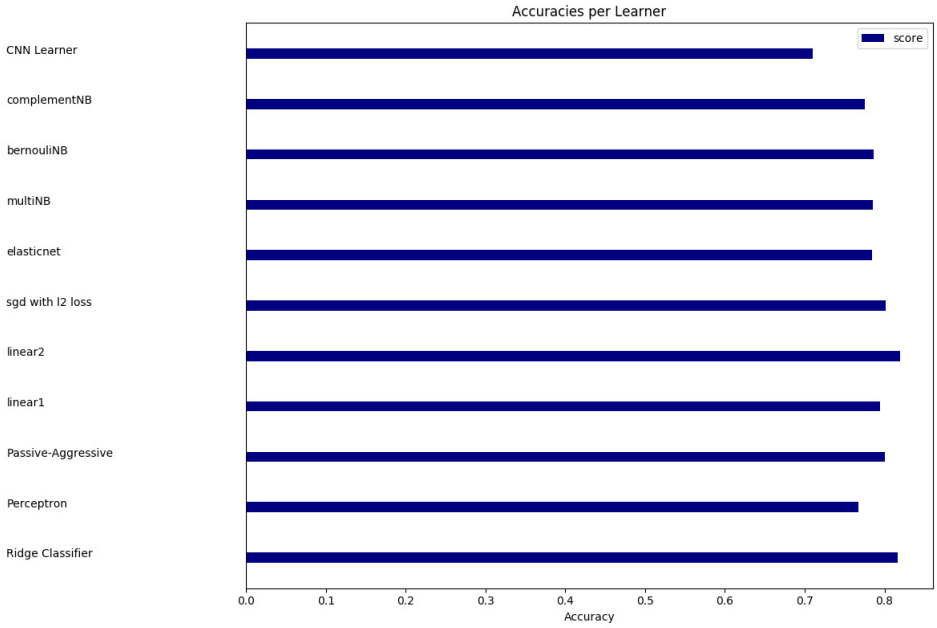
1. שימוש בספריית sklearn, סיננו פ'יצרים לוקטור פ'יצרים באמצעות הפונקציות של הספרייה, ובעזרתם בנינו sparse matrix המתאימה לציוצים, ואותה שלחנו כארגומנט לשלל classifiers שונים של הספרייה.

2. רשת ניורונים המשתמשת ב־preprocessing של החבילה tweet-preprocessing על מנת לפשט את הטקסט. משם נכנסו לרשת קונבולוציונית המתחילה ב־Embedding של Keras ואז לקונבולוציות של $1, 2, 3, 5$-grams מקבילות, ובנוסף במקביל שתי שכבות fully connected, את כל זה משטחים ומעבירים לשכבת היציאה שהיא fully connected עם 10 יציאות.

3. ניסינו גם את האלגוריתם הפשוט של naive-bayes וביצענו smoothing ע"י add-one-smoothing כדי שלא יהיו ערכים ששווים לאפס.

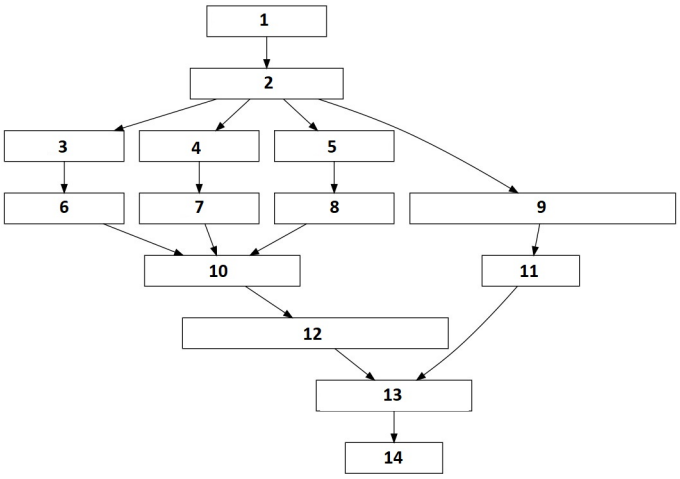
תוצאות

1. להלן ציוני ספריית sklearn:



כאשר הטסט הכי טוב שקיבלנו היה linear2 עם דיוק של 82%.

2. רשת הניורונים: דיוק של 72% על הולידציה, ו־72% גם על סט ה־$test$ בו לא נגענו עד הסוף. מבנה הרשת:



(א) הכניסה לרשת, מערך של ציוצים.

(ב) Embedding - ממיר את הציוצים למספרים.

(ג) קונבולציה של 1-gram, בעומק

(ד) קונבולציה של 2-gram, בעומק 8

(ה) קונבולציה של 3-gram, בעומק 8

(ו) קונבולציה בעומק גדול יותר

(ז) קונבולציה בעומק גדול יותר

(ח) קונבולציה בעומק גדול יותר

(ט) פסודו־שיטוח, כזה שעובד עם input בגודל None

(י) שרשור הקונבולוציות

(יא) שכבת fully connected בגודל 400

(יב) פסודו־שיטוח, כזה שעובד עם input בגודל None

(יג) שרשור נוסף

(יד) fully conneted - היציאה מהרשת.

לבסוף אנו מאמנים את כלל המסווגים ומחזירים את ההיפותזה ששגיאת הטסט שלה היא הנמוכה ביותר.