פרויקט 4: הטמעות מילים ו-POS לסיווג סנטימנטים

שאלת מחקר:
באיזו מידה העשרת WORD EMBEDDING במידע POS משפרת את סיווג הסנטימנטים בטקסטים קצרים ולא פורמליים (למשל, ציוצים), ואילו קטגוריות POS ספציפיות תורמות ביותר לדיוק החיזוי? כלומר המטרה היא לבחון האם שילוב תגי POS עם WORD EMBEDDING משפר את ביצועי סיווג הסנטימנטים.

א. כתוב השערה כיצד שילוב הקשר POS עשוי להועיל לניתוח סנטימנטים.

ב. השתמש במערך נתונים מתויג כמו Yelp או SST (Stanford Sentiment Treebank).
ג. בצע טוקניזציה ולמטיזציה

ד. בצע תיוג POS
ה. נתח התפלגות תגים ושגיאות אופייניות.

ו. האם תוכל להדגים שגיאת תיוג ולהסביר מה הסיבה לכך
ז. בצע WORD EMBEDDING
ח. הטמע את ה POS ב WORD EMBEDDING
ט. בנה מסווג רשת נוירונים פשוטה והפעל אותו על שני ה EMBEDDINGS
י. השווה את התוצאות באמצעות ACCURACY, F1, PRECISION

שם: אלכסנדר גינזבורג

תז: 208839613

קורס: עיבוד שפה טבעית

מרצה: ד"ר שרון ילוב הנדזל

שאלת מחקר

באיזו מידה העשרת EMBEDDING WORD במידע POS משפרת את סיווג הסנטימנטים בטקסטים קצרים ולא פורמליים)למשל, ציוצים(, ואילו קטגוריות POS ספציפיות תורמות ביותר לדיוק החיזוי? כלומר המטרה היא לבחון האם שילוב תגי POS עם WORD EMBEDDING משפר את ביצועי סיווג הסנטימנטים.

פרויקט זה בוחן האם העשרת ה Word -Embeddings עם תגיות POS משפרת את סיווג הסנטימנטים בטקסטים קצרים כמו בציוצים ומזהה את הקטגוריות של POS התורמות ביותר לדיוק החיזוי. המחקר משתמש ב-1,328 ציוצים שהושגו באמצעות SCRAPER על עמודים עם הS`URL הבאים:

1) https://x.com/JewishWarrior13

2) https://x.com/sentdefender

(160 חיוביים, 1,168 שליליים) ו-1,000 ביקורות Yelp (מאוזנות 0/1) כדי לענות על שאלת המחקר: האם שילוב תגיות POS עם Word Embeddings משפר את ביצועי סיווג הסנטימנטים

השערה

שילוב מידע על תפקידי המילים (POS) בEmbeddings שלהם יעזור לסווג סנטימנטים בצורה טובה יותר, במיוחד בטקסטים לא פורמליים, על ידי הבנת ההקשר של המילים. למשל, המילה "טוב" יכולה להיות תואר שמבטא משהו חיובי או תואר הפועל שיכול להיות נייטרלי, וזה עשוי לעזור לתקן ולהשפר את חיזוי המודל.

מתודולוגיה

מאגרי נתונים :

- ציוצים: 1,399 ציוצים שנאספו לאחרי ניקוי נשאר כ-1,328 לאחר הסרת 71 רשומות ריקות.
- Yelpביקורות YELP: 1000ביקורות מ ,yelp_polarity-מעובדות בריצות 50D ו-300D

עיבוד מקדים

- spaCy : Tokenization
- spaCy:Lemmatization
- תיוג POS - en_core_web_sm

ניתוח POS

- בחינת pos וזיהוי שגיאות אפשריות (למשל, המילה "טוב" עלולה להיתפס בטעות כ-ADV במקום תואר). (מצורפת דוגמה מטה עבור יחס YelpDim50 vs SSTDim50)
-

Embedding

- שימוש ב-GloVe עם ממדים של D50 ו-D300 עבור Word Embeddings, תוך שילוב עם POS Embeddings One-hot.

סיווג

- בניית רשת נוירונים פשוטה ב-Keras, עם 10 epochs עבור Yelp ו-20 עבור ציוצים, חלוקה ל-80/20 בין אימון ובדיקה, ויישום משקולות כיתה עבור ציוצים.

מדידת תוצאות

- מדידה באמצעות Accuracy, F1, Precision, ומטריצות בלבול מנורמלות.

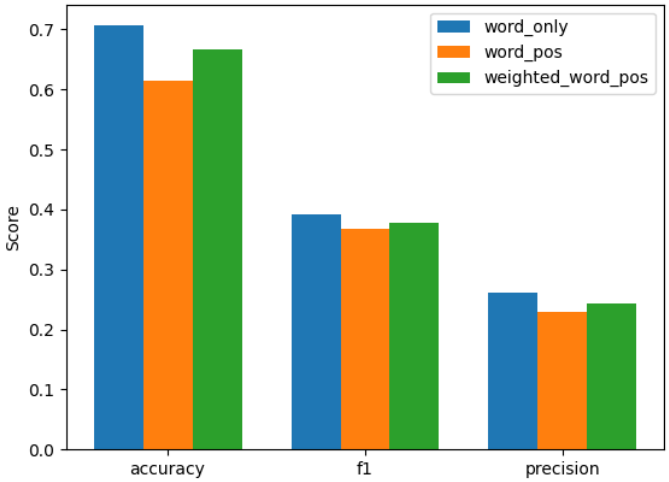
תוצאות

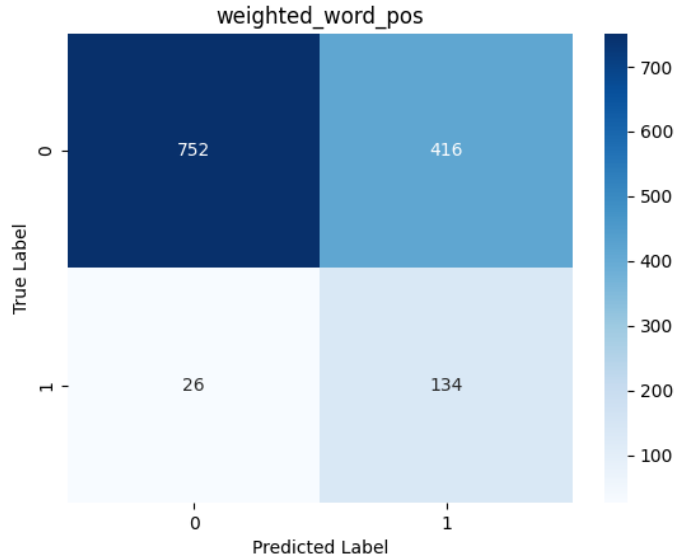
Dataset	Dimension	Model	Accuracy	F1	Precision
Tweets	50D	Word-only	0.7063	0.3925	0.2614
Tweets	50D	Word+POS	0.6152	0.3668	0.2287
Tweets	50D	Weighted word+POS	0.6672	0.3775	0.2436
Tweets	300D	Word-only	0.9239	0.7292	0.6385
Tweets	300D	Word+POS	0.8396	0.5912	0.4266
Tweets	300D	Weighted word+POS	0.8404	0.5891	0.4270
Yelp	50D	Word-only	0.8299	0.8284	0.8355
Yelp	50D	Word+POS	0.8330	0.8348	0.8259
Yelp	50D	Weighted word+POS	0.8323	0.8307	0.8386
Yelp	300D	Word-only	0.8919	0.8946	0.8732
Yelp	300D	Word+POS	0.8932	0.8955	0.8765
Yelp	300D	Weighted word+POS	0.8922	0.8945	0.8757

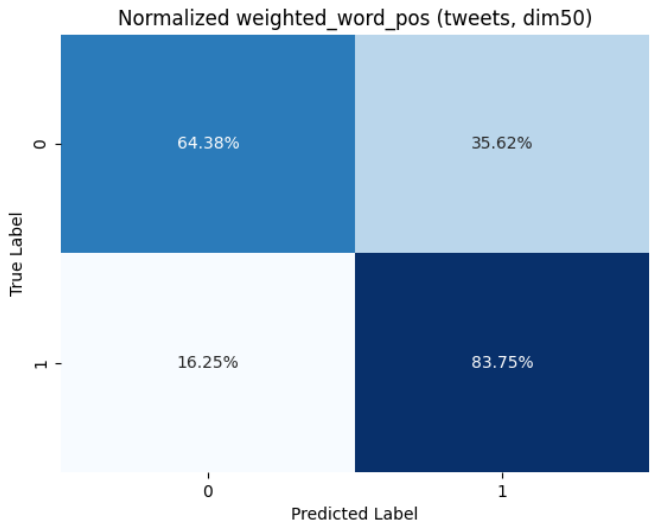
השפעת POS

בציוצים עם 300D, נרשמת עלייה בולטת ב-F1 כאשר משתמשים ב-Word-only (+33.67%), אבל מודלי הPOS מראים ביצועים נמוכים יותר בהשוואה ל-50D. ב-Yelp עם 300D, יש שיפור של +6.07% ב-F1 כאשר משלבים Word+POS (מ-0.8348 ל-0.8955).

תוצאות tweets - 50D

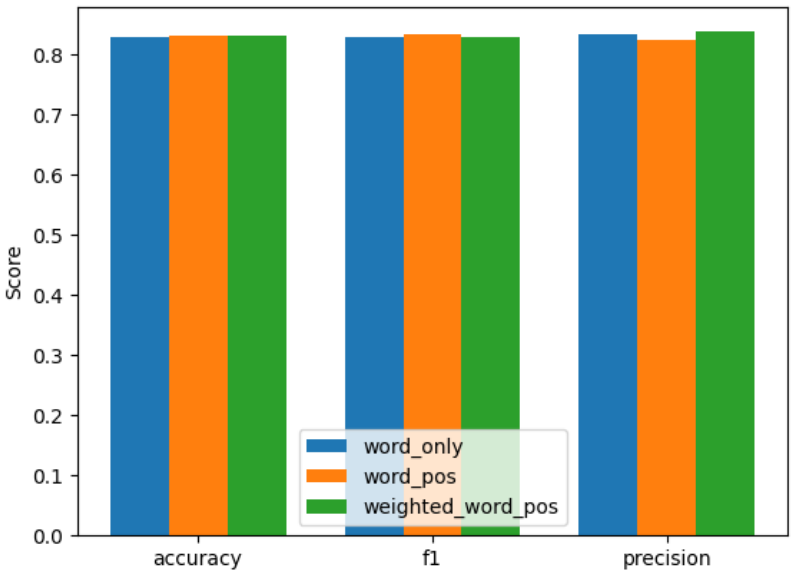


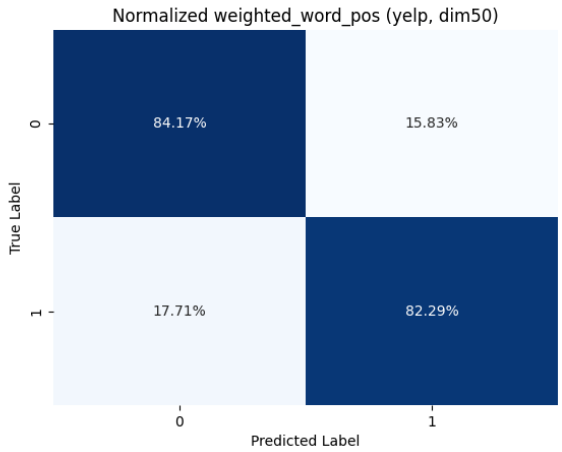


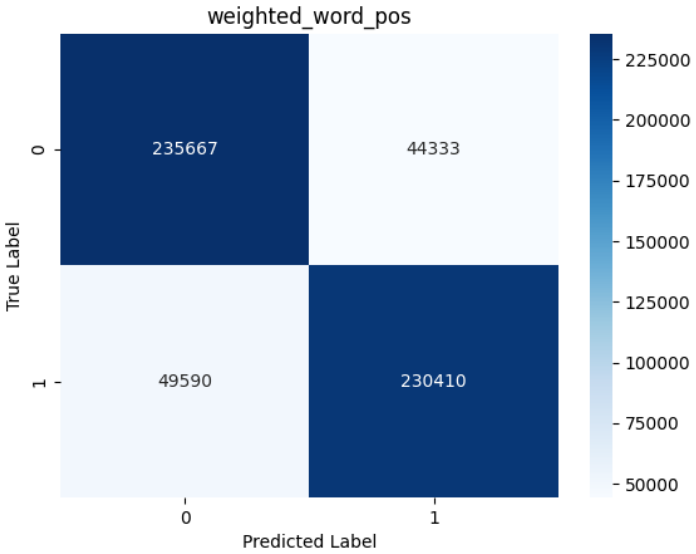


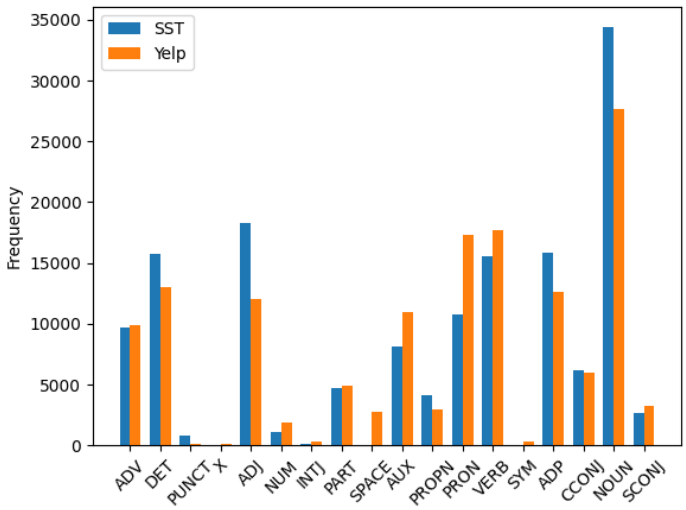
- השוואה בין Accuracy, F1 ו-Precision של המודלים Word-only, Word+POS ו- Weighted Word+POS עבור ציוצים עם D50.
- Confusion matrix משתמש בתיוג בינארי:00,01,10,11:
  - True Negative=00
  - False Positive=01
  - False Negative=10 ,
  - Positive True Positive=11 ,
- **<u>הערה:</u>** פורמט זהה יוצג לשאר התוצאות מהטבלה מעלה

תוצאות Yelp - 50D

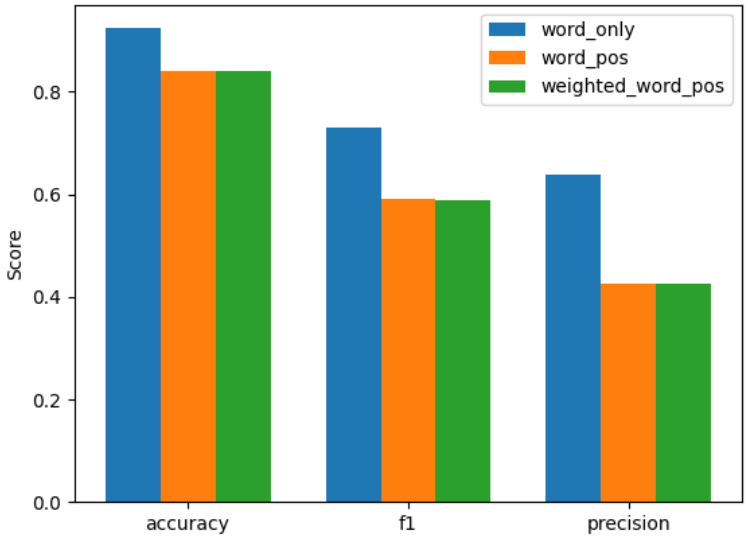


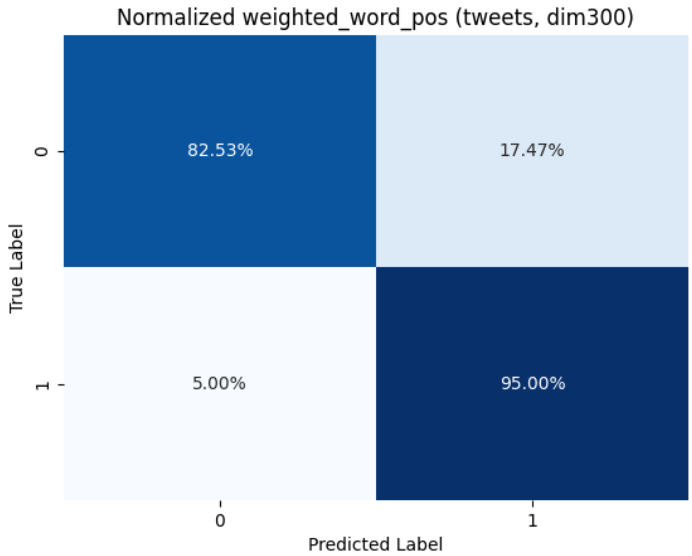


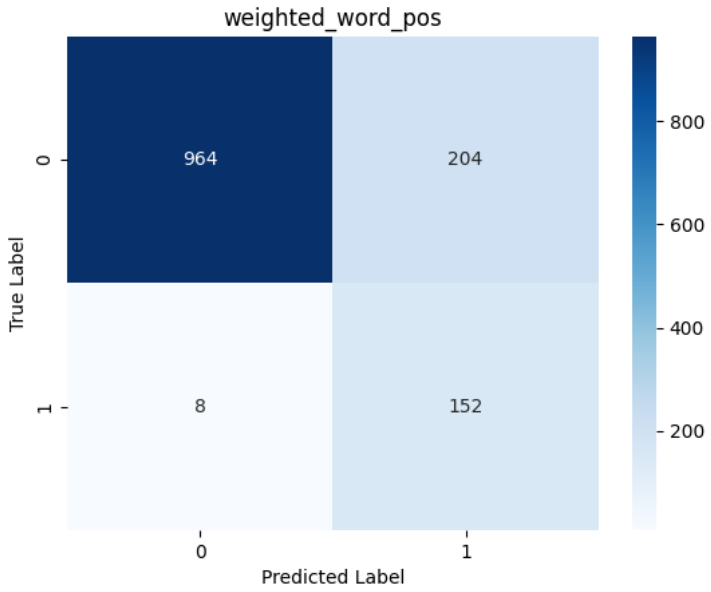




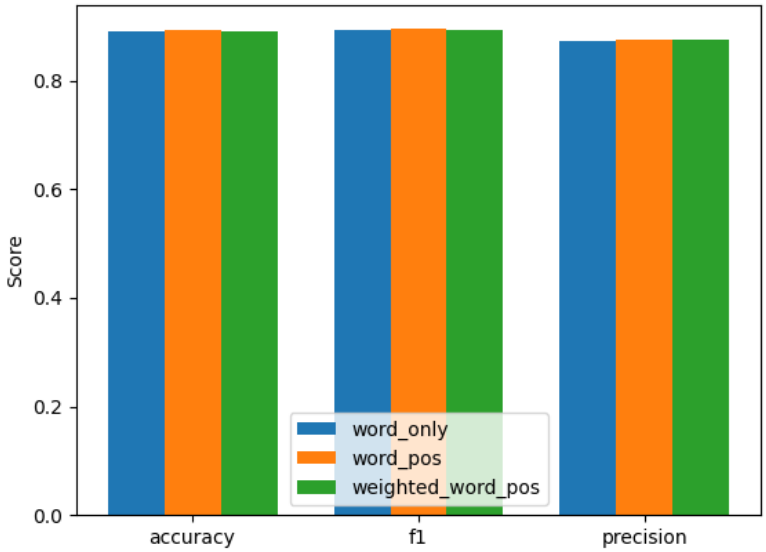
תוצאות tweets - 300D

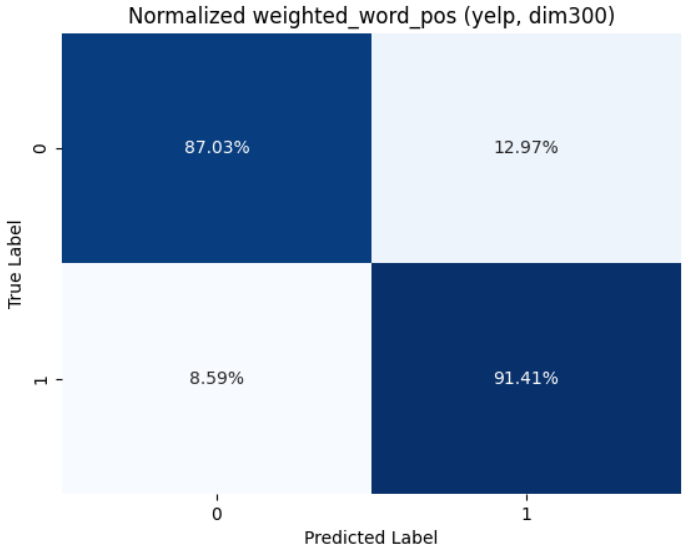


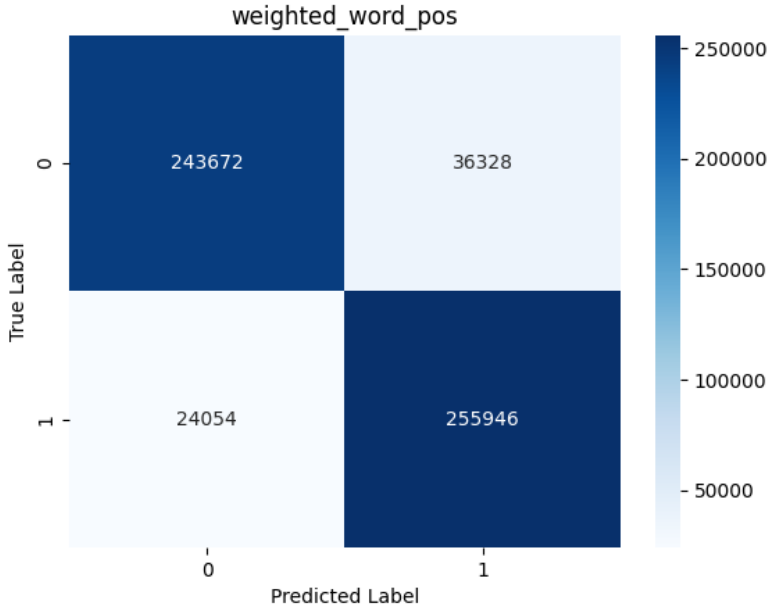




תוצאות Yelp - 300D







הצעה לשינוי אלגוריתם

ניתן לבצע שימוש במודל LSTM במקום רשת נוירונים פשוטה ב-Keras.

שיקול לכך הוא:

המודל הנוכחי מתקשה לזהות סדר וקשרים ארוכי טווח בטקסטים, במיוחד בציוצים קצרים. LSTM יכול לשפר את הבנת ההקשר על ידי התמקדות בדקויות של מילים ומיקומן, זה חשוב במיוחד עבור ציוצים, שבהם הסדר של המילים (למשל, "לא טוב" לעומת "טוב לא") משפיע יותר מאשר בטקסטים ארוכים, מה שעשוי להעלות את המדדים.

ביצוע שינוי זה גם לא מורכב, נדרש יהיה לבצע עדכון פונקציית build_model ולהריץ את הPipeline מחדש.