NLP Final Project

Erez Lavi

בחרתי לבצע את פרויקט 2 – השוואת סנטימנט לאורך זמן במודל LSTM לעומת Transformer.

מתודולוגיה

1. בחירת מערך נתונים 50 אלף ביקורות IMDB.
2. EDA בסיסי של הדאטה ופיצול למדגם מאוזן של 5 אלף ביקורות חיוביות ושלילות.
3. טוקניזציה, למטיזציה ותיוג POS בעזרת NLTK.
4. ביצוע Embedding בעזרת Word2Vec skip-gram.
5. בניה ואימון מודלים LSTM ו BERT.
6. השוואת ביצועי המודלים על ידי f1, accuracy, confusion matrix בהרצה ראשונה.
7. שינוי סדר האימון של הדאטה ואימון מחודש של 2 המודלים.
8. השוואה בין התוצאות של המודלים לפני ואחרי שינוי הסדר.
9. בדיקה ידנית של המודלים עם ביקורות לדוגמה.

אופן בניית ואימון המודלים

חלוקה 80% אימון ו20% בדיקות עבור 2 המודלים

חשוב לציין שלאחר הצגת התפלגות אורך הטוקנים בביקורת החלטתי להגדיר padding אחיד באורך 200 עבור המודלים, כלומר ביקורות ארוכות במיוחד נחתכו, זה עזר לחסוך במשאבי חישוב של הGPU.

LSTM

ראשית כל, המודל קיבל כקלט skip-gram Embedding matrix שנבנתה מראש על בסיס הטוקנים של הביקורות לאחר למטיזציה. זו שיטה שאמורה לתת נקודת פתיחה למודל וביצועים טובים יותר, מכיוון שהווקטורים שמתקבלים כקלט אינם אקראיים ומכילים כבר קשרים סמנטיים. לאחר כמה ניסויים החלטתי להגדיר 5 epochs ו64 batch size,מעבר לזה לרוב נוצר מצב של התאמת יתר (אבל לעתים גם ניתנה רמת דיוק גבוהה יותר).

BERT

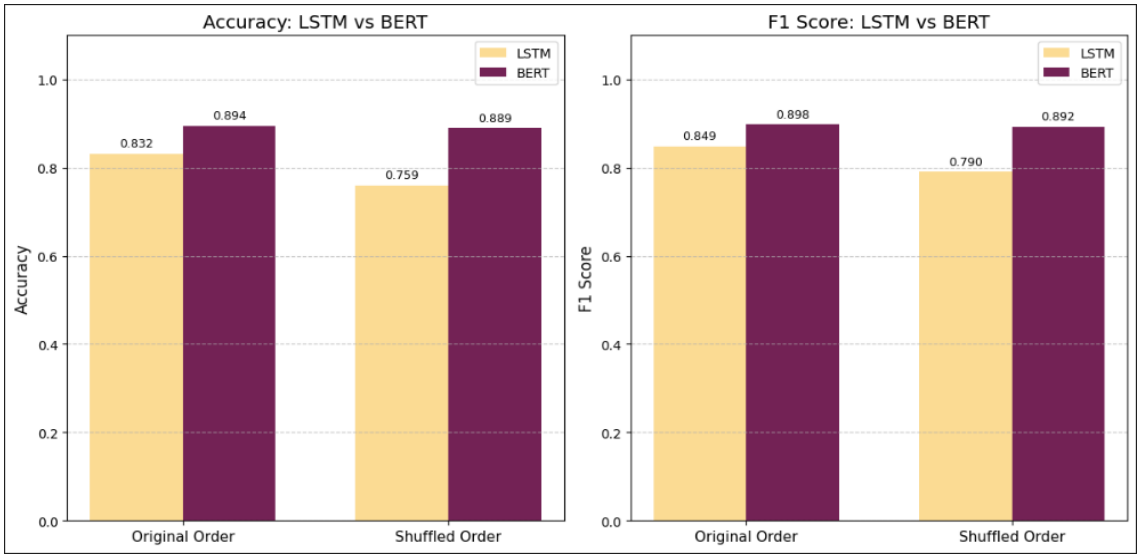
כאן בניית המודל התבצעה בצורה מעט שונה, המודל קיבל כקלט את הביקורות כרשימת מחרוזות ועבר את תהליך הטוקניזציה על ידי הtokenizer של BERT. השתמשתי בסוג **DistilBERT** . זמן האימון שלו היה ארוך פי 10 מאשר ב LSTM אף על פי שנמשך על ידי 2 epochs בלבד (learning rate 2e-5 עם adam optimizer).

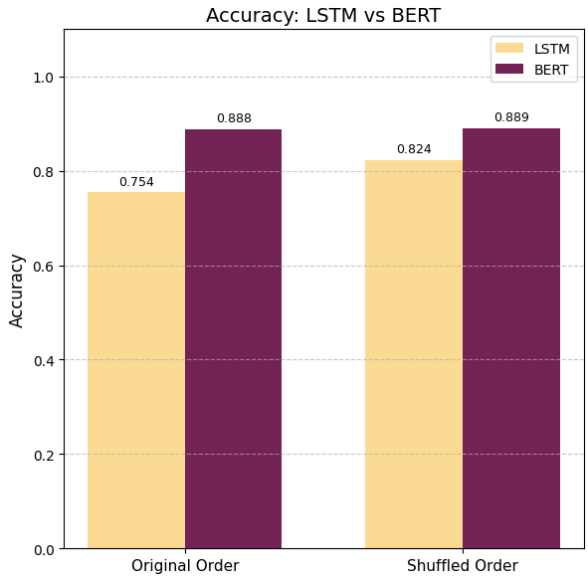
ההסבר לזמן הריצה קשור בין היתר לכך שמבנה הארכיטקטורה שלו שונה (טרנספורמר) ומבוסס על self attention שזה אלגוריתם שעובד בסיבוכיות ריבועית - הוא מחשב את הקשר בין כל זוג טוקנים בקלט. זאת בהשוואה למודל LSTM שרץ בסיבוכיות לינארית לגודל הקלט.

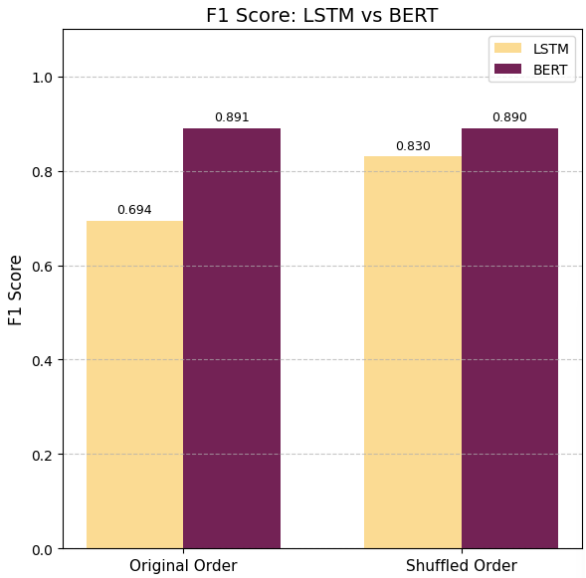
ניתוח תוצאות

כפי שניתן לראות בגרף למטה, השינוי בסדר הקלט באימון המודלים השפיע בעיקר על המודל LSTM. לעומת זאת, מודל BERT לא הושפע מכך בכלל ושמר על עקביות יחסית בין שני האימונים. **אבל** השאלה המתבקשת היא - האם השוני בתוצאות טמון באופן שבו שני המודלים בנויים?

מודל BERT באופן כללי הוא מודל יציב יותר, אומן על כמויות דאטה עצומות ולכן שינוי מהסוג הזה ככל הנראה לא אמור להשפיע על התוצאה מה שמתכתב גם עם התוצאות. LSTM הוא מודל קטן יותר ולכן הגיונית כל שינוי אקראי בו עשוי לגרור תזוזה משמעותית יותר ברמת הדיוק בתוצאות.





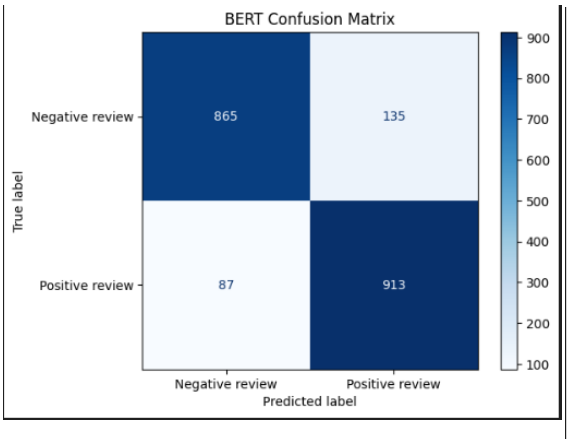


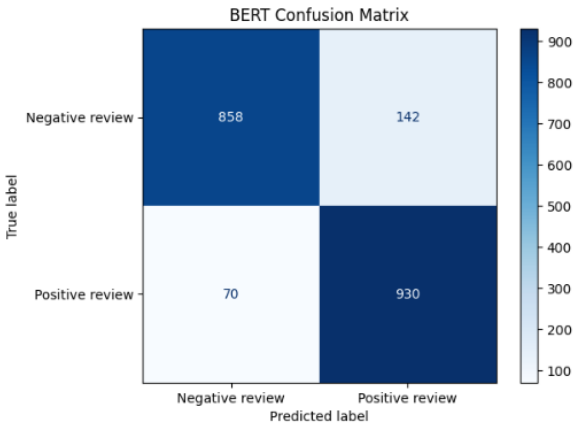
כאן בגרף שלמעלה ניתן לראות בהרצה אחרת של התוכנית כי התקבלו תוצאות הפוכות.. באימון הראשוני של הLSTM התקבל ציון נמוך יותר ולאחר שינוי הסדר גבוה יותר, יחד עם זאת גם כאן הBERT נתן תוצאות כמעט זהות להרצות קודמות.

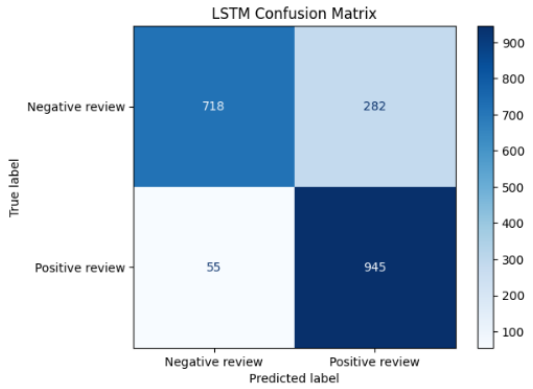
מכל הנתונים שהתקבלו, **המסקנה הכללית** היא **שLSTM** יותר משתנה בין אימונים לפי משתנים כמו סדר הדאטה, המשקולות האקראיות ההתחלתיות, מדגמים אקראיים שנלקחים פר epoch ועוד תהליכים סטוכטיים שמשפיעים על הביצועים.

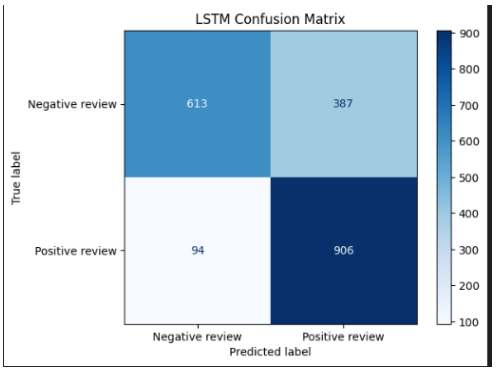
לעומתו מודל **BERT** משמעותית יותר יציב, הרוב המוחלט של המשקלים קבוע מראש, רק אחוז קטן מושפע מנתוני האימון החדשים, ניתן להגיד שזה יותר fine tuning מאשר אימון ולכן גם ללא שום דאטה ככל הנראה שהמודל היה מגיע לביצועים די מרשימים.

בconfusion matrices למטה ניתן לראות שבאופן כללי LSTM מזהה באחוז גבוהה יותר באופן שגוי ביקורות שליליות כחיוביות (false negatives) לעומת BERT.









בדיקה על ביקורות לדוגמה

בסוף התוכנית, הרצתי את המודלים על כמה דוגמאות. ניתן לראות שעבור הדוגמה הראשונה שהיא פשוטה מאוד, כל המודלים ידעו לחזות בקלות שמדובר בביקורת חיובית. דוגמה 2 היא גם יחסית קלה אבל יש בה ניגוד במשפט, נראה שהמודלים ברובם הצליחו לזהות נכון. 2 הדוגמאות האחרונות הן יותר טריקיות, לבסוף ניתן לראות שעבור הדוגמה האחרונה BERT חזו נכון אבל LSTM לא, וזה מתכתב גם עם רמות הדיוק בתוצאות הסופיות.

```
=== Sample 1: "I loved the performance!" ===
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
[LSTM] Original: Positive (0.911) | Shuffled: Positive (0.623)
[BERT] Original: Positive (0.967) | Shuffled: Positive (0.980)

=== Sample 2: "This movie was realy not good!" ===
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
[LSTM] Original: Positive (0.558) | Shuffled: Negative (0.482)
[BERT] Original: Negative (0.022) | Shuffled: Negative (0.025)

=== Sample 3: "Oh, this movie was a masterpiece... of boredom." ===
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
[LSTM] Original: Positive (0.793) | Shuffled: Negative (0.422)
[BERT] Original: Positive (0.652) | Shuffled: Positive (0.727)

=== Sample 4: "The movie tried so hard to be clever, it ended up being exhausting" ===
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
[LSTM] Original: Positive (0.931) | Shuffled: Positive (0.685)
[BERT] Original: Negative (0.024) | Shuffled: Negative (0.024)
```

סיכום

לפי דעתי ניתוח סנטימנט הוא שימוש מעניין מאוד לתחום של עיבוד שפה טבעית ויש לו יישומים פרקטיים רבים - במערכות המלצה, שיווק, שירות לקוחות, שוק ההון ועוד..

לכן היה מרתק ליישם זאת בעזרת שני מודלים שלמדנו במהלך הקורס, ולנסות לבחון כיצד שינויים בקלט עשויים להשפיע על הדיוק והביצועים שלהם.

שאלה מסקרנת לטעמי או הצעה לשינוי היא:

האם ניתן יהיה בעתיד או אפילו כיום להגיע לרמות הדיוק של מודלי ה Transformers אבל על ידי מודלים יעילים יותר (סיבוכיות לינארית) עבור מקרי שימוש ספציפיים יחסית, כמו למשל ביקורות סרטים?