**זיהוי רגשות בשפה העברית**

מגיש: נאור סאסי 302727052

**מבוא**

עיבוד שפה טבעית (NLP) נמצא בשימוש נרחב בתחומים רבים כגון ניתוח סנטימנטים, תרגום וכו'.

סיווג רגשות הוא אחד ממשימות ה-NLP המאתגרות ביותר בשל האופי הסובייקטיבי של הרגשות והקושי לזהות רגשות מנתונים טקסטואליים. זיהוי רגשות אלו יכול להראות התנהגויות צפויות, מצבים נפשיים, וכ"ו בהם שרויים אנשים. בעבודה זאת אבדוק את הרגשות(תוחלת, שמח, אמון, פחד, הפתעה, עצב, גועל, כעס ) בשפה בעברית באמצעות AlephBERT.

לצורך הניתוח השתמשתי ב-3 מערכי נתונים. אחד מערך נתונים לאימון (הערות שפורסמו במאמרי חדשות ynet, ישראל היום, ובחדרי חדרים הקשורים לקורונה). השני בדיקה המכילה קבוצה אקראית של הערות שנלקחו ממערך הנתונים(הערות הקשורים לקורונה). השלישי בדיקה קבוצה אקראית של הערות שפורסמו בתגובה למאמרים שאינם קשורים לקורונה אבל מאותם אתרי חדשות.

**אתגר**

בהתחלה ניתחתי את מערך הנתונים והבנתי שמערך הנתונים לא מאוזן בעל מספר רגשות לכל דוגמא. יש רגשות עם פחות מ-200 דוגמאות (שזה נמוך מאוד ברוב משימות הסיווג של למידת מכונה) ויש כמה רגשות בסביבות ה-400 דוגמאות ויש רגש עם מעל 1000 דוגמאות. עם זאת אני ממשיך לראות את הדיוק שאפשר להביא ממערך נתונים זה.

בנוסף אבדוק על מערך נתונים שבו כל דוגמא יש רגש יחיד.

**גישה**

אני התחלתי עם עיבוד הטקסט שיהיה מותאם אישית: הורדת נתונים כפולים ועמודות שאני לא משתמש (ע"י המחברת DATA\_NLP).

אני מחלק את האימון ל-2 . אחד לימוד על הנתונים לאחר שהטקסט עבר התאמה אישית ושימוש ב- tokenizer של AlephBERT. השני לימוד על נתונים לאחר התאמה אישית ושהורדנו קישורים, כתובות אתרי אינטרנט, תווים מיוחדים ומילות עצירה(stop-word).

עבור הסיווג אני משתמש ב:

רגרסיה לוגיסטית (Logistic Regression) – למידה בסיסית פשוטה יחד עם מציאת ההיפר-פרמטרים הטובים ביותר ע"י GridSearch.

יער אקראי (Random Forest), סיווג וקטור תמיכה לינארי (Linear SVC) וערמתם ביחד באמצעות StackingClassifier.

למידה עמוקה – למידה באמצעות מודל AlephBERT בלי כיוון ועם כיוון עדין.

**תוצאות**

תוצאות המשימות המבוססות משפטים של ניתוח רגשות:

טבלה איכות המודל לפי מדד F1-score על הנתונים לאחר שהטקסט עבר התאמה אישית ושימוש ב- tokenizerשל AlephBERT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AlephBERT  עם תוספת שכבות למידה | AlephBERT  בלי תוספת שכבות למידה | Random Forest+ Linear SVC | Logistic Regression |  |
| 0.94 | 0.99 | 0.53 | 0.6 | Train |
| 0.42 | 0.42 | 0.44 | 0.45 | Val |
| 0.33 | 0.29 | 0.27 | 0.27 | Test insample |
| 0.66 | 0.71 | 0.62 | 0.62 | Test outsample |

טבלה איכות המודל לפי מדד F1-score על הנתונים לאחר שהטקסט עבר התאמה אישית ושהורדנו קישורים, כתובות אתרי אינטרנט, תווים מיוחדים ומילות עצירה(stop-word)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AlephBERT  עם תוספת שכבות למידה | AlephBERT  בלי תוספת שכבות למידה | Random Forest+ Linear SVC | Logistic Regression |  |
| 0.4 | 0.4 | 0.52 | 0.59 | Train |
| 0.4 | 0.4 | 0.41 | 0.44 | Val |
| 0.22 | 0.22 | 0.05 | 0.23 | Test insample |
| 0.67 | 0.67 | 0.01 | 0.61 | Test outsample |

המסווג מבוסס AlephBERT הנלמד לעומק הפיק דיוק טוב יותר מאשר רגרסיה לוגיסטית ויער אקראי עם סיווג וקטור תמיכה לינארי.

AlephBERT בלי תוספת שכבות למידה לפי מדד F1-score.

|  |  |  |
| --- | --- | --- |
| רגש | insample | outsample |
| תוחלת | 0.31 | 0.60 |
| שמח | 0.00 | 0.00 |
| אמון | 0.00 | 0.00 |
| פחד | 0.00 | 0.00 |
| הפתעה | 0.00 | 0.00 |
| עצב | 0.00 | 0.00 |
| גועל | 0.38 | 0.84 |
| כעס | 0.00 | 0.00 |

AlephBERT עם תוספת שכבות למידה לפי מדד F1-score.

|  |  |  |
| --- | --- | --- |
| רגש | insample | outsample |
| תוחלת | 0.44 | 0.41 |
| שמח | 0.00 | 0.00 |
| אמון | 0.00 | 0.00 |
| פחד | 0.16 | 0.00 |
| הפתעה | 0.00 | 0.00 |
| עצב | 0.06 | 0.00 |
| גועל | 0.37 | 0.82 |
| כעס | 0.00 | 0.00 |

AlephBERT על המידע שכל דוגמא יש רגש יחיד (קובץ NLP\_NEW)

|  |  |
| --- | --- |
| emotion | F1-score |
| joy | 0.69 |
| fear | 0.90 |
| surprise | 0.62 |
| sadness | 0.79 |
| love | 0.82 |
| anger | 0.90 |

**מסקנה**

בעקבות ניקיון המידע ושכל משפט יכול להיות עם מספר רגשות המידע לא מאוזן.

הלמידה בוצע על רגש אחד לכל משפט אשר נבדק ללא הצלחה טובה.

לא הצלחתי לזהות את הרגשות עם המספר דוגמאות המופחת בלמדה ללא תוספת שכבות.

לעומת זאת הצלחנו עם תוספת שכבות לזהות עוד 2 רגשות אבל סה"כ זיהוי לא מוצלח.

בדיקה על מידע עם משפטים מפורשים לרגש בודד הניב תוצאות טובות.

יש צורך לנתח את המשפטים עם מספר ליבלים (label).

**מקורות**

המידע שהשתמשתי לאימון ובדיקת איכות המודל<https://github.com/avichaychriqui/HeBERT/blob/main/data.zip>

[**https://huggingface.co/onlplab/alephbert-base**](https://huggingface.co/onlplab/alephbert-base)

[**https://arxiv.org/abs/2104.04052**](https://arxiv.org/abs/2104.04052)

**פירוט על העבודה**

ספריית Transformers מספקת מגוון רחב של דגמי Transformer כולל AlephBERT .

זה עובד עם TensorFlow ו- PyTorch זה כולל גם tokenizers שבניתי מראש שעושה עבורי את המשימות.

השתמשתי ב- AlephBERT הבסיסי ובניתי על גבי סיווג רגשות.

טעינת המודל :

alephbert = BertModel.from\_pretrained('onlplab/alephbert-base')

דרישות AlephBERT :

הוסף אסימונים מיוחדים להפרדת משפטים

העבר רצפים באורך קבוע (הכנס ריפוד)

מערך של 0 (אסימון משטח) ו-1 (אסימון אמיתי) הנקרא attention mask.

טעינה של BertTokenizer שמאומן לפני.

בזה השתמש כדי להמיר טקסט ל- Tokens ואז מ-Tokens למספרים שלמים ייחודים (IDS).

alephbert\_tokenizer = BertTokenizerFast.from\_pretrained('onlplab/alephbert-base')

ישנם Tokens מיוחדים :

[SEP] – סימון לסיום משפט

[CLS] - הוספנו את Token הזה לתחילת כל משפט, כדי ש- AlephBERT ידע שאני עושה סיווג.

[PAD] – הסימון מיוחד לריפוד

[UNK] - AlephBERT מבין Tokens שהיו בסט האימונים. כל השאר קודדתי בלא ידוע.

אני השתמשתי במתודה encode\_plus אשר עושה את כל העבודה הזאת.

    encoded\_review = alephbert\_tokenizer.encode\_plus(

        text,

        max\_length=MAX\_LEN,

        add\_special\_tokens=True,

        return\_token\_type\_ids=False,

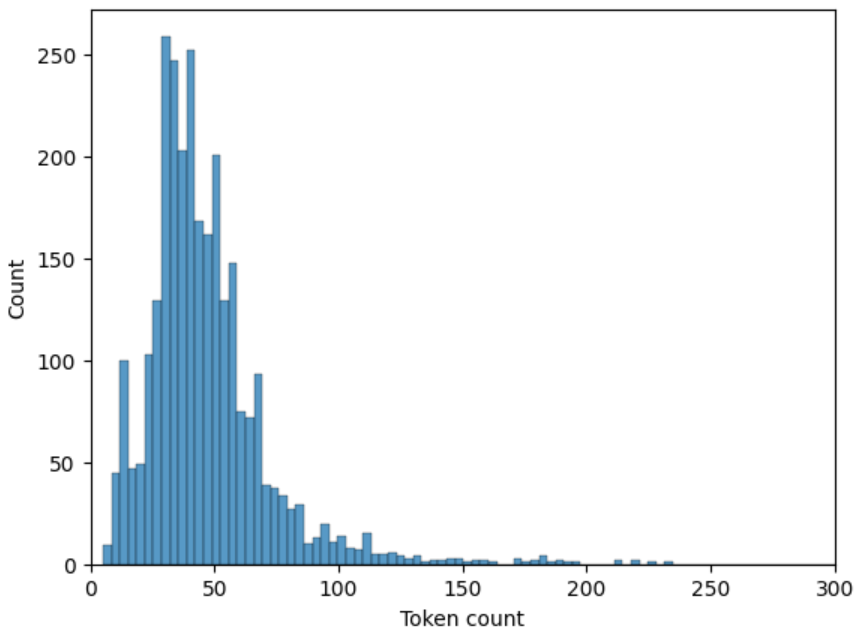
        pad\_to\_max\_length=True,

        return\_attention\_mask=True,

        return\_tensors='pt',

    )

ניתן לראות שרוב הביקורות מכילות פחות מ-80 Tokens אבל בחרתי 200 כדי להיות הכי בטוח.



GPReviewDataset – יצירת מערך נתונים של PyTorch.

SentimentClassifer – השתמשתי בו במודל של ניתוח סנטימנטים והעברתי אותו לניתוח רגשות עם שינויים קטנים .זה יצירת מודל בסיסי בשימוש ב AlephBERT.

EmotionClassifier – מודל לזיהוי רגשות עם תוספות שכבות.

.to(device) להעביר לgpu.

device = torch.device("cuda:0" if torch.cuda.is\_available() else "cpu")

AdamW – אופטימיזציה עם ירידת מששקל קבועה שניתן להשתמש בה לכוונון עדין של מודלים.

Train\_epoch – אימון המודל שלנו לepoch אחד.

Evel\_model – הערכת המודל.