פרויקט בבינה מלאכותית

נושא הפרויקט:

סיווג פרסומים העוסקים בנושא אלימות במשפחה ומערכות יחסים אלימות כקריטיים או לא קריטיים

מגישות

גל קסטן – 316353176, galkesten@campus.technion.ac.il

דנית כהן – 318263803, danitcohen@campus.technion.ac.il

.

מבוא

אלימות במשפחה, יכולה להיות מוגדרת [1] ,כדפוס התנהגות במערכת יחסים המשמש להשגה או שמירה של כוח ושליטה על בן זוג אינטימי. ההתעללות מתבטאת בפעולות פיזיות, מיניות, רגשיות, כלכליות או פסיכולוגיות המשפיעות על אדם אחר. היא כוללת את כל ההתנהגויות שמפחידות, מאיימות, מתמרנות, משפילות, מאשימות, פוצעות או פוגעות במישהו. התעללות במשפחה יכולה לקרות לכל אחד מכל גזע, גיל, נטייה מינית, דת או מין.
היא יכול להתרחש במגוון מערכות יחסים כולל זוגות נשואים, זוגות החיים יחד או זוגות בתחילת דרכם. אלימות במשפחה משפיעה על אנשים מכל רקע חברתי -כלכלי ומכל רמות ההשכלה. קורבנות של התעללות במשפחה עשויים לכלול גם ילד או קרוב משפחה אחר, או כל בן בית אחר.
במחקר שנערך ע"י ארגון הבריאות העולמי [2], נמצא כי כמעט אחת מכל שלוש נשים בעולם הייתה קורבן לאלימות פיזית ו/או אלימות מינית מצד בעל קודם או נוכחי או בן זוג אינטימי לפחות פעם אחת בחייה. זה אומר שכ-642 מיליון נשים בעולם מגיל 15 ומעלה סבלו מאלימות במשפחה במהלך חייהן.
במחקר נוסף שנערך ע"י UNODC [3] נמצא כי כ- 137 נשים בעולם נהרגות בכל יום ע"י בן משפחה שלהם.
אלימות במשפחה מופנית בדרך כלל כלפי נשים וילדים, עם זאת גם גברים עשויים לסבול מאלימות במשפחה. על פי [4] , 1 מתוך ארבעה גברים בארה"ב סבל מאונס, אלימות מילולית ו/או מעקב ע"י בן זוג אינטימי במהלך חייו.

בשנים האחרונות, אתרי הרשתות החברתיות חדרו לחייהם הפרטיים של אנשים ברחבי העולם.
יותר ויותר אנשים מוצאים מפלט ברשת , בו הם בוחרים לשתף את סיפורם האישי ומצוקתם בפורומים שונים וברשתות חברתיות כמו Reddit, Twitter, Facebook .
ניתן למצוא ברשת פורומים שונים המבקשים לתת תמיכה וסיוע לאנשים שנמצאים במצוקות שונות כמו : דיכאון, בעיות במערכות יחסים, מחשבות אובדניות, קורבנות אונס, קורבנות אלימות ועוד.
בין היתר יש ברשת האינטרנט אינספור קבוצות תמיכה עבור אנשים הסובלים מאלימות במשפחה. קבוצות אלו מכילות מידע רב על חייהם הפרטיים של אנשים. בכל יום מתפרסמים ברשת האינטרנט מספר עצום של הודעות המתארות סיטואציות קשות- אנשים שסובלים מאלימות פיזית ומילולית מצד בן זוגם ונמצאים בסכנה ממשית לחייהם או אנשים שסבלו בעבר מאלימות ושרויים בדיכאון ובמצוקה נפשית חמורה.
המידע הרב שמכילה הרשת החברתית עשוי לסייע לארגוני חירום וסיוע שונים המטפלים באלימות במשפחה לאתר קורבנות אלימות חדשים הנמצאים במצב קריטי ולהציע להם סיוע.
עם זאת, הכמות העצומה של המידע שמתפרסם מדי יום ברשת יוצרת קושי באיתור יעיל ומהיר של קורבנות אלימות חדשים. מעבר ידני על פורומים ייעודים העוסקים באלימות במשפחה עשוי להיות קשה ליישום מאחר וגם בתוך הפורומים האלה יש פרסומים רבים שאינם רלוונטיים לארגוני הסיוע כגון סיפורי גבורה של קורבנות או פרסומים להעלאת מודעות. לכן, לדעתנו, יש צורך בפיתוחה של מערכת אוטומטית שתדע להבחין בין פרסומים של קורבנות אלימות השרויים במצב קריטי וזקוקים לעזרה לבין פרסומים שאינם מתארים סיטואציה קריטית. מערכת שכזו יכולה לסייע באיתור מהיר של אירועים קריטיים ולסייע למנהלי פורומים ולאנשי מקצוע לתעדף אותם ולהעבירם לארגוני סיוע מתאימים.

בעבודה זו ננסה לבנות מערכת אוטומטית לזיהוי פרסומים קריטיים בנושא אלימות במשפחה באמצעות טכניקות מעולם למידת המכונה. נבחן דרכים שונות לעיבוד מקדים של הפרסומים, דרכים לייצוג וקטורי של הפרסומים ומס' אלגוריתמי למידה שונים במטרה ליצור מסווג בינארי אשר ידע להבחין בין פרסום המתאר סיטואציה קריטית של אלימות במשפחה לבין פרסום המתאר סיטואציה שאינה קריטית.

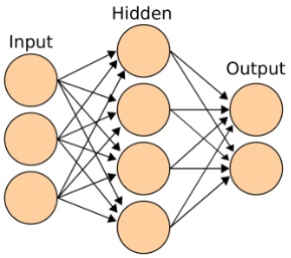
מושגי יסוד :

בפרק זה נציג מספר מושגי יסוד בהם נשתמש בפרקים הבאים לצורך הבנת האלגוריתמים השונים בהם השתמשנו במהלך הכנת הפרוייקט.

1) רשת עצבית מלאכותית-

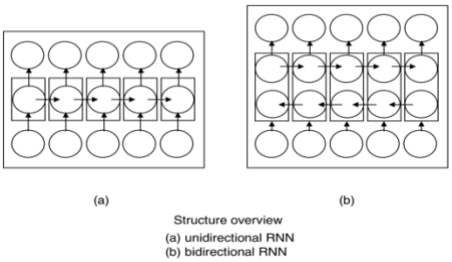
רשת עצבית מלאכותית היא מודל חישובי שהושפע מהמבנה של רשתות עצביות במוח שלנו.
היחידה הבסיסית המרכיבה רשת עצבית מלאכותית היא נוירון- יחידת חישוב בסיסית , בעלת מספר כניסות ותוצאה אחת, שערכה הוא פונקציה כלשהי של הכניסות. הפונקציה המופעלת בנוירון נקראת פונקציית הפעלה, activation function.
הרשת עצמה מורכבת ממספר רב של נוירונים ומקובל לתאר אותה כגרף מכוון ממושקל, שצמתיו הם הנוירונים השונים והקשתות שלו הם הקשרים בין הנוירונים. הנוירון למעשה מקבל כקלט את הסכום הממושקל (משקל לפי הקשתות) של כל הפלטים של השכנים הנכנסים אליו. המשקל בכל קשר קובע עד כמה רלוונטי המידע שעובר דרכו, והאם על הרשת להשתמש בו על מנת לפתור את הבעיה.
הרשת היא היררכית ומסודרת בשכבות. השכבה הראשונה נועדה לקלוט מידע לרשת, השכבה האמצעית ידועה כשכבה החבויה (במודלים שונים עשויים להיות יותר מאחת כזו), ולבסוף השכבה האחרונה אשר נועדה להחזיר את המידע המעובד כפלט.
המשקולות שעל הקשרים בין הנוירונים הן אלה שאוגרות את הידע של הרשת במודל המפושט, לכן אלגוריתמי הלימוד השונים מבצעים את כוונון המשקולות לערכים שנותנים תוצאות חישוב טובות עבור מערך דוגמאות הלימוד. משימה זאת אינה פשוטה בגלל השכבות החבויות.
תהליך הלמידה מתבצע על ידי "תגמול" "ועניש" של קשרים שונים ועל ידי חשיפת רשת הנוירונים לדוגמאות רבות. "תגמול" ו"עניש" של הקשרים מתבצע על ידי שינוי המשקל של אותו הקשר, כך שכל קשר ש"מתוגמל" משקלו יגדל וכל קשר ש"נענש" משקלו ירד.
בהינתן רשת בה כוונו המשקולות, חישוב על ידה מתבצע על ידי הזנת הקלטים לשכבת הקלט וקבלת התוצאה שחלחלה ברשת לשכבת הפלט [6] [5].

ארכיטקטורות שונות של רשתות למידה בהן נעשה שימוש במסגרת הפרויקט :

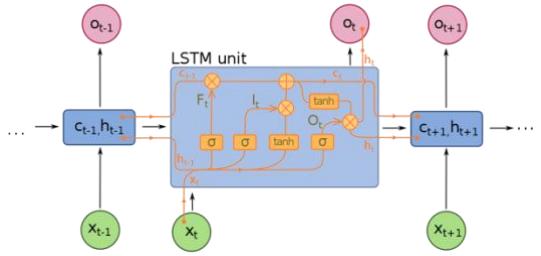


- **רשת זרימה קדימה ("feedforward network"):**
  רשת זו מתוארת כגרף מכוון ללא מעגלים.
  ברשת כזו, המידע זורם רק בכיוון אחד, קדימה, מצמתי הקלט, דרך הצמתים החבויים (באם קיימים) אל צמתי הפלט, ללא מעגלים או לולאות עצמיות [5].

- **רשת RNN(Recurrent Neural Network)**- רשתות אלו נועדו "לפענח" מידע סדרתי - כלומר, להביא בחשבון את "סדר ההופעה" של התופעה בה מטפלים. רשתות אלו כוללות מרכיב של זיכרון - כך שהרשת "יודעת להתחשב" בהיסטוריה. הרשת מבצעת לכל אלמנט בסדרה את אותו תהליך ולכן היא "רשת עצבית חוזרת", ובאמצעות הזיכרון שלה היא יודעת להתחשב בפלט של חישובים מאלמנטים קודמים. ברשתות כאלו, בניגוד לרשתות זרימה קדימה, יכולות להיות קשתות עצמיות ומעגלים בגרף הרשת.

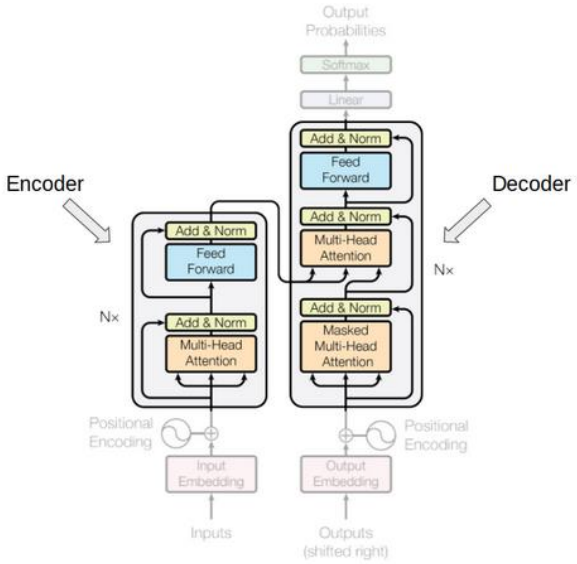


רשת RNN דו כיוונית- במקרים מסוימים, כמו למידה של שפה טבעית (NLP),
נרצה שמודל ה-RNN יקרא את הנתונים משני הכיוונים - קדימה ואחורה - למידה דו-כיוונית – Bidirectional. הארכיטקטורה של רשת זו משלבת בין רשת RNN שמעבדת את הרצף קדימה בזמן מתחילת הרצף ורשת RNN שמעבדת את הרצף אחורה בזמן, כלומר מתחילה לקרוא אותו מסוף הרצף [8] [7].



- **רשת LTSM(Long short-term memory)**
סוג של רשת עצבית חוזרת (RNNs) שיש לה את היכולת ללמוד ולזכור לאורך רצפים ארוכים של נתוני קלט באמצעות "שערים" המסדירים את זרימת המידע של הרשת. באופן זה, LTSM יכול באופן סלקטיבי לזכור או לשכוח מידע [9] .

- **רשת Transformer-**
הרשת עצמה הוצגה לראשונה במאמר [10], והיא עוצבה בארכיטקטורת "מקודד-מפענח".
בדומה לרשתות RNN, רשתות מסוג transformes עוצבו כדי להתמודד עם מידע סדרתי, בין היתר עם משימות כמו תרגום משפטים משפה לשפה, סיכום טקסט ועוד.
בניגוד לRNN, רשת עצבית מסוג transformer אינה משתמשת ברכיבי זיכרון, ואינה מעבדת את המידע באופן סדרתי אלא במקביל, באמצעות מנגנון שנקרא מנגנון "תשומת הלב" (attention).
מנגנון "תשומת הלב"("attention") מייצר וקטור חדש עבור כל אלמנט בסדרת הקלט, כאשר הוקטור אומר עד כמה האלמנט הנוכחי במוצא קשור לכל אחד מהאלמנטים בסדרה המקורית. באופן הזה כל איבר במוצא נותן תשומת לב שונה לכל אחד מאיברי הכניסה, ולכן המנגנון נקרא attention. מאחר וניתן להפעיל את המנגנון במקביל על כל האלמנטים, זמן האימון של הרשת מהיר יותר מזה של הRNN.
כתחליף לרכיב הזיכרון, הtransformer משתמש בגישת "positional encoding" כדי לייצג את הסדר בין איברי הסדרה. במסגרת שיטה זו, מוסיפים לכל אחד מאיברי הקלט פיסת מידע לגבי המיקום שלה בסדרה[12] [11] .



2) תהליך זיקוק ידע (knowledge distillation):
תהליך זיקוק ידע הוא תהליך המשמש לדחיסת מודל גדול לצורך הרצתו על מכשירים סלולריים או התקני קצה קטנים. בתהליך זה, אנו מאמנים רשת גדולה ומורכבת (רשת זו מכונה "המורה") אשר יכולה לחלץ תכונות חשובות מהנתונים שניתנו ולכן יכולה לייצר תחזיות טובות יותר. לאחר מכן אנו מאמנים רשת קטנה(רשת זו מכונה "התלמיד") בעזרת המודל הגדול והמורכב. המודל המוקטן, הסטודנט, מאומן במטרה לחקות את הפלט של המורה במקום להיות מאומן על המידע הגולמי.
רשת קטנה זו תוכל לייצר תוצאות דומות, ובמקרים מסוימים, היא אף יכולה להיות מסוגלת לשכפל את תוצאות הרשת המורכבת [13] .

3) שיכונים (embeddings):
בעולם למידת המכונה, שיכון הוא פונקציית מיפוי של וקטורים עם מס' גבוה של ממדים לוקטורים עם מס' נמוך של ממדים. השימוש הנפוץ של שיכונים הוא הפיכת וקטורים דיסקרטיים ודלילים לוקטורים דחוסים. במצב אידאלי, השיכון יצליח "ללכוד" חלק מהסמנטיקה של הקלט ע"י כך שהוא ימקם קלטים דומים מבחינה סמנטית קרוב אחד לשני בתוך המרחב . בעולם עיבוד השפה הטבעית, נהוג להשתמש במושג שיכון(embedding) כדי להתייחס לוקטור הייצוג הסופי של האובייקט (כלומר לתוצאה של הפעלת פונ' המיפוי על אובייקט מסויים כמו מילה או משפט) [14].

- **שיכוני מילים (word embedding)-** בעולם עיבוד השפה הטבעית, נהוג להשתמש בשיכונים לצורך ייצוג מילים. כדי ליצור את השיכונים, נהוג להשתמש ברשתות עצביות מלאכותיות אשר מצליחות ללמוד ייצוגים וקטוריים מבוזרים למילים. בדרך כלל, רשתות אלו ייצרו עבור המילה וקטור דחוס וקצר, עם מס' מימדים d שנע בדר"כ בין 50-1000. הוקטור ייצג את המשמעות הסמנטית של המילה, כך שלמילים בעלות וקטורים קרובים במרחב הוקטורי צפוי להיות משמעות דומה.

חשוב לציין כי יש המשתמשים במונח word embeddings כדי לציין כל ייצוג וקטורי של מילה (גם ייצוג דליל כמו bag of words) אך אנחנו נשתמש במונח זה בפרוייקט בשימוש הנפוץ שלו המתאר וקטורים מבוזרים ודחוסים[15] .

- **שיכוני משפטים (sentence embeddings)**- בדומה לשיכוני מילים, שיכוני משפטים הם וקטורים נומריים דחוסים ומבוזרים שמטרתם לייצג רצפי טקסט ארוכים יותר ממילה כמו משפטים ופסקאות קצרות. יצירת שיכוני משפטים נעשית בדר"כ ע"י מקודדים הבנויים מרשתות עצביות מלאכותיות. ניתן להסתכל על שיכוני משפטים כמעין הרחבה של שיכוני מילים, כך שגם במקרה זה המטרה ביצירת הוקטורים היא ללכוד בתוכם "מידע סמנטי" כך שלמשפטים דומים יהיה ייצוג וקטורי דומה[16] .

4) **<u>העברת לימוד ("transfer learning"):</u>**
טכניקה בעולם "למידת המכונה" במסגרתה משפרים את הלמידה של מודל על משימה חדשה(משימת היעד) ע"י העברת ידע ממשימה אחרת אבל דומה (משימת המקור), שכבר נלמדה[17] .
בעולם הNLP החל שימוש נרחב ב"העברת לימוד" ע"י מודלים מאומנים מראש. מודלים אלו מאומנים על קורפוס גדול מאוד ויכולים לייצר ייצוגים אוניברסליים לטקסט, על אף שאומנו לצורך פתרון משימה ספציפית המודל המאומן מייצר ייצוג אוניברסלי לטקסט ולאחר מכן ניתן להשתמש בייצוג שנוצר לטקסט כדי לאמן מודל אחר לפתור משימה ספציפית. השימוש במודלים מאומנים מראש מסייע רבות במשימות NLP בהן המידע שיש לצורך אימון הוא קטן ואינו מספיק לצורך אימון של רשת עצבית מלאכותית. בנוסף, הוא מאפשר גם למפתחים עצמאיים לייצר מודלי NLP איכותיים על אף שאין רשותם משאבים חישוביים כדי לאמן רשתות למידה עמוקות על מאגרי מידע המכילים ביליוני פריטים.
בגרסה הראשונית, השתמשו במודלים מאומנים מראש המייצרים שיכוני מילים כמו word2vec כבסיס למודלים מורכבים יותר, שכן שיכוני המילים תופסים את הקרבה הסמנטית בין מילים שונות ועל כן מהוות ייצוג טוב יותר למילים מאשר שיטות אחרות. הבעיה בשיכוני מילים כמו word2vec היא שהייצוג עבור מילה מסוימת הוא חסר הקשר ולכן עדיין צריך לאמן את המודל כולו מאפס כדי להתאימו לבעיה ספציפית[18] .
בגרסה המאוחרת יותר, משתמשים במודלים מאומנים מראש הקרויים מקודדים והם מייצרים ייצוגים אוניברסאליים למשפטים ופסקאות כמו BERT. הפלט של מקודדים אלו נקרא גם שיכוני מילים מבוססי הקשר שכן הם מייצרים ייצוג עבור מילה במשפט שהוא מבוסס הקשר, כלומר הייצוג של אותה מילה ישתנה בין משפטים שונים[18] .

<u>5) מידול שפה(Language modeling):</u>
מידול שפה הוא משימה מרכזית בעולם הNLP אשר מטרתה ללמוד את פונקציית ההתפלגות מעל רצפים של מילים בשפה. בהינתן רצף מילים באורך n, הפונקציה מחזירה את ההתפלגות של כל רצף המילים- $P(w_1, w_2 \dots w_n)$.
מלבד חישוב ההסתברות לכל רצף אפשרי של מילים בשפה, מודל השפה מחשב את ההסתברות של כל מילה בשפה להופיע אחרי רצף נתון של מילים.
השימוש הכי מוכר ואינטואיטיבי של מודל שפה הוא השלמה אוטומטית, שמציעה את המילה או המילים הכי סבירות בהינתן מה שהמשתמש הקליד עד כה. אולם, מידול שפה משמש כבסיס למגוון משימות בעולם הNLP, בין היתר במשימות של תרגום טקסטים, סיכום טקסטים, מתן תשובות על שאלות וכד'[21] [20] [19] .

<u>6) BERT (Bidirectional Encoder Representations from Transformers):</u>
BERT הוא מודל לייצוג שפה שפותח ע"י חוקרים בגוגל ופורסם לראשונה במאמר[22] .
BERT עוצב במטרה לייצר מודל לייצוג שפה מאומן מראש אשר מפתחים אחרים יכולים להשתמש בו בקלות לפתרון משימות NLP ספציפיות ע"י כיוונון מס' פרמטרים במודל ושימוש במאגרי נתונים ייעודים למשימה. בדר"כ מאגרים אלו יהיו קטנים בהרבה מהמאגר עליו אומן BERT וכך השימוש במודל זה יאפשר תהליך יעיל של "העברת לימוד".
הארכיטקטורה של BERT היא "multi-layer bidirectional Transformer encoder", ארכיטקטורה זהה לארכיטקטורה המקורית של הTransformer. השימוש בארכיטקטורה זו, כפי שהסברנו קודם ,מאפשר ללמוד את ההקשר של מילה מבין כל המילים המקיפות אותה, בניגוד לשיטות אחרות שהיו נהוגות עד אותה תקופה ובהן היה נהוג ללמוד הקשר של מילה ע"י קריאת רצף המילים מימין לשמאל ומשמאל לימין.
חידוש נוסף במודל BERT הוא משימות הNLP עליהן אומנה רשת הלמידה. במקום לאמן את BERT לחזות את המילה הבאה במשפט (מידול השפה), בדומה לאיך שאומנו מודלים אחרים שפורסמו עד אותה תקופה, BERT אומן על שתי משימות של למידה לא מפוקחת :
*Masked LM (MLM) - טכניקה המבוססת על משימת מידול השפה שבה מחליפים אחוז מסוים מאסימוני הקלט באסימון mask, ולאחר מכן המודל מנסה לחזות את הערך המקורי של המילים שמוסכו , על פי ההקשר של המילים הלא ממוסכות ברצף.

* Next Sentence Prediction (NSP) – המודל מקבל זוג משפטים כקלט ומנסה לחזות האם המשפט השני הוא משפט המשך למשפט הראשון במסמך המקורי. במהלך האימון , 50% מהמשפטים היו זוג משפטים בהם המשפט השני היה המשך למשפט הראשון ו50% מזוגות המשפטים האחרים היו משפטים אקראיים שלא היה ביניהם קשר.

7) <u>NLI (Natural language inference):</u>

זוהי משימה מעולם הNLP אשר מטרתה לקבוע האם בהינתן הנחת יסוד, היפותזה היא נכונה (כלומר ניתנת לגרירה מהנחת היסוד), שגויה (כלומר סותרת את הנחת היסוד) או לא ניתנת לקביעה מהנחת היסוד (כלומר ניטרלית ביחס להנחת היסוד)[23] .
מאגר המידע הנפוץ ביותר לצורך פתרון משימה זו פורסם ע"י סטנפורד ונקרא- "The Stanford Natural Language Inference (SNLI) Corpus" . המאגר מכיל מכיל אוסף של כ-540 אלף זוגות משפטים שנכתבו בידי אדם, כאשר כל זוג משפטים מורכב מהנחת יסוד והיפותזה ומתוייג באחד מ3 הקשרים- גרירה, סתירה או ניטרלי[24] .

דוגמה למשפטים במאגר[23] :

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.

תיאור הפתרון המוצע לבעיה

בהמשך להסבר שניתן במבוא, נגדיר בצורה פורמלית את הבעיה אותה אנחנו מנסות לפתור במסגרת הפרויקט- בהינתן פרסום אשר פורסם באחת מהרשתות החברתיות, יש לסווג את הפרסום לאחת מ2 הקטגוריות הבאות: פרסום קריטי ופרסום שאינו קריטי. נדגיש כי הפרוייקט עוסק אך ורק בפרסומים הכתובים בשפה האנגלית.

ניתן לראות שהבעיה הנתונה היא בעיית סיווג בינארית ולכן בחרנו לנסות לפתור אותה כבעיית למידה מפוקחת.

לפני שנמשיך לתאר את הפתרון שלנו לבעיה הנתונה, נגדיר באופן מפורש כל אחת מקטגוריות הסיווג:

פרסום קריטי- פרסום אשר עוסק באלימות במשפחה, וכותב הפרסום מתאר סיטואציה מסוכנת אשר הוא או אדם קרוב אליו נמצאים בה, ונדרשת התערבות מיידית מצד גורם סיוע חיצוני.
הסיטואציה המסוכנת יכולה להיות פיזית- חשש שבן משפחה/בן בזוג יפגעו פיזית באותו אדם אשר נכתב עליו הפרסום, אך היא יכולה להיות גם נפשית- סיטואציה שבה האדם הנפגע שרוי במצב נפשי קשה בדיכאון וזקוק לסיוע נפשי.

דוגמה לפוסטים קריטיים:

"I'm in a bad bad situation I need help! I'm scared to stay at the house! The shit he dose to me is crazy and I can't deal with it or the head games no more! I have an appointment with DSS at 11 tomorrow idk what there gonna do but I got police reports for them idk if that will help but I definitely need a restraining order! I don't wanna go into what he's doing to me because I'll just get upset and it took a lot just for me to write this! I need help I can't loose my job because I have no where to stay and I'm scared of a shelter it's to cold to sleep out side and I can't deal with being at the house because of the things he keeps doing! There's gotta be some place that will help me and not just stick me in a shelter!!

" I try not to post things about my husband my abuser but a person can only stay strong for so long. I've been dealing with it for 5 years I'm stronger than this I don't even know why I've put up with it for so long. Battered women's syndrome is real and it sucks. Mentally and physically I'm at my lowest. How do you overcome and become yourself again?

פרסום שאינו קריטי- פרסומים אלה מתחלקים בעיקרם ל2 קטגוריות:

- פרסומים העוסקים באלימות במשפחה אך מתארים סיטואציה שאינה קריטית. פרסומים כאלה יכולים להיות פרסומים שמטרתם העלאת מודעות לתופעת אלימות במשפחה או פרסומים המתארים סיפור הצלחה אישי של אדם שחווה אלימות במשפחה אך הצליח לצאת מסיטואציה הזו והיום שרוי במצב טוב ואינו תחת סכנה פיזית או נפשית.
- פרסומים שאינם עוסקים באלימות במשפחה. אלו יכולים להיות גם פרסומים שליליים, שבהם אנשים מתארים בעיות במערכות יחסים, פרידות, מצבי דיכאון ורצון לשים קץ לחייהם, אך אין אזכור בפרסום לאלימות במשפחה. אומנם פרסומים אלו יכולים לתאר סיטואציות קריטיות, אולם מטרתנו בפרויקט זה היא לאתר נפגעי אלימות במשפחה שזקוקים לעזרה ולכן מבחינתו פרסומים אלה אינם רלוונטיים כפרסומים קריטיים.

דוגמאות לפוסטים שאינם קריטיים:

"After a year and 10 months, I left my abuser last February. I was so scared and had nowhere to go. 6 months later I have a an apartment with my now boyfriend and best friend. Living my best life, if you ever thought it wasn't possible it is!

"Everyday may depression and anxiety get worse where to the point I cant eat, sleep pr even do what i need to do, and my heart feels heavy very heavy like theirs a big burden inside my heart that makes me not to move or to lose interest in everything.
Then my mom notice my behaviour so she ask whats your problem, and she also say your too young to get problem or to feel depress, your not even starting to work. Then I feel like i gonna cry because i want her to hug me, to care for me, or to talk to her, but I can because of what she say, I can even say that I feel like dying everyday.

"Hi Guys,
I'm a journalist researching an article about domestic violence from the point of view of a male victim. This is a subject I feel passionate about and I'd like to raise awareness for a side that doesn't get talked about as much. I'm posting as I'd like to find a few case studies of male victims, so if this is you and you're happy to talk (anonymously is fine) please get back to me."

איסוף נתונים ויצירת מאגר הנתונים

מאחר ובחרנו לפתור את הבעיה הנתונה כבעיה של למידה מפוקחת, השלב הראשוני בפתרון הבעיה הוא למצוא או לייצר מאגר נתונים המכיל דוגמאות מתויגות של פרסומים קריטיים ולא קריטיים.
ככל הידוע לנו, מצאנו עבודה קודמת אחת בלבד שנעשתה בנושא [25], אך מאגר הנתונים בו השתמשו לצורך עבודה זו לא פורסם. ניסיונות ליצירת קשר עם כותבי המאמר לא צלחו ולכן החלטנו לייצר מאגר נתונים משלנו.

כיוון שמאגר הנתונים עליו בוחרים לאמן את המודל הוא קריטי לצורך תהליך הלמידה, רצינו לבחור דוגמאות איכותיות למאגר הנתונים. המטרה הייתה הן לבחור דוגמאות יחסית קצרות שהן חד משמעיות מבחינת הסיווג שלהן כלומר שברור האם הן מתארות סיטואציות קריטיות/לא קריטיות. בנוסף, היה חשוב לנו לבחור דוגמאות שהן יחסית קשות- רצינו לאמן את המסווג לדעת להבחין בין פרסומים קריטיים/לא קריטיים אשר מכילים עולם תוכן ואוצר מילים דומה.

אספנו דוגמאות משתי רשתות חברתיות עיקריות:
* Facebook- אספנו פרסומים מקבוצות שונות העוסקות באלימות במשפחה ומערכות יחסים אלימות בצורה ידנית.
* Reddit- מכיל פורומים רבים המאורגנים לפי תחומי עניין שונים המכונים subreddits".
אספנו פרסומים מ"subreddits" העוסקים באלימות במשפחה- "DomesticViolence ",
"Abusive Relationships " . בנוסף לצורך מציאת פרסומים לא קריטיים אספנו פרסומים גם מsubreddits
כמו: "Relationship Advice", "Depression", "Love", "Heartbreak" וכד'.

המשימה של איתור פרסומים המתארים סיטואציה קריטית התגלתה כמשימה קשה מאחר ולא היה ניתן לשייך פרסומים רבים לאחת מ2 הקטגוריות- פרסומים רבים תיארו סיפור אישי של אלימות במשפחה אך לא ניתן היה להסיק מסיפור זה אם אותו אדם זקוק לעזרה כעת או לא. בנוסף פרסומים רבים היו ארוכים מדי ומסורבלים.
בסופו של דבר הצלחנו לאתר כ200 דוגמאות שסווגו כסיטואציה קריטית.
עבור הפרסומים שמתארים סיטואציות שאינן קריטיות- הצלחנו למצוא יותר דוגמאות אך מאחר ורצינו שמאגר הנתונים יהיה מאוזן, החלטנו לבחור את הדוגמאות שנראו לו המלמדות ביותר. שאינן קריטיות מתחלקות בין פרסומים העוסקים באלימות במשפחה ובין פרסומים שאינם עוסקים באלימות במשפחה אך מתארים מצוקות שונות של אנשים ובעיות במערכות יחסים. מצאנו כי פרסומים אלה בעלי עולם תוכן דומה לעולם האלימות במשפחה ורצינו לאמן את המודל שלנו לסווג בין דוגמאות קשות, שכן המשימה של לסווג פרסומים מעולמות תוכן שונים לגמרי היא קלה יותר.
בסופו של דבר איתרנו כ-200 דוגמאות שסווגו כסיטואציה שאינה קריטית.

חשוב לציין כי כל פרסום תויג על ידינו במהלך האיסוף הידני של הפרסומים. פרסומים שהיה עליהם אי הסכמה בינינו, לא הוכנסו למאגר הנתונים. בנוסף , השתדלנו לבחור פרסומים יחסית קצרים, כאשר רובם מכילים עד כ500 מילים ורק מספר מועט של פרסומים מכילים יותר מילים.

עיבוד מקדים של הטקסט

בשלב זה ביצענו עיבוד מקדים מינימלי של כל הפרסומים במאגר הנתונים שיצרנו.
עיבוד מקדים זה כלל-

1) **Tokenization-** כפי שנראה בהמשך, הקלט עבור רוב השיטות לייצוג טקסט הוא לא הטקסט בשלמותו, אלא יש לפצל את הטקסט למילים, אסימונים. משימה זו של שבירת הטקסט לאסימונים נקראת Tokenization. אומנם המשימה נראית פשוטה, שכן רוב המילים באנגלית מופרדות ע"י רווח , אך מצב זה לא תמיד מתקיים [15]. למשל, עבור רוב משימות הNLP, נצטרך להפריד את סימני הפיסוק כאסימונים נפרדים, אבל לפעמים נרצה לשמור את סימני הפיסוק בתוך המילה, למשל כמו במילה – Ph.D.
כמו כן, סימני פיסו' יכולים להופיע גם בתוך מספרים- למשל 55,000 וגם אותם לא נרצה להפריד [15].

חשוב לציין שתהליך שבירת האסימונים הוא תהליך שהטקסט כמעט תמיד עובר מאחר ורוב האלגוריתמים המהנדסים תכונות מהטקסט מקבלים כקלט את האסימונים של הטקסט או מקבלים את הטקסט בשלמותו אך מפרקים אותו לאסימונים בתוך האלגוריתם. חשוב לציין כי גם אופן שבירת האסימונים עשוי להשפיע על התוצאות אבל החלטנו לא לבחון אלגוריתמים שונים ליצירת האסימונים אלא השתמשנו במודל מוכר אחד הנקרא "Penn Treebank tokenization". אלגוריתם זה מומש ע"י

ספריית NLTK. האלגוריתם מבצע שבירה של הטקסט לאסימונים באמצעות ביטוים רגולריים. בין היתר הוא מפריד קיצורים ל2 מילים נפרדות למשל-
they'll יומר ל they ו'll, מפריד את כל הפיסוק לאסימונים נפרדים, שומר על מילים המחוברות ע"י מקף ביחד ועוד [15].

2) case-folding- לאחר שבירת הטקסט לאסימונים, המרנו את כל האותיות הגדולות באסימונים לאותיות קטנות . המטרה בסוג זה של עיבוד היא יצירת צורה נורמלית למילים, כך שהמחשב לא יבחין בין מילים עם משמעות זהה כמו the,The כמילים שונות.

בנוסף, במסגרת הניסויים שלנו, עליהם נפרט בהמשך, בחנו שיטות עיבוד מקדים נוספות מעל העיבוד המינימלי שביצענו:

1)הסרת "stop words"- שיטה נוספת לעיבוד מקדים של טקסט היא הסרה של מילים נפוצות במיוחד בשפה, שנושאות בקרבן מעט מאוד אינפורמציה. דוגמאות למילים כאלה הן למשל a, the, is, are וכו'. מילים כאלו נקראות "stop words". כיום אין רשימה אוניברסלית יחידה של "stop words" שמקובלת על כל עולם הNLP, אלא יש מגוון רשימות , מרשימות ארוכות שמכילות כ-300 מילים לרשימות קצרות שמכילות כ-7 עד 12 מילים[27] [26] .
חשוב לציין שלא תמיד כדאי להסיר stop words מאחר ומילים אלה יכולות להוסיף הקשר למשפט, למשל במשימת "ניתוח הרגש"- ביקורת על סרט יכולה להיות : " The movie was not good at all",אך לאחר הסרת stop words נקבל - "movie good". ניתן לראות שלאחר הסרת הstop words, ההקשר המקורי של המשפט נעלם לגמרי.
מאחר והשיטה אינה תמיד מניבה שיפור בביצועים [28], החלטנו שכדאי לבחון שיטה זו לעיבוד מקדים במסגרת הניסויים שלנו ולא להשתמש בשיטה זו באופן עיוור. לצורך הניסויים השתמשנו ברשימת הstop-words המוצעת ע"י ספריית NLTK.

2) Lemmatization- שיטה לעיבוד מקדים של טקסט אשר מייצרת צורה סטנדרטית למילים/אסימונים. במסגרת שיטה זו קובעים עבור כל אסימון את צורת השורש שלו. לדוגמה עבור המילים sang,sung,sings תיבחר הצורה הסטנדרטית של הפועל sing. עבור המלים am,is,are תבחר הצורה הסטנדרטית be. בנוסף גם מילים אשר בצורת רבים יומרו לצורת יחיד- dinners יומר לdinner [15].

קיימים מגוון אלגוריתמים לצורך ביצוע Lemmatization.
אנחנו בחרנו להשתמש ב"WordNetLemmatizer " במסגרת הפרוייקט. wordNett הוא מאגר מידע לקסיקלי גדול, זמין באופן חופשי וציבורי לשפה האנגלית, שמטרתו לבסס מערכות סמנטיות מובנות בין מילים. Wordnet מציע פונקציה מורפולוגית אשר עליה מתבסס האלגוריתם בו השתמשנו [29].
האלגוריתם עצמו ממוש ע"י ספריית NLTK.

הנדסת תכונות מהטקסט (Feature engineering)

לאחר ביצוע העיבוד המקדים, היה עלינו להנדס מאפיינים ותכונות מהטקסט. אלגוריתמי הלמידה המפוקחת השונים מצפים לקבל כקלט אוסף תכונות, כאשר עבור כל דוגמה מייצרים וקטור תכונות נומרי המייצג אותה.
מאחר ובחרנו בבעיית למידה הקשורה לעולם עיבוד השפה הטבעית, עמד בפנינו אתגר גדול יותר- איך לייצר מהטקסט וקטור תכונות נומרי שיכיל מידע על מבנה הטקסט וישמר את המשמעות הסמנטית שלו.
לאחר מחקר שביצענו על יצירת תכונות מטקסט, החלטנו שיהיה מעניין לבחון בשלב הניסויים סוגים שונים של תכונות שניתן ליצור מהטקסט ושילוב של תכונות.
על הניסויים עצמם נפרט בהמשך, אך בשלב זה נציג התכונות שבחרנו לבחון בניסויים.
חשוב לציין כי במהלך הפרקים הבאים נכנה את הדוגמאות שלנו במאגר הנתונים בחלק מהמקרים בשם "מסמכים"- הכוונה במילה "מסמך" – אוסף של מילים (יותר ממילה אחת) שיכול להיות משפט, מספר משפטים או פסקה קצרה.

ניתן לחלק את השיטות שבחנו ליצירת תכונות מהטקסט ל3 משפחות עיקריות(בעמוד הבא):

תכונות מבוססות מילים(נכנה אותן תכונות לקסיליות)

מאפיין עיקרי של משפחה זו שהתכונות של המסמכים יהיו המילים עצמן. בשל כך אנחנו נכנה תכונות אלו בפרויקט בשם תכונות לקסיקליות- התכונות מורכבות מאוצר המילים, מהלקסיקון.
השיטות השונות שמשתמשות במילים כתכונות באופן ישיר מייצרות וקטורים שהמאפיין שלהם הוא שכל מילה מיוצגת ע"י מימד בוקטור.
הוקטורים הנוצרים מהמסמך יהיו לרוב דלילים (וקטורים שמרבית איבריהם בעלי ערך 0), כאשר לכל מימד יש משמעות.

בעבודה בחרנו לבחון מספר מודלים שונים המייצרים תכונות לקסיקליות:

- **"Bag of words"-** כפי שהשם של שיטה זו מציע, המודל מייצר מהמסמך "שק" של מילים, כאשר עבור כל מילה מופיעה התדירות שלה במסמך.
באופן פורמלי, מודל זה מייצר עבור כל מסמך טקסט במאגר המסמכים וקטור דליל של ספירת מופעים של מילים, היסטוגרמה על אוצר המילים הנתון.
המודל יוצר מכלל המסמכים במאגר מילון שכולל את כל המילים שמופיעות במסמכים. לאחר מכן, עבור כל מסמך במאגר יוצרים וקטור בו כל מימד מייצג מילה מהמילון (האינדקס של המימד מייצג מילה מסוימת) והערך של המימד מייצג את מספר הפעמים שאותה מילה הופיעה במסמך.
הוקטור עבור כל מסמך הוא בגודל קבוע מאחר וכל וקטור מכיל את כל המילים במילון.
יש לציין כי שיטת ייצוג זו מתעלמת לגמרי מסימני פיסוק ולא מחשיבה אותם כחלק מהמילים במילון [30]. אנחנו השתמשנו בפרוייקט במודל שמומש ע"י ספריית sklearn ונקרא CountVectorizer .

- **"TF-IDF"**(term frequency-inverse document frequency)-
שיטה ליצירת תכונות מהטקסט המתבססת על המדד הסטטיסטי TF-IDF שמטרתו להעריך כמה מילה רלוונטית למסמך באוסף של מסמכים. שיטה זו היא מעיין מודל משופר של Bag of words, שמטרתו לתת משקלים למילים ולאזן את ההשפעה של מילים שמופיעות במסמכים רבים באוסף ולכן הן נושאות בהן פחות מידע באופן אמפירי לעומת מילים שמופיעות בחלק קטן יותר של המסמכים.
הרעיון בבסיס השיטה- החשיבות של מילה לא תלויה רק בתדירות שלה אלא גם בכמה מסמכים מופיעה.
שיטה זו מייצרת עבור כל מסמך וקטור בו כל מימד מייצג מילה מהמילון של כלל המסמכים, בדומה לBag of words, אך בניגוד ל Bag of words הערך של מימד זה יהיה ציון הTF-IDF שניתן למילה זו במסמך.

איך מחושב ציון הTF-IDF עבור מילה במסמך ?

$$TFIDF(w) = TF(w,d) * IDF(w)$$

כאשר:

- TF(w,d) = התדירות של המילה w במסמך d כלומר מס' הפעמים שמופיעה המילה w במסמך d.
$IDF(w) = \log\left(\frac{N}{df(w)}\right)$ כאשר N הוא מספר המסמכים במאגר וdf(w) הוא מספר המסמכים שהמילה w מופיעה בהם. ככל שהמילה w תופיע ביותר מסמכים, המנה תקטן ולכן גם ציון הIDF יקטן. ציון הIDF הוא למעשה המשקל שניתן למילה- מילים שמופיעות בפחות מסמכים יקבלו משקל גבוה יותר מאחר והן יכולות ליצור הפרדה בין אוסף המסמכים המכיל אותן לאוסף המסמכים שלא מכיל אותן.
מילים שחוזרות ברוב המסמכים יקבלו משקל נמוך יותר[15] [30] . אנחנו השתמשנו במסגרת הפרוייקט במודל שמומש ע"י ספריית sklearn ונקרא TfidfVectorizer .

תכונות מבוססות שיכונים (נכנה אותן תכונות סמנטיות)

המאפיין העיקרי של משפחה זו היא שהתכונות של המסמך ישמרו את המשמעות הסמנטית שלו. מודלים ממשפחה זו לא יתבססו על המילים באופן ישיר כאשר הם ייצרו וקטור תכונות, אלא ייצרו שיכונים (embeddings) עבור המסמכים, כלומר ימפו כל אחד מהמסמכים לאותו מרחב וקטורי וינסו לשמר את המשמעות הסמנטית של המסמכים ע"י יצירת וקטורים דומים למסמכים דומים. מאחר והתכונות המיוצרות ע"י מודלים ממשפחה זו מסייעות בשימור הסנמטיקה – נכנה אותן תכונות סמנטיות.
מודלים אלו לרוב מייצרים עבור כל מסמך וקטור דחוס בעל מס' מועט יחסית של מימדים. בשונה מהתכונות הלקסילקיות, עבור התכונות הסמנטיות אין משמעות לכל תכונה בנפרד (אין משמעות למימד מסוים) אלא

התכונות קשורות אחת לשניה – הייצוג של המסמך הוא ייצוג מבוזר כלומר המידע על המילים והקשרים הסמנטיים בין המילים מפוזר לאורך כל הוקטור ומוטמע בתוכו[15] .

<u>בעבודה בחרנו לבחון מספר מודלים שונים המייצרים תכונות סמנטיות:</u>

- **fastText**- מודל זה פותח ע"י חוקרים מFacebook ופורסם לראשונה במאמר [31].
מודל זה מייצר וקטור בעל 300 מימדים לכל מילה תוך התחשבות במורפולוגיה, במבנה המילים הקיימות בשפה ורכיביהן בעלי המשמעות. המודל מתחשב ביחידות משנה של המילה ומייצר ייצוג עבור כל מילה ע"י סכימה של n-gram character (רצף של n תווים מהמילה). מודל זה נגזר מהמודל skip-gram אשר בהינתן מילה מרכזית מנסה לחזות מה המילים המקיפות אותה.
מודל זה מייצר ייצוג למילה ולא למסמך ולכן חישבנו עבור כל מסמך את <u>הוקטור הממוצע-</u> ביצענו ממוצע על וקטורי המילים שקיבלנו מהפעלת fastText על כל מילה במסמך. בחרנו להשתמש דווקא במודל זה מאחר וfastText יודע לייצר שיכוני מילים גם עבור מילים שלא מכיר , בניגוד לword2vec למשל. כיוון שראינו כי בהרבה מהפרסומים יש שגיאות כתיב ושימוש בסלנג הנחנו כי עדיף להשתמש במודל זה.
אנחנו השתמשנו ב<u>מודל</u> מאומן מראש שהוצע ע"י ספריית fastText. המודל אומן על כ-300 ביליון מילים באנגלית שנלקחו ממאגרי המילים של Wikipedia וcommon crawl.

- **doc2vec**- מודל זה הוצג לראשונה במאמר [32].
doc2vec הינו מודל המייצר ייצוג וקטורי למסמכים והוא הכללה של המודל word2vec.
word2vec הוא שם כולל לזוג מודלים המייצרים שיכוני מילים. word2vec יכול להשתמש באחת מבין 2 ארכיטקטורות כדי לייצר ייצוג מבוזר עבור המילים – האחת היא ארכיטקטורת skip-gran (כמו בfastText) , והשנייה היא – CBOW(Continuous Bag Of Words) , ארכיטקטורה בה המודל מנסה לחזות את המילה הנוכחית מתוך חלון של מילות הקשר הנמצאות סביבה. יוצרי המודל doc2vec עשו שימוש בארכיטקטורת CBOW ועשו בה שינוי פשוט- במקום להשתמש רק במילות ההקשר כדי לחזות את המילה הנוכחית, הם הוסיפו וקטור שנקרא "וקטור פסקה" אשר היה ייחודי עבור כל מסמך והשתמשו גם בו כדי לבצע את משימת החיזוי. כך , בזמן שהם אימנו את המודל לייצר שיכוני מילים , הם לימדו אותו גם לייצר את "וקטור הפסקה" עבור כל מסמך, ובסוף האימון וקטור הפסקה טמן בחובו ייצוג נומרי למסמכים. חשוב לציין כי doc2vec מייצר וקטור ייצוג עבור מסמך ללא קשר לאורך המסמך.המודל בו בחרנו להשתמש מומש ע"י ספריית genism . לא מצאנו את מודל מאומן מראש של doc2vec ולכן אימנו את המודל בעצמנו על הקורפוס הנוצר מהמסמכים שלנו בלבד.

- **inferSent** – מודל זה פותח ע"י חוקרים מחברת Facebook ופורסם לראשונה במאמר [33].
במאמר הציעו החוקרים מודל מבוסס רשת עצבית מלאכותית שאומן מראש ומטרתו לייצר ייצוג וקטורי אוניברסלי עבור משפטים (כאשר הכוונה במשפט הוא כל קטע טקסט קצר, יכול להיות גם פסקה קצרה). החידוש במודל הוא השימוש בלמידה מפוקחת לצורך אימון המודל- המודל אומן לפתור את משימת הNLI באמצעות מאגר הSLNI. החוקרים שיערו כי אם יאמנו מודל של למידה עמוקה לבצע את משימת הNLI , הם יוכלו ללמוד דרך לקודד משפטים לייצוג נומרי בעל משמעות אוניברסלית.
המודל עצמו אומן בצורה הבא- ראשית התבצע קידוד ע"י מקודד הבנוי מרשת עצבית מלאכותית בעל ארכיטקטורת רשת LTSM דו כיוונית . בכל שלב באימון, המקודד קודד את ההנחה וההיפותזה ופלט עבור כל אחד מהם וקטור המייצג אותם. המקודד עצמו מקבל כקלט את שיכוני המילים של כל משפט (במאמר השתמשו בשיכוני מילים מסוג GloVe).
לאחר הקידוד בוצעו פעולות מתמטיות שונות על שני הוקטורים מהשלב הקודם שיצרו וקטור אחד מאוחד. הוקטור המאוחד נשלח כקלט למסווג אשר סיווג וקטור זה לאחת מ3 הקטגוריות של משימת הNLI. בכל שלב , הרשת תיקנה את עצמה בהתאם לפתרון משימת הNLI וכך היא למדה איך לייצר קידודים טובים יותר. המודל אשר השתמשנו בו לצורך הניסויים היה <u>מודל מאומן מראש</u> של infersent (<u>פורסם ע"י פייסבוק</u>) עם שיכוני מילים מסוג GloVe (המודל נקרא infersent1). זהו המודל המקורי מהמאמר.

- **Universal Sentence Encoder**- מודל זה פותח ע"י חוקרים מחברת Google ופורסם לראשונה במאמר [34].
מודל זהו הוא למעשה מקודד אשר אומן מראש ומטרתו לייצר שיכונים (embeddings) אוניברסליים עבור טקסים כמו משפטים, צירופים או פסקאות קצרות. הקלט למודל הוא טקסט באנגלית והפלט מהמודל הוא וקטור בן 512 מימדים. החידוש המרכזי של המודל הוא אימון המודל לפתרון מגוון של

בעיות NLP (בשונה מפייסבוק שאימנו את המודל inferSent לפתור רק את משימת הNLI) ועל בסיס הטעויות שהמודל עושה בפתרון הבעיות לעדכן את השיכון של הטקסט.
במאמרם הציגו החוקרים שני מקודדים- מקודד אחד מבוסס רשת עצבית מלאכותית מסוג "Transformer" ומקודד שני מבוסס רשת מסוג "DAN- Deep Averaging Network".
אנחנו בחרנו להשתמש בפרויקט במודל מאומן מראש מסוג "DAN" אשר השיג תוצאות דיוק נמוכות במעט אך הוא רץ בסיבוכיות זמן לינארית ובעל צריכת זיכרון נמוכה, ובכך התאפשר להריצו על מחשבנו האישיים. המודל בו השתמשנו נלקח מספריית TensorFlow Hub.

המודל עצמו אומן באופן הבא- - בשלב הראשוני הטקסט מומר לאותיות קטנות , ועובר Tokenization ע"י האלגוריתם (TreeBank(PTB. אל הרשת מסוג DAN מועברים שיכוני המילים שנוצרו בשלב הקודם. המקודד מסוג DAN מחשב את הוקטור הממוצע של כל שיכוני המילים והרצפים בני שני המילים. לאחר מכן, הוקטור הממוצע מועבר לרשת זרימה קדימה בת 4 שכבות. מתקבל וקטור בעל 512 ממדים כפלט. כדי ללמוד ולכוונן את המקודד, המקודד ייצר שיכונים ששימשו לאימון מס' מודלים אחרים על משימות ייעודיות ומגוונות בתחום הNLP. בין היתר המודל מאומן על מאגר הנתונים SLNI בדומה לinferSent.

את שלושת המודלים הבאים שניסינו בחרנו לקחת מהספרייה "Sentence Transformers", ספריה זו מספקת דרך פשוטה לחשב שיכונים עבור טקסט. היא כוללת מגוון של מודלים מאומנים מראש שעברו כוונון פרמטרים כך שיוכלו לשמש לפתרון מגוון של משימות. המודלים מבוססים על רשתות עצביות מבוססות ארכיטקטורה של Transformers והם הוערכו באופן נרחב בנוגע לאיכות שלהם ביצירת שיכונים עבור טקסט.

- **MPNet**- המודל שהשתמשנו בו במסגרת הפרויקט מבוסס על המודל MPNet שפותח ע"י חוקרים מחברת Microsoft ופורסם לראשונה במאמר [35] .
MPNet הוא מודל מאומן מראש לייצוג שפה אשר יכול לייצר שיכונים עבור משפטים ופסקאות קצרות.
החידוש במודל זה היה המשימה עליה אומן- החוקרים הציעו שיטת אימון המשלבת בין שיטת הMLM שהוצעה במאמר BERT לבין שיטת הPLM(Permuted language model) שהוצעה במאמר Xlnet. על שיטת הMLM הסברנו בפרק המושגים, אך נזכיר כי זו שיטה שבה חלק מאסימוני הקלט ממוסכים והמשימה של המודל היא לחזות את האסימונים הממוסכים.
שיטת הPLM היא שיטה שבה המודל מאומן לחזות את האסימון הבא בקלט בהינתן האסימונים הקודמים לו, כמו מודלי שפה מסורתיים, אבל במקום לקרוא את המשפט באופן סדרתי , הוא לומד לחזות את המילה הבאה בקלט על כל הפרמוטציות האפשריות של הקלט.
המודל עצמו אומן על משימה משולבת של MLM וPLM שבה הקלט עובר פרמוטציה כלשהי, ומחולק לשני חלקים , החלק השמאלי הוא חלק שאותו המודל מקבל כקלט והחלק הימני הוא חלק ממוסך.
המודל מאומן לחזות את החלק הימני הממוסך של הקלט.

אנחנו השתמשנו במודל המאומן "all-mpnet-base-v2" שמציעה הספרייה. מודל זה השתמש במודל המאומן מראש MPnet כבסיס , עבר כוונון פרמטרים ואומן פעם נוספת. האורך המקסימלי של טקסט המתקבל כקלט למודל זה הוא טקסט המכיל כ512 מילים, טקסט יותר ארוך מזה נחתך. המודל מחזיר וקטור שיכון עבור הטקסט בן 384 מימדים . מבין כל המודלים באתר, מודל זה מציע את הדיוק הגבוה ביותר במגוון משימות ולכן בחרנו לבחון אותו.

- **MiniLM** – מודל שפותח ע"י חוקרים מחברת Microsoft ופורסם לראשונה במאמר [36].
במאמר, מציעים החוקרים שיטה של "זיקוק ידע" (הוסבר בפרק המושגים), באמצעותה ניתן לבצע דחיסה של מודל שפה מאומן מראש המבוסס ארכיטקטורת Transformer. החוקרים מציעים להשתמש במנגנון הattention שמצוי בשכבה האחרונה של הTransformer כדי לעזור ל"סטודנט" (המודל הדחוס) לחקות באופן עמוק את "המורה" (מודל המקור) ומראים כי העברת ידע רק מהשכבה האחרונה של המורה משיגה תוצאות טובות ביותר בהשוואה לזיקוק של הידע שכבה אחר שכבה. החוקרים בחנו את השיטה שלהם על המודל BERT ועל מודל נוסף מאומן מראש – "UniLM v2" כמודלים בתפקיד "המורה". החוקרים ייצרו ממודלים אלו מספר מודלים מסוג "סטודנט" אשר כונו MiniLM. מודלים אלו פורסמו לשימוש פומבי.
השיטה של החוקרים התגלתה כמוצלחת והצליחה לשמר כ99% מהדיוק של המודלים בחלק מהמשימות.
המודל המאומן מראש שאותו הציעה הספרייה ובו השתמשנו נקרא "all-MiniLM-L6-v2". מודל זה הוא מודל דחוס שנוצר מדחיסת המודל המאומן מראש "UniLM v2". המודל הדחוס הוא בן 6 שכבות.
הוא עבר כוונון פרמטרים ואומן שוב. האורך המקסימלי של טקסט המתקבל כקלט למודל זה הוא

טקסט המכיל כ512 מילים והפלט הוא וקטור בן 384 מימדים. בחרנו לבחון מודל זה מאחר וצוין באתר כי מודל זה מהיר פי 5 מMPNet ועדיין משיג תוצאות איכותיות.

- **DistilRoBERTa-** RoBERTa הוא מודל שהוצע ע"י חוקרים מFacebook במאמר [37]. המודל נבנה על בסיס המודל BERT אך עם מס' שינויים , בין היתר הסרה של האימון על משימת הNSP, ואימון עם קצב למידה גבוה יותר על קבוצות גדולות יותר. החוקרים הראו כי כוונון מחדש של תהליך הלמידה של BERT יכול לשפר את התוצאות של BERT במגוון משימות NLP.
אנחנו השתמשנו בפרויקט במודל שהוצע ע"י הספרייה ונקרא "all-distilroberta-v1".
DistilRoBERTa הוא מודל של RoBERTa אשר עבר תהליך זיקוק. המודל בן 6 שכבות ואומן באותה צורה שאומן המודל DistilBERT (מודל מזוקק של BERT שצמצם את הגודל של המודל ב40% אבל שמר על 97% מהיכולות של המודל [38]). המודל אומן מראש על OpenWebTextCorpus (מאגר נתונים קטן פי 4 מזה שהשתמשו כדי לאמן את RoBERTa).
האורך המקסימלי של טקסט המתקבל כקלט למודל זה הוא טקסט המכיל כ512 מילים והפלט הוא וקטור בן 768 ממדים. בחרנו לבחון במודל זה כיוון שהוא מודל דחוס, שהשיג תוצאות גבוהות במגוון משימות וניתן להריצו בקלות על המחשבים האישים שלנו.

תכונות תחביריות

שיטות ממשפחה זו מייצרות תכונות על בסיס המבנה התחבירי של המסמך- יכולות להתבסס על ניתוח תחבירי שהתבצע על הטקסט כמו ניתוח תלויות של הטקסט (Dependency Parsing) או ניתוח חלקי הדיבר של הטקסט. לא מצאנו מודלים מאומנים מראש או אלגוריתמים בשימוש נרחב היוצרים תכונות תחביריות מהטקסט. עם זאת כן מצאנו מספר מחקרים בהם חקרו את ההשפעה של תכונות תחביריות מבוססות חלקי דיבר על משימות סיווג [41] [40] [39].

לאחר שביצענו מספר ניסויים על התכונות הנפוצות יותר שבהן נהוג לייצג טקסט (המשפחות הקודמות אותן הזכרנו), החלטנו לנסות גם לחקור את ההשפעה של תכונות תחביריות מבוססות חלקי דיבר על יכולות הסיווג של המודל שלנו ולבדוק האם ייתכן כי לפרסומים קריטיים יש מבנה תחבירי מסוים שיכול לסייע בזיהוי שלהם.
על הניסויים שביצענו, והסבר על התכונות התחביריות אותן יצרנו מהטקסט, נרחיב בהמשך בפרק המסביר את הניסויים.

בחירת אלגוריתם למידה + אימון המודל

בשלב זה בפרוייקט היה עלינו לבחור אלגוריתם למידה ולאמן אותו על הוקטורים שהנדסנו ממאגר הנתונים בשלב הקודם, במטרה לייצר מסווג בינארי שידע להבחין בין הדוגמאות הקריטיות לדוגמאות שאינן קריטיות.
החלטנו לא לבחור אלגוריתם למידה אחד, אלא לבחון מס' אלגוריתמי למידה בשלב הניסויים ולאחר מכן לבחור את המסווג שהניב את התוצאות הטובות ביותר כמסווג הסופי.

כעת נציג את אלגוריתמי הלמידה אותם בחרנו לבחון במסגרת הפרוייקט :

- **SVM(Support Machine Vector)-** אלגוריתם זה מקבל אוסף של דוגמאות מתויגות במרחב n-ממדי, ומנסה למצוא מישור המפריד בצורה טובה כמה שניתן בין דוגמאות האימון השייכות לקטגוריות השונות.
המסווג הנוצר באמצעות מודל SVM הינו לינארי, כאשר חלוקת הדוגמאות במרחב הוקטורי נעשית באופן כזה שייווצר מרווח גדול ככל האפשר בין המישור המפריד לבין הנקודות הממוקמות הכי קרוב אליו. מרווח זה מכונה שוליים(margin) , כאשר בצד האחד של השוליים נמצאות דוגמאות עם label אחד, ובצד השני נמצאות הדוגמאות עם ה- label השני.
מסווגים ליניאריים מוגבלים ביכולת ההכללה שלהם בגלל הפשטות שלהם. לכן, כאשר לא ניתן להפריד אוסף דוגמאות באמצעות מפריד לינארי, משתמשים ב"הפרדה א-לינארית". גישה זו מאפשרת להשתמש ב SVM-לסיווג לא לינארי ,על ידי טרנספורמציה לא לינארית, כמו למשל "תעלול הגרעין" . במסגרת שימוש ב"תעלול הגרעין" מבצעים מיפוי של הדוגמאות למרחב אחר, בו ניתן למצוא עבורן הפרדה לינארית, וממילא יהיה אפשר להשתמש באלגוריתם SVM.
המפריד הליניארי במרחב הוקטורי החדש הוא מפריד לא ליניארי במרחב המקורי[12] .
אנחנו השתמשנו בפרויקט בSVM לא לינארי עם גרעין מסוג "rbf" .אלגוריתם זה מומש ע"י ספריית sklearn. פונקציית הrbf מחשבת עד כמה שני וקטורים במרחב דומים אחד לשני.

- **MLP(Multi-layer Perceptron)-**
אלגוריתם זה הוא אלגוריתם למידה מפוקחת המבוסס על רשת עצבית מלאכותית עם ארכיטקטורה של רשת זרימה קדימה. האלגוריתם לומד פונקציה $f: R^m \rightarrow R^O$ כאשר m הוא מספר הממדים של וקטור הקלט וO הוא מספר הממדים של הפלט. בהינתן סט של תכונות $X = x_1, x_2, \dots, x_m$ וסיווג y, האלגוריתם יכול ללמוד פונ' קירוב עבור משימת הסיווג.
האלגוריתם מסוגל ללמוד גם פונקציות שאינן ליניאריות. אנחנו השתמשנו ב"MLP classifier" שמומש ע"י ספריית sklearn. אלגוריתם זה מאומן באמצעות "שיטת הפעפוע-לאחור", שמטרתה לכוונן את משקלי הקשתות על הרשת במהלך תהליך האימון [42].

- **Naive Bayes-** משפחה של אלגוריתמי למידה מפוקחת הסתברותיים שעושים שימוש בחוק בייס ומתבססים על ההנחה הנאיבית כי אין תלות בין תכונות האובייקטים המסווגים כאשר סיווגם ידוע. על אף שהנחה זו אינה תמיד מדויקת, היא מקלה מאוד על חישוב ההסתברויות.
המסווג שנוצר לאחר הפעלת אלגוריתם זה הוא הסתברותי- הוא מחשב את ההסתברות של האובייקט להיות שייך לכל אחת ממחלקות הסיווג ופולט כתשובה את המחלקה עם ההסתברות הגבוהה ביותר [43].
נזכיר כי חוק בייס מאפשר לחשב את ההסתברות המותנית של מאורע כאשר יודעים את ההסתברויות המותנות ההפוכות.
במסגרת הפרוייקט השתמשנו בשני אלגוריתמים ממשפחת *Navie Bayes* :
Multinomial Naïve Bayes- מסווג זה מתאים לסיווג של וקטור המכיל תכונות דיסקרטיות כמו מודל *Bag of words* אשר סופר מופעים של מילים. מודל זה עשוי להתאים גם עבור שברים כמו במקרה של *TFIDF* . לכן, החלטנו לבחון אלגוריתם למידה זה בשילוב עם מודלים ממשפחת "התכונות הלקסיקליות" .
Gaussian Naïve Bayes- מסווג זה מתאים לשימוש במקרה של סיווג של וקטור המכיל תכונות רציפות, ולכן בחרנו לשלב מודל זה עם מודלים ממשפחת "התכונות הסמנטיות" שיוצרות שיכונים מהטקסט.
שני האלגוריתמים שהשתמשנו בהם מומשו ע"י ספריית *sklearn*.

- **Random Forest-** זהו אלגוריתם למידה מבוסס יער עצי החלטה. במסגרת תהליך האימון נבנים עצים רבים המשמשים כוועדה . הכוונה בוועדה שהיא המסווג מחליט את הסיווג שלו על פי כל חברי הועדה- בעת קביעת הסיווג של האובייקט, כל עץ פולט לאיזו מחלקה לדעתו משתייך האובייקט והמחלקה שמקבלת את מירב הקולות מוחזרת כפלט של המסווג.
בעת שלב האימון, נבנים עצי ההחלטה באמצעות שני מקורות של רנדומליות כדי לייצר שונות בין העצים :
1) כל עץ נבנה מתת קבוצה של סט האימון. תת הקבוצה נדגמת מכלל קבוצת הדוגמאות באמצעות דגימה רנדומלית עם חזרה .
2) בעת בניית עץ החלטה רגיל, כאשר מגיע הזמן לפצל צומת, אנו בוחנים כל תכונה אפשרית ובוחרים את התכונה שמייצרת את ההפרדה הגדולה ביותר בין הדוגמאות. לעומת זאת, בעת בניית עת ביער אקראי, בוחנים רק תת קבוצה של התכונות בעת פיצול צומת.

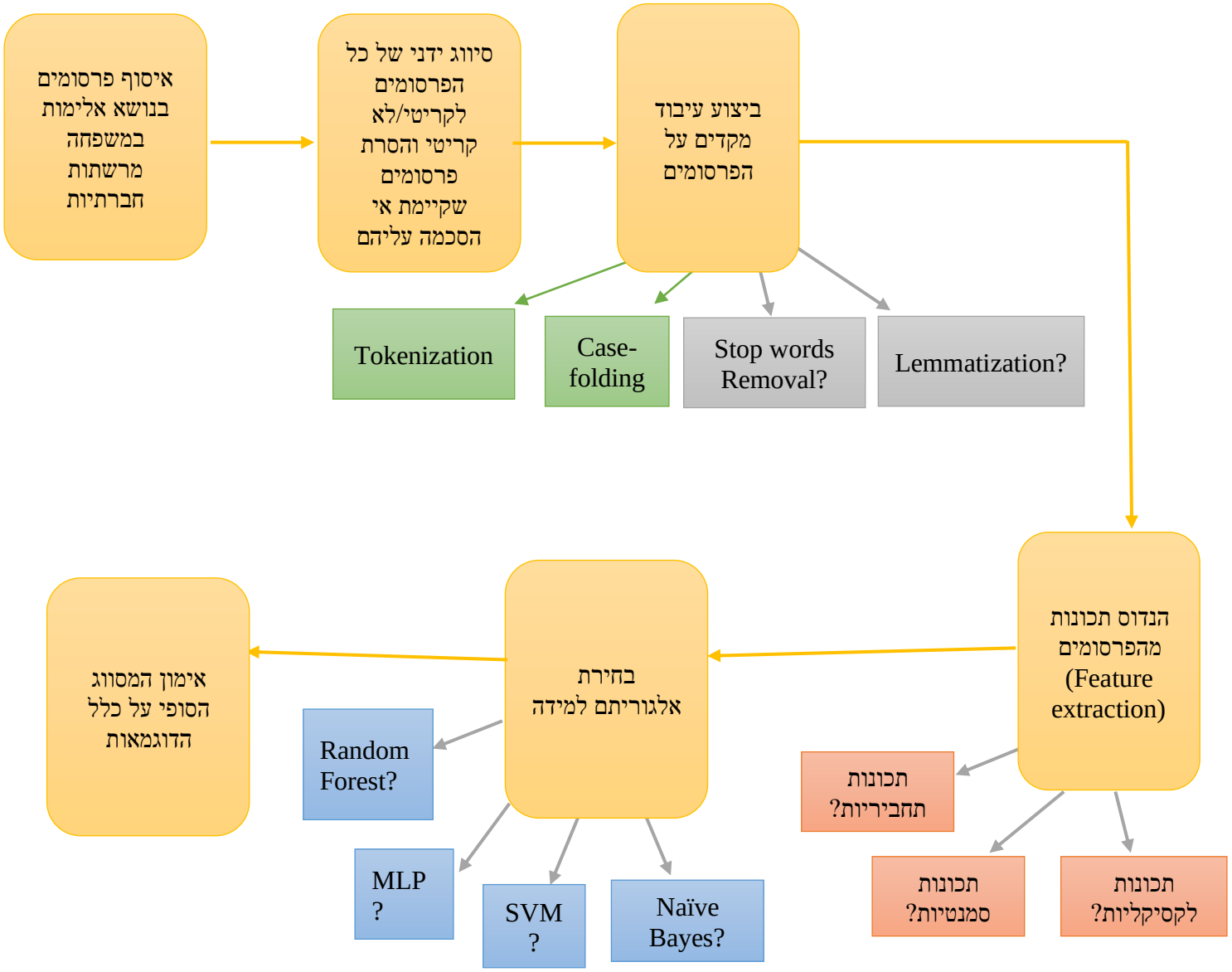
אנחנו השתמשנו במסווג *RandomForest* של *sklearn* . המימוש של *sklearn* לא משתמש בהכרעת הרוב כדי לקבוע את המחלקה אליה משתייך האובייקט , אלא מחשב את ההסתברות הממוצעת של כל מחלקה על פני כל העצים ומחזיר את המחלקה שההסתברות שלה הגבוהה ביותר. ההסתברות של מחלקה מסויימת בעץ יחיד היא החלק היחסי של הדוגמאות ממחלקה זו מכלל הדוגמאות בעלה [44].

יצירת מסווג סופי

בתום הניסויים , לאחר שבחרנו שיטות עיבוד מקדים שונות, משפחות שונות של תכונות ואלגוריתמי למידה שונים, יצרנו את המסווג הסופי. כדי ליצור את המסווג הסופי, בחרנו צירוף של שיטת עיבוד מקדים למסמך, שיטת ייצוג למסמך ואלגוריתם למידה אשר הניב לנו את אחוז הדיוק הגבוה ביותר.
בשלב זה , אימנו את המסווג הסופי על כל 400 הדוגמאות במאגר וכעת הוא מוכן להפעלה – בהינתן פרסום מרשת חברית המסווג יסווג האם הפרסום הוא קריטי/לא קריטי.

את המסווג הסופי שהחלטנו לבחור נציג בתום פרק הניסויים.

תיאור סכמתי של דרך הפתרון שלנו לבעיה :



מתודולוגיה ניסויית ותיאור הניסויים

בפרק זה נציג את הניסויים שביצענו במסגרת הפרויקט, הפרמטרים שהחלטנו לבחון והתוצאות של כל ניסוי.
באופן כללי, מטרת הניסויים העיקרית שלנו הייתה לייצר מסווג אופטימלי שיוכל להתמודד עם הבעיה הבינארית של סיווג פרסומים לקריטי/לא קריטי.

במסגרת הפרוייקט ביצענו שלושה ניסויים מרכזיים :

א. בניסוי הראשון נבחנו שלושה משתנים –
שיטות שונות לעיבוד מקדים של הטקסט, תכונות שונות המייצגות את הטקסט ואלגוריתמי למידה שונים היוצרים מסווגים שונים. התכונות השונות שבחנו נלקחו ממשפחת התכונות הלקסיקליות והסמנטיות בלבד שכן אלו המודלים המקובלים יותר לייצוג טקסט.
רצינו לבדוק במסגרת ניסויים אלו את ההשפעה של כל אחד מהמשתנים על אחוז הדיוק של המודל ובנוסף לבדוק את האינטראקציה הכפולה והמשולשת בין המשתנים השונים.
במסגרת הניסוי הראשון ראינו כי לסוג התכונות המייצגות את הטקסט יש השפעה רבה על יכולת הסיווג של המודל ולפיכך החלטנו במסגרת השלבים הבאים לבחון שילובים בין תכונות השייכות למשפחות שונות.

ב. בניסוי השני בחנו קומבינציות של תכונות לקסיקליות ותכונות סמנטיות. הכוונה בקומבינציות היא לפעולת השרשור- שרשרנו וקטורי תכונות של שתי משפחות ביחד ובחנו את ההשפעה של הקומבינציות על אחוז דיוק המודל.

ג. בניסוי השלישי בחנו קומבינציות של תכונות תחביריות עם תכונות לקסיקליות ו/או סמנטיות. בחנו שילובים כפולים ומשולשים.

לאחר התייעצות עם הגורמים המנחים בפרוייקט, ולאור העובדה כי מאגר הנתונים שלנו קטן והכיל 400 דוגמאות, לא חילקנו את המאגר לקבוצת אימון וקבוצת מבחן אלא השתמשנו בשיטת "תוקף צולב"(cross validation) על מנת להעריך כל אחד מהמודלים שניסינו. הציון על פיו הערכנו כל מודל שבחנו היה אחוז הדיוק של המודל בסיווג הפרסומים ל2 המחלקות- קריטי/לא קריטי.
במסגרת שיטת "תוקף צולב", בכל פעם שבחנו מודל מסוים, מאגר הנתונים שלנו חולק ל5 קבוצות. המודל עצמו נבחן במסגרת 5 איטרציות, כאשר בכל איטרציה המודל אומן על 4 מתוך 5 הקבוצות ובסוף האימון נוצר מסווג. לאחר מכן, חושב אחוז הדיוק של אותו מסווג כמס' ההצלחות בחיזוי על קבוצת המבחן. הציון להערכת המודל כולו התקבל בסיום 5 האיטרציות והיה אחוז הדיוק הממוצע של 5 המסווגים שנוצרו.

השתמשנו ב"stratifiedkfold " שמציעה ספריית sklearn כדי לבצע את התוקף הצולב. ממשק זה אפשר לנו ליצור קבוצות המכילות מס' שווה של דוגמאות שליליות וחיוביות עבור כל אחד מהfolds. היתרון בכך הוא שאם המסווג טועה במידה שונה על מחלקות שונות (טועה יותר על מינוסים למשל), נרצה שעדיין המדד יהיה אחיד בכל ה- folds ולא מושפע מזה. בנוסף חשוב לציין שתמיד ביצענו את אותה חלוקה על מאגר הדוגמאות (השתמשנו ב random seed=42) על מנת לבטל את השינויים שיכולים להיגרם כתוצאה מחלוקה שונה של הדוגמאות לקבוצת אימון ומבחן.

בעמודים הבאים נפרט על כל אחד מהניסויים שביצענו ונדווח על התוצאות שהתקבלו לאחר הרצת כל ניסוי.

ניסוי מס׳ 1

כפי שציינו קודם, במסגרת ניסוי זה נבחנו שלושה משתנים שעשויים להשפיע על יכולת הסיווג של המודל:

1. **שיטות עיבוד מקדים לטקסט-** בדקנו ארבע שיטות שונות לעיבוד מקדים של כל אחד מהפרסומים.
כפי שציינו קודם, כל הפרסומים כבר עברו תהליך של tokenization וcase-folding. לכן כל אחת מהשיטות הנוספות שנוסתה, נוסתה על גבי העיבוד המקדים המינימלי הזה.
השיטות שנבחנו:

- Minimal preprocess- זוהי למעשה "קבוצת הביקורת". במסגרת שיטה זו לא השתמשנו בעיבוד נוסף מעבר לעיבוד המקדים המינימלי שכבר בוצע.
- Stop words removal- במסגרת שיטה זו ביצענו הסרה של מילים מרשימת "stop words" לכל אחד מהפרסומים, נוסף לעיבוד המקדים המינימלי.
- Lemmatization- במסגרת שיטה זו העברנו כל אחד מהפרסומים תהליך של Lemmatization- עבור כל פרסום, כל אחד מהאסימונים הוחלף בצורת השורש שלו, נוסף לעיבוד המקדים המינימלי.
- Stop words removal+ Lemmatization-- במסגרת שיטה זו כל אחד מהפרסומים עבר עיבוד מקדים באמצעות שתי השיטות שתוארו קודם, נוסף לעיבוד המקדים המינימלי.

2. **סוג התכונות שמייצגות את הטקסט-** בניסוי זה בדקנו מודלים שונים המייצרים תכונות מהטקסט ממשפחת התכונות הלקסיקליות וממשפחת התכונות הסמנטיות, אותן הצגנו בפרקים הקודמים.
אלו המודלים המוכרים יותר לייצוג טקסט ולכן בחרנו לבדוק כל אחד מהם באופן עצמאי ולבחון איזה מודל עשוי לייצר תכונות שייצגו את הפרסומים שלנו במאגר בצורה מיטבית.
המודלים אותם בחנו:
ממשפחת התכונות הלקסיקליות-

- Bag of words- הוא ישמש כמעיין "קבוצת ביקורת" מאחר וזה מודל פשוט לייצוג טקסט.
- TFIDF

ממשפחת התכונות הסמנטיות-

- fastText
- doc2vec
- inferSent
- MiniLM
- MPNet
- DistilRoBERTa
- Universal Sentence Encoder

3. **אלגוריתמי למידה-** את אלגורימי הלמידה שבחנו בניסוי זה בחנו תוך שימוש בערכי ברירת המחדל שמציעה ספריית sklearn. האלגוריתמים בהם השתמשנו בשלב זה:

- Naïve bayes- עבור משפחת התכונות הלקסיקליות השתמשנו ב Bayes Multinomial Naïve ועבור משפחת התכונות הסמנטיות השתמשנו בGaussian Navie Bayes, הסיבות לכך הוסברו בפרקים הקודמים.
- Multilayer Perceptron(MLP)
- SVM
- Random Forest

ברור לנו כי בין שלושת המשתנים צפוי להיות קשר כלשהו מאחר וכל אחד מהמשתנים משפיע על שלב אחר בתהליך יצירה של המסווג:
- שיטות שונות של עיבוד מקדים יגרמו למודלים מסוימים ,כמו Bag of words למשל , ליצור תכונות שונות מהטקסט.
- סוג התכונות שייצגו את הטקסט (ניסינו מגוון מודלים שונים) ישפיע בשלב אימון אלגוריתם הלמידה, שהרי אלגוריתם הלמידה משתמש בתכונות אלו כדי "ללמוד" את מאגר הנתונים והוא מחזיר את המסווג שהכי מתאים לסיווג התכונות הספציפיות.

-סוג אלגוריתם הלמידה ישפיע על המסווג הסופי שיימצא- לכל אלגוריתם למידה יש דרך אחרת בה הוא ילמד את התכונות ולכן כל אלגוריתם למידה יחזיר מסווג שונה גם כשמדובר באותן התכונות.

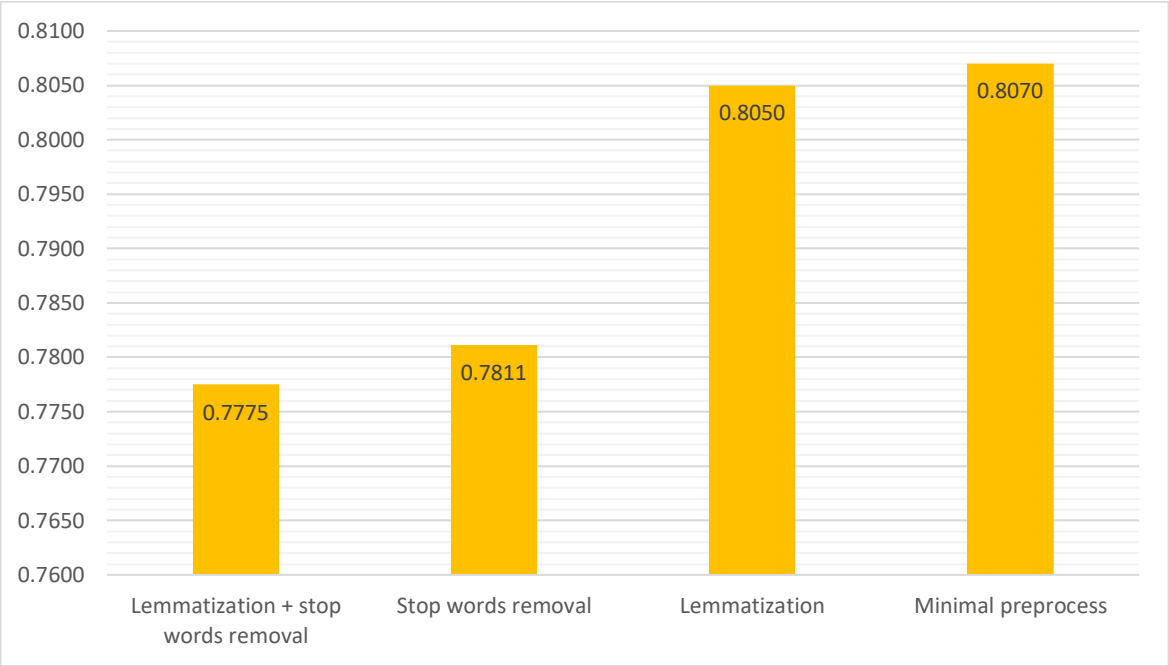
הניסוי עצמו התבצע כך-
בכל שלב בניסוי נבחנה קומבינציה משולשת של 3 המשתנים- נבחרה שיטת עיבוד מקדים אחת, סוג אחד של וקטור תכונות עבור כל פרסום ונבחר אלגוריתם למידה אחד. אלגוריתם הלמידה אומן בכל פעם על 4 קבוצות באמצעות שיטת תוקף צולב כך שבכל שלב בניסוי נוצרו חמישה מסווגים שונים כתוצאה מהקומבינציה המשולשת ואחוז הדיוק של המודל נקבע כממוצע על חמשת המסווגים. במסגרת ניסוי זה נוסו כל הקומבינציות האפשריות בין 3 המשתנים.

תוצאות ניסוי מספר 1:

מאחר וקיבלנו תוצאות גולמיות רבות, החלטנו להציג את הממצאים באמצעות גרפים המתייחסים לממוצעים של תנאים שונים.

בדיקת ההשפעה של כל אחד מהמשתנים על אחוז הדיוק הממוצע:

השפעת שיטת עיבוד מקדים-



תרשים 1: אחוז הדיוק הממוצע של המודל כפונקציה של שיטת העיבוד המקדים.
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור מודלים בהם הטקסט עבר עיבוד מקדים בשיטת עיבוד מקדים מסוימת , כאשר הממוצע עצמו מחושב על פני כל אלגוריתמי הלמידה ועל פני כל סוג התכונות המייצגות את הטקסט.

ניתן לראות מהתוצאות כי מודלים אשר עברו עיבוד מקדים באמצעות שיטת הביקורת שלנו – "Minimal preprocess" שבה לא ביצענו שום עיבוד מקדים נוסף, הניבו בממוצע את התוצאות המיטביות. בנוסף , מודלים שעברו עיבוד מקדים באמצעות שיטת הLemmatization היו בעלי אחוז דיוק ממוצע קרוב מאוד לשיטת הביקורת. לעומת שתי שיטות אלו, מודלים שעברו עיבוד מקדים באמצעות שיטת הסרת stop words הניבו תוצאות פחות טובות בממוצע, נמוכות בכ2% ואילו מודלים שעברו עיבוד מקדים באמצעות השיטה המשולבת הניבו בממוצע את התוצאות הנמוכות ביותר, נמוכות בכ2.5% מהשיטות המוצלחות יותר.

למרות ששיטת הסרת stop words היא שיטת עיבוד מקדים נפוצה, מצאנו מס' עדויות בספרות [28] [45] לכך שהסרת stop words פגעה במודלים שניסו לבצע את משימת "ניתוח הרגש" (sentiment analysis) על ציוצים מtwitter. במקרים אלו התברר שאכן מילות ה"stop words" נשאו מידע חיוני שעזר בסיווג.

במקרה שלנו, ייתכנו מס׳ הסברים לכך שהסרת stop words פגמה באחוז הדיוק של המסווג.
ראשית, ייתכן שגם במקרה שלנו, בדומה למשימת ניתוח הרגש- הסרת "stop words" מסירה מידע חיוני מהמשפט שגורם לאובדן הקשר בחלק מהמקרים.
אם נסתכל על הדוגמה הבאה למשל-
"I left my abusive ex boyfriend today for good. He's someone I never want to encounter again after all our legal shit is over with. This big weight is off my chest."

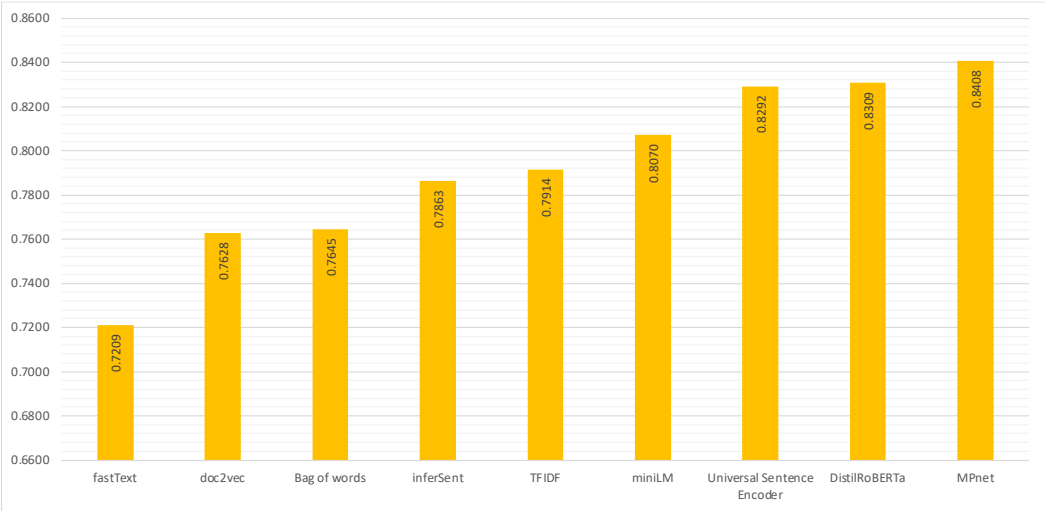
לאחר הסרת stop words נקבל את הדוגמה הבאה:
"left abusive ex boyfriend today good . someone never want encounter legal shit . big weight chest"

ניתן לראות שהביטוי "This big weight is off my chest" שהוא בעל משמעות חיובית איבד את ההקשר שלו לאחר הסרה של stop words והפך לביטוי עם רגש שלילי- "big weight chest".

שנית, סיבה נוספת בגינה שיטת הסרת "stop words" נכשלה יכולה להיות הרשימה עצמה. מחקרים אחרונים [46] הראו כי רשימות "stop words" נפוצות ברשת שמשתמשים בהן פעמים רבות באופן עיוור עשויות לסבול ממחדלים מפתיעים- אי התאמה של רשימות כאלו לתהליך הTokenization, רבות מהן כוללות מילים שנויות במחלוקת כמו cry,great או כוללות רק את צורת השורש של מילה מסוימת ולא את כל הצורות בהן יכולה להופיע.
רבות מהבעיות של רשימות אלו שהן מיוצרות על בסיס סטטיסטיקות שנעשות על מאגרי מסמכים (חיפוש של המילים הנפוצות ביותר) אך יש מילים נפוצות ביותר שגם עשויות להביע רגש כמו great ולכן הבעיה של יצירת רשימת "stop words" מיטבית אינה טריוויאלית.
אם כך ייתכן והשימוש ברשימת ה-"stop words" הספציפית הזו לא הייתה מיטבי עם מאגר המסמכים שלנו ורשימה אחרת, שיותר מותאמת לדוגמאות שלנו הייתה יכולה להניב שיפור בתוצאות.

ניתן להניח כי הירידה בתוצאות בשיטה המשולבת– stop words+Lemmatization removal נבעה מסיבות דומות, כאשר תופעת אובדן ההקשר של המשפט גברה עוד יותר מאחר והמילים גם הוחלפו בצורת השורש שלהן.

השפעת סוג התכונות



תרשים 2: אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות המייצגות הטקסט.
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור מודלים בהם השתמשנו בסוג מסוים של תכונות המייצגות טקסט , כאשר הממוצע עצמו מחושב על פני כל אלגוריתמי הלמידה ועל פני כל שיטות העיבוד המקדים.

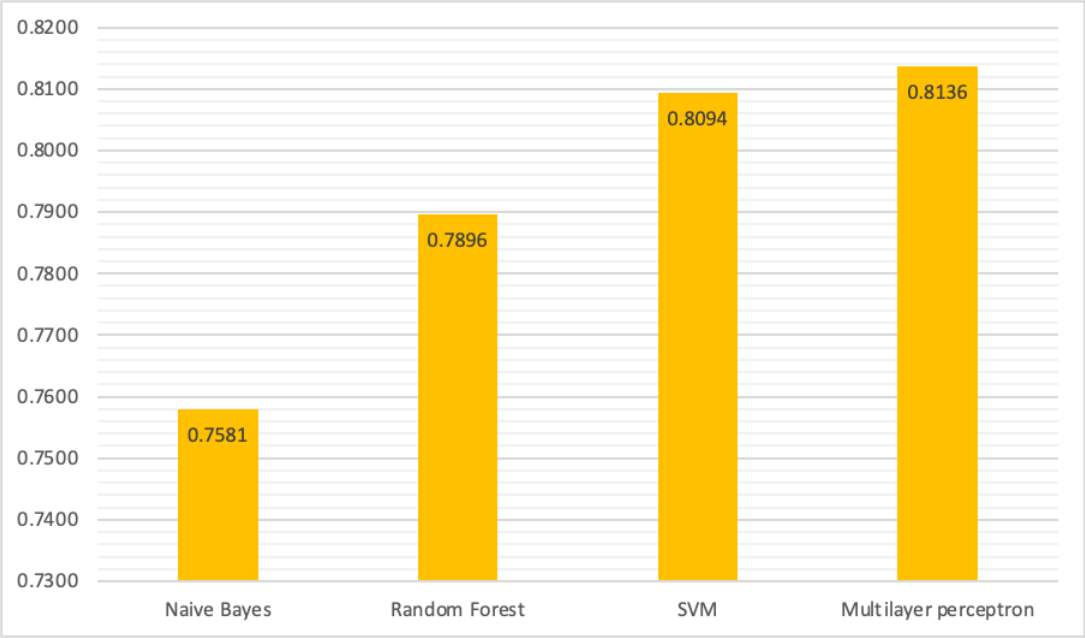
ניתן לראות כי שימוש במודל "MPNet" ליצירת תכונות הביא ליצירת מסווגים בעלי אחוז הדיוק הממוצע הגבוה ביותר. מודלים נוספים ליצירת תכונות שהניבו אחוז דיוק גבוה בממוצע היו DistillRoBERTa ו-Universal Sentence Encoder.

לעומתם, ניתן לראות שימוש במודל "fasttext" ליצירת תכונות הביא ליצירת מסווגים פחות מוצלחים , ואחוז הדיוק הממוצע שלהם היה נמוך יחסית. הייתה לנו ציפייה שמודל fastText יניב תוצאות טובות יותר ממודל "Bag of words" שכן מודל fastText מייצר תכונות סמנטיות- שיכוני מילים ובכך משמר את המשמעות הסמנטית שלהם. עם זאת, חשוב לזכור כי במסגרת שימוש במודל זה ביצענו עבור כל מסמך ממוצע על כל שיכוני המילים שמכילים אותו. לכן, הסבר אפשרי לתופעה הוא שהתכונות הסמנטיות איבדו ממשמעותן כתוצאה מביצוע הממוצע. ככל הנראה שימוש בשיכוני מילים עשוי להתאים למסמכים קצרים יותר המכילים מספר מועט של מילים כך שביצוע הממוצע עדיין יצליח לשמר את המשמעות הסמנטית של המסמך.

הבחנה נוספת שניתן להסיק היא כי "doc2vec" ו "inferSent" לא ייצרו תכונות סמנטיות מוצלחות מאוד. ניתן לראות שאחוז הדיוק הממוצע שהתקבל בשימוש במודלים אלו קרוב מאוד לאחוז הדיוק הממוצע שהתקבל בשימוש במודלים פשוטים יותר כמו "Bag of words" או "TF-IDF" . הסתייגות אחת ממסקנה זו שהיא שהמודל doc2vec לא אומן מראש על מאגרי מסמכים נרחבים אלא רק על המאגר הקטן שלנו ולכן ייתכן והוא לא הגיע למלוא הפוטנציאל שלו ביצירת שיכונים.

בהשוואה לשיטת עיבוד מקדים, נמצא כי יש טווח רחב יותר לתוצאות שקיבלנו במסגרת בחינה של התכונות השונות – ההבדל באחוז הדיוק הממוצע בין מודלים שהשתמשו בתכונות הטובות ביותר לייצוג טקסט לבין מודלים שהשתמשו בתכונות הכי פחות טובות היה של כ-12% ולכן ניתן לראות שלאופן בו בוחרים לייצג את הטקסט יש השפעה גדולה על יכולת הסיווג של המודלים שהתקבלו.

השפעת סוג אלגוריתמי הלמידה



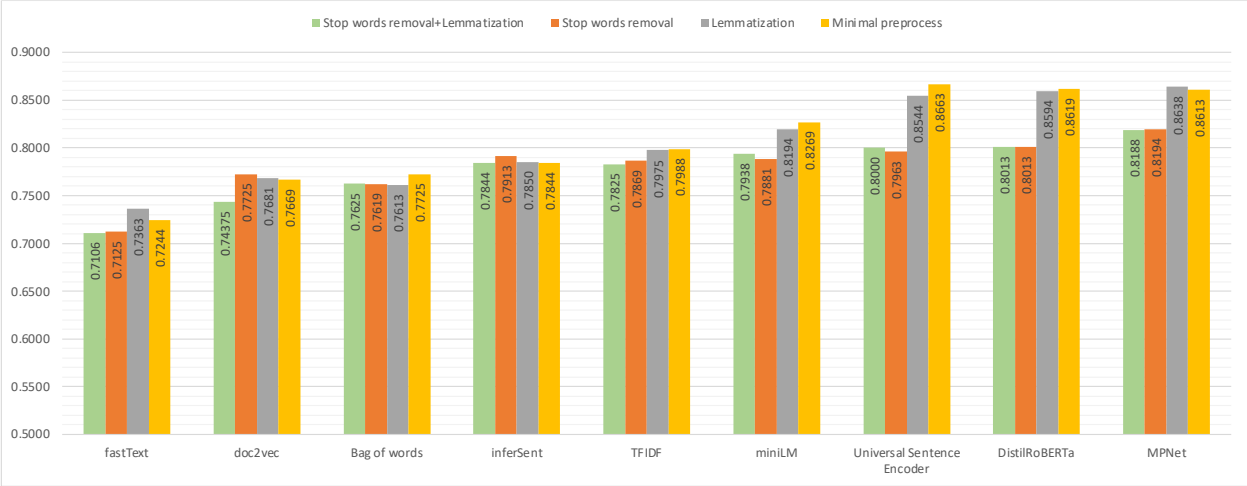
תרשים 3 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג אלגוריתם הלמידה .
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור מודלים בהם השתמשנו בסוג מסוים של אלגוריתם למידה כדי ליצור את המסווג הסופי , כאשר הממוצע עצמו מחושב על פני כל שיטות העיבוד המקדים ועל פני כל סוגי התכונות המייצגות את הטקסט.

ניתן לראות כי בממוצע אלגוריתם הלמידה שהיה המוצלח ביותר היה "Multilayer Perceptron" ואחריו היה המודל SVM. תוצאה זו אינה מפתיעה במיוחד שכן אלגוריתם זה מבוסס רשת עצבית מלאכותית, ורשתות עצביות מלאכותיות הראו תוצאות מבטיחות בשנים האחרונות במשימות סיווג ומגוון משימות אחרות [47]. עוד ניתן לראות כי משפחת האלגוריתמים של "Naïve Bayes" הניבה בממוצע תוצאות פחות בהשוואה למודלים האחרים, בהפרש של כ- 6% מהמודל "Multilayer Perceptron".

בדיקת אינטראקציות בין המשתנים על אחוז הדיוק של המסווג:

מאחר ובדקנו בניסוי זה שילוב של שלושה משתנים, סביר להניח שתהיה אינטראקציה בין המשתנים השונים, ושההשפעה של כל משתנה על התוצאות לא תהיה בלתי תלויות באחר.
לכן החלטנו לחקור גם את ההשפעה של האינטראקציות בין הזוגות השונים של התוצאות:

אינטראקציה בין משתנה שיטת העיבוד המקדים למשתנה סוג התכונות המייצגות את הטקסט

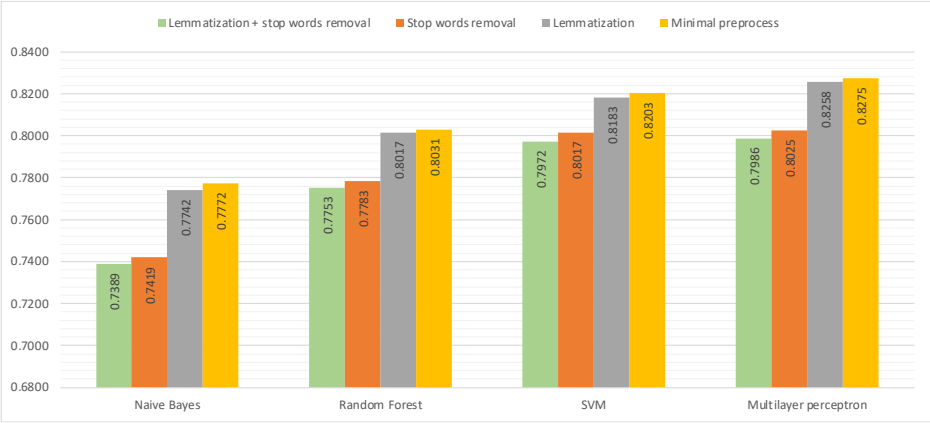


תרשים 4: אחוז הדיוק הממוצע של המודל כפונקציה של שיטת העיבוד המקדים ושל סוג התכונות המייצגות את הטקסט. כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור שיטת עיבוד מסוימת x סוג תכונות מסוים והממוצע חושב על פני 4 אלגוריתמי הלמידה השונים.

מהגרף ניתן לראות כי אכן הייתה אינטראקציה בין שני המשתנים.
בניגוד לגרפים הקודמים, פה ניתן לראות כי עבור מודלים בהם התכונות המייצגות את הטקסט נוצרו באמצעות doc2vec ו inferSent , שיטת עיבוד מקדים מסוג הסרת "stop words" הביאה לאחוז דיוק ממוצע גבוה יותר בהשוואה לשיטות עיבוד מקדים אחרות.
בנוסף, ניתן לראות כי גם כאשר משווים בין הסוגים השונים של התכונות, עבור רוב סוגי התכונות, עיבוד מקדים של Lemmatization או Minimal preprocess הביא לאחוז דיוק ממוצע גבוה יותר.

עוד ניתן לראות כי עם שילוב של עיבוד מקדים מסוג Lemmatization או Minimal preprocess , התכונות המיוצרות באמצעות המודלים Universal Sentence Encoder, DistilRoBERTa ו MPNet מייצגות את הטקסט בצורה הטובה ביותר מאחר ואחוז הדיוק הממוצע של המודלים שהשתמשו בהן היה כ86%. ניתן לראות כי שילוב בין שיטת עיבוד מקדים מינימלי למודל Universal Sentence Encoder מביא לאחוז הדיוק הממוצע הגבוה ביותר.
מגמה בולטת נוספת שהאינטראקציה בין סוג התכונות לשיטת העיבוד המקדים הייתה חזקה יותר עבור ארבעת המודלים הימניים בגרף. למשל, עבור המודל Universal Sentence Encoder, ההשפעה של שיטת העיבוד המקדים הייתה החזקה ביותר- הבדל של עד כ6% בין אחוזי הדיוק הממוצעים.
לעומת זאת, ההשפעה של שיטת העיבוד המקדים על המודלים מצדו השמאלי של הגרף לא היה חזקה במיוחד- עבור המודל Bag of words או inferSent למשל, ההשפעה של שיטת העיבוד המקדים הייתה קטנה, ככל והובילה להבדלים קטנים באחוזי הדיוקים הממוצעים.
סיבה אפשרית להבדלים בין סוגי התכונות השונות היא שמודלים כמו MPNet, DistilRoBERTa או Universal sentence encoder ככל הנראה מסתמכים יותר על הstop words בניסיון שלהם לקודד את המסמכים לוקטורים ולשמר את המשמעות הסמנטית שלהם. ככל הנראה מקודדים אלה מצליחים לייצר קידוד סמנטי איכותי יותר של המסמך באמצעות שימוש ב"stop words" . ייתכן גם כי פה נעוצה הסיבה מדוע מקודדים אלו הציגו ממוצע תוצאות טובות יותר בהשוואה למודלים אחרים שייצרו תכונות סמנטיות.

אינטראקציה בין שיטת העיבוד המקדים לסוג אלגוריתם הלמידה

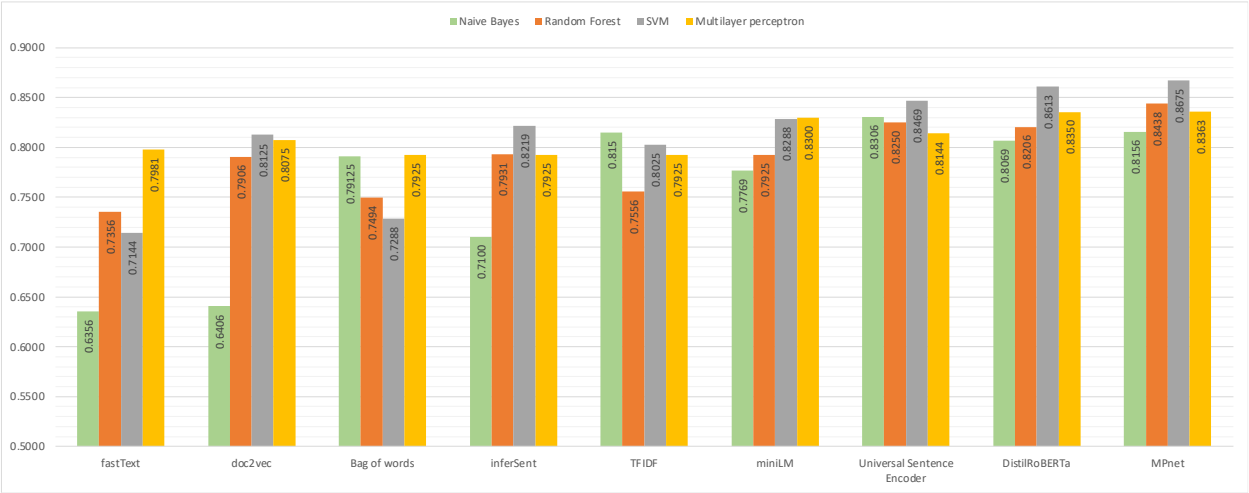


תרשים 5 : אחוז הדיוק הממוצע של המודל כפונקציה של שיטת העיבוד המקדים ושל סוג אלגוריתם הלמידה. כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל שיטת עיבוד מסוימת x אלגוריתם למידה מסוים והממוצע חושב על פני כל סוגי התכונות השונות.

ניתן לראות כי במקרה זה האינטראקציה בין המשתנים לא חזקה במיוחד, מאחר ואין תוצאות מעניינות שלא הסקנו כבר קודם, כשחקרנו כל משתנה בנפרד.
ניתן להבחין כי כמו שכבר ראינו כאשר חקרנו את ההשפעה של שיטת עיבוד מקדים, שיטות ה Minimal preprocess וה-Lemmatization מניבות תוצאות טובות יותר גם כשמשווים אותן על כל אחד מאלגוריתמי הלמידה בנפרד.
הבחנה נוספת היא שאלגוריתמי הלמידה SVM ו-MLP מניבים תוצאות טובות יותר גם כאשר משווים בין אלגוריתמי הלמידה על פני שיטות עיבוד מקדים שונות.

נקודה שכן יכולה להעיד על הבדלים בין האלגוריתמים השונים היא שעבור מודלים בהם השתמשנו באלגוריתמי משפחת ה"Naïve Bayes" , הסרת ה"stop words" פגמה מעט יותר באחוז הדיוק, בהשוואה לסוגי אלגוריתמי למידה האחרים. ככל הנראה אלגוריתמים אלו יותר הסתמכו על הstop words בעת תהליך הסיווג.

אינטראקציה בין סוג התכונות המייצגות את הטקסט לסוג אלגוריתם הלמידה



תרשים 6 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות המייצגות את הטקסט ושל סוג אלגוריתם הלמידה. כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל סוג תכונות מסוים x אלגוריתם למידה מסוים והממוצע חושב על פני כל שיטות העיבוד המקדים.

בתרשים 6 אפשר לראות אינטראקציה חזקה בין המשתנים.
ראשית, ניתן לראות שSVM פעל בצורה מיטבית על בשילוב עם תכונות סמנטיות (חוץ ממודל fastText). לעומת זאת שילוב בין תכונות שיוצרו באמצעות מודל Bag Of Words ואלגוריתם SVM הניב מודלים עם אחוז דיוק ממוצע נמוך.
הסבר אפשרי לתופעה זו הוא שהמודלים השונים המייצרים תכונות סמנטיות, ממפים את המסמכים השונים לוקטורים במרחב וקטורי ומייצרים ייצוג וקטורי מבוזר ורציף עבור המסמכים, כך שמסמכים בעלי משמעות סמנטית קרובה יהיו קרובים במרחב הוקטורי. ייצוג זה הוא מיטבי עבור SVM מאחר וSVM מנסה למצוא מישור המפריד בין הדוגמאות הקריטיות והלא קריטיות ועל כן עדיף שהייצוג שלהם יהיה בעל משמעות מתמטית במרחב ולא אקראי.
בנוסף, אלגוריתם SVM רגיש לנרמול, scaling- תכונות עם טווח ערכים גבוה עשויות להיות בעלות חשיבות גדולה יותר מאשר תכונות עם טווח ערכים נמוך[48] . לכן, ככל הנראה SVM פעל פחות טוב בשילוב עם מודל Bag Of words המייצר וקטורים לא מנורמלים בהשוואה למודל TFIDF לדוגמה, שמייצר וקטורים מנורמלים.

אינטראקציה נוספת שניתן לראות היא במשפחת אלגוריתמי "Naive Bayes" . שילוב בין אלגוריתם זה לבין ייצוג הטקסט ע"י תכונות לקסיקליות שיוצרו על ידי המודלים-Bag of words וTFIDF , הביא ליצירת מודלים בעלי אחוז דיוק ממוצע גבוה בהשוואה למודלים אחרים שיוצרו באמצעות תכונות אלו. יש לזכור שבמקרה זה הפעלנו את המודל המולטינומי של "Naïve Bayes" , שאידאלי עבור וקטורים המייצגים ספירת מופעים של מילים .
לעומת זאת, שילוב של תכונות סמנטיות והאלגוריתם "Naïve Bayes" הביא ליצירת מודלים עם יכולות סיווג הרבה פחות מוצלחות, במיוחד בשימוש עם תכונות שיוצרו באמצעות המודלים fastText, doc2vec, inferSent יש לזכור כי במקרה זה השתמשנו במודל "Gussian Naïve Bayes" שמתאים לערכים רציפים, ונראה שמודל זה פחות מתאים לשימוש עם התכונות שניסינו מהמשפחה הסמנטית. עם זאת, ניתן לראות שהייתה אינטראקציה כלשהי בין תכונות שיוצרו ע"י Universal Sentence Encoder לבין המסווג הגאוסיאני שהביאה ליצירת מודלים עם אחוז דיוק גבוה יחסית.

הבחנה נוספת היא שכאשר מוציאים את אלגוריתמי Naïve Bayes מחוץ למשוואה, התכונות הסמנטיות המיוצרות ע"י המודלים doc2vec וinferSent אכן משפרות את יכולת הניבוי של המסווגים בהשוואה לתכונות הלקסיקליות המיוצרות ע"י TFIDF וBag of words. הבחנה זו תומכת בטענה כי תכונות סמנטיות , מבוססות שיכונים עדיפות על התכונות הלקסיקליות, שלרוב מהוות ייצוג פשוט עבור הטקסט. ניתן לראות כי הירידה בתוצאות הממוצעות בשימוש במודלים doc2vec וinferSent שראינו בעמודים הקודמים נבעה בשל חוסר התאמה של אלגוריתם הלמידה הגוסיאני לתכונות. עם זאת, בשילוב של תכונות אלו עם אלגוריתמי למידה אחרים, נראה כי הן עדיפות על התכונות הלקסיקליות.

זאת ועוד, ניתן להבחין כי אנחנו מקבלים מגרף זה תמיכה למסקנה שלנו כי תכונות המיוצרות באמצעות המודלים – MPNet, Universal Sentence Encoder, DistilRoBERTa מייצגות את מאגר הנתונים שלנו בצורה הטובה ביותר. ניתן לראות כי האינטראקציה שלהן עם סוג האלגוריתם הלמידה לא חזקה במיוחד בהשוואה לסוגי תכונות אחרות וכי מודלים המשתמשים בתכונות אלו היו בעלי אחוזי דיוק גבוהים.

אינטראקציה משולשת בין שיטות עיבוד מקדים, סוג התכונות וסוג אלגוריתם הלמידה

כדי לחקור את האינטראקציה בין שלושת המשתנים, יצרנו ארבעה גרפים שונים, כאשר כל גרף מציג את אחוז דיוק המודל כפונקציה של שיטת עיבוד מקדים וסוג התכונות המייצגות את הטקסט עבור אלגוריתם למידה מסוים. הגרפים למעשה מציגים את התוצאות הגולמיות(הציון הממוצע של התוקף הצולב) שהתקבלו עבור כל קומבינציה משלושת. נרצה לבדוק האם יש שוני בין הגרפים השונים שיכול להעיד על קיומה של אינטראקציה משולשת.

הגרפים מוצגים בעמוד הבא.

להלן התוצאות:

ניתן לראות מהתוצאות מס' תופעות מעניינות.

* ניתן לראות שההשפעה של שיטת עיבוד מקדים על התכונות הנוצרות מהטקסט לאחר מכן אינה אינה דטרמינסטית. קיימת תלות בין שיטת העיבוד המקדים, סוג התכונות ואלגוריתם הלמידה.. ניקח לדוגמה את התכונות הנוצרות באמצעות מודל inferSent- קודם ראינו כי בממוצע שיטת הסרת stop words הטיבה עם מודל זה ואפשרה לייצר תכונות המייצגות את הטקסט בצורה טובה יותר. עם זאת, מהתבוננות בגרפים, אפשר לראות כי שילוב של stop words removal עם infersent ואלגוריתם למידה mlp היה פחות מיטיבי.
מגמות דומות אפשר להבחין בקרב מודלים נוספים המייצרים תכונות .

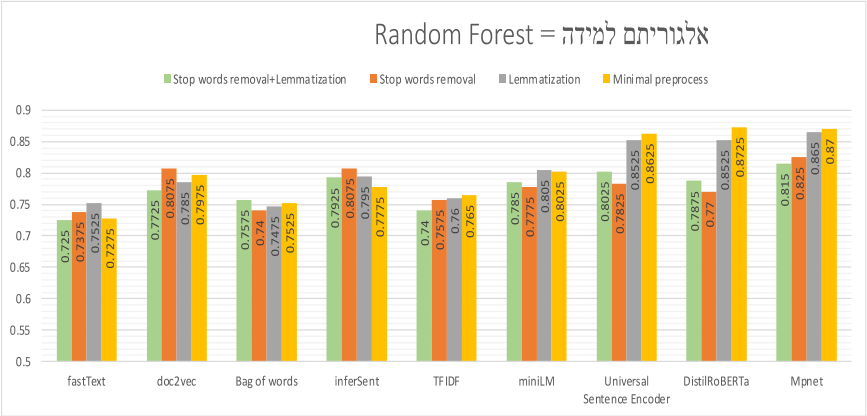
בנוסף ניתן להבחין במספר אינטראקציות משולשות בין שלושת המשתנים שהובילו למודלים בעל אחוז דיוק גבוה-
*שילוב של minimal pre process+ תכונות מסוג DistillRoBERTa+ אלגוריתם random Forest הוביל לאחוז דיוק של 87.25%. זאת למרות שבממוצע אלגוריתם random Forest יצר מסווגים פחות מוצלחים.

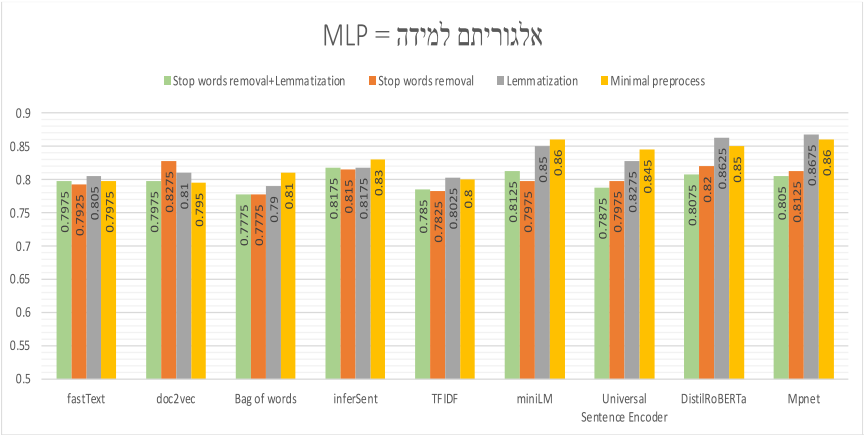
*שילוב של Lemmatization+ תכונות מסוג MPNet+ אלגוריתם MLP הצליח להגיע לתוצאות מרביות של 86.75%. נשים לב שעל אף שבממוצע ראינו כי אלגוריתם MLP יצר מסווגים עם אחוז הדיוק הגבוה ביותר, הוא דווקא לא הצליח לספק את התוצאות הגולמיות הגבוהות ביותר. הוא בעיקר הצליח "לחזק את החלשים" והצליח ליצור מסווגים יותר טובים כאשר היו שיטות ייצוג פחות מוצלחות כמו – fastText או bag of words.

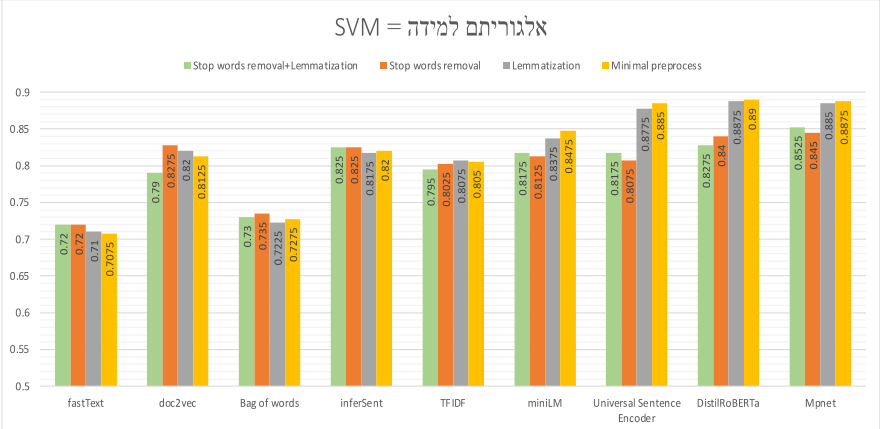
*אלגוריתם SVM הצליח לספק את התוצאות הגולמיות הגבוהות ביותר בשילוב התכונות שנוצרו ע"י: MPnet,DistillRoBERTA ו-Universal Sentence Encoder ובשילוב עם עיבוד מקדים מסוג minimal preprocess. כנראה המודלים אלו ייצרו ייצוג וקטורי שהיטיב מאוד עם היכולת של SVM למצוא מסווג אופטימלי. השילוב המוצלח ביותר היה minimal preprocess+DistillRoBERTa+SVM , כאשר נמצא מסווג עם אחוז דיוק של 89%.

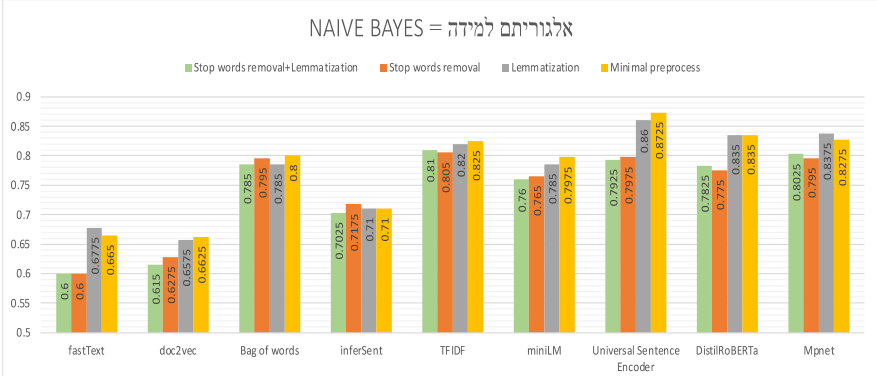
*קיומו של קשר משולש חזק בין אלגוריתם למידה Naïve Bayes לתכונות מסוג Universal Sentence encoder ושיטות עיבוד מקדים מסוג Minimal preprocess. למרות שהאלגוריתם מהסוג הגאוסיאני לרוב לא יצר מסווגים מוצלחים כפי שאפשר לראות, השילוב שלו עם Univrsal sentence encoder ועיבוד מקדים מינמלי היה מוצלח. המסווג הטוב ביותר שנוצר היה בעל אחוז דיוק של 87.25%.

תרשים 7 : אחוז דיוק המודל כפונקציה של שיטת עיבוד מקדים, סוג התכונות המייצגות את הטקסט וסוג אלגוריתם למידה









מממצאי ניסוי מספר 1 עולות המסקנות המרכזיות הבאות:

- תכונות המיוצרות באמצעות מודל fastText אינן מייצגות היטב את הפרסומים ממאגר הנתונים שלנו ולכן החלטנו לא להמשיך להשתמש במודל זה בניסויים הבאים.
- שיטות עיבוד מקדים מסוג stop words removal וstop words removal+Lemmatization גם כן היו פחות מוצלחות בעיבוד מקדים של המסמכים במאגר הנתונים שלנו. אומנם היו מס' מועט של מודלים ליצירת תכונות שבהם הסרת stop words שיפרה במעט את התוצאות והניבה תכונות טובות לייצוג טקסט, אך גם במקרים אלו השיפור לא היה דטרמינסטי והייתה תלות באלגוריתם הלמידה שיצר את המסווג הסופי. לכן החלטנו לא להמשיך עם שיטות עיבוד מקדים אלו בניסויים הבאים.
- משפחת אלגוריתמי הלמידה מסוג Naïve Bayes יצרה מסווגים פחות מוצלחים בממוצע, בעיקר הGaussian Naïve Bayes ולכן החלטנו לא להמשיך עם משפחה זו לניסויים הבאים.

עוד מסקנות כלליות שהסקנו הן כי המודלים שייצרו את התכונות הטובות ביותר עבור המסמכים שלנו היו: MPNet, DistillRoBERTa, וUniversal Sentence Encoder. שיטות אלו הן ממשפחת "התכונות הסמנטיות" והן מקודדות מסמכים לוקטורים באמצעות יצירת שיכונים למסמכים שלנו. באופן כללי, ראינו כי ברוב המקרים תכונות סמנטיות התעלו על תכונות לקסיקליות. מכאן, הגענו למסקנה, שהמודלים השונים המייצרים תכונות סמנטיות מצליחים לשמר את המשמעות של כל מסמך בצורה טובה יותר.

שיטות העיבוד המקדים המיטביות עבור המסמכים שלנו היו minimal preprocess וLemmatization ואלגוריתם הלמידה שייצר את המסווגים המוצלחים ביותר היה SVM , כאשר גם MLP ו Random forest הצליח לספקו תוצאות טובות יחסית במקרים רבים.

ניסוי מס׳ 2

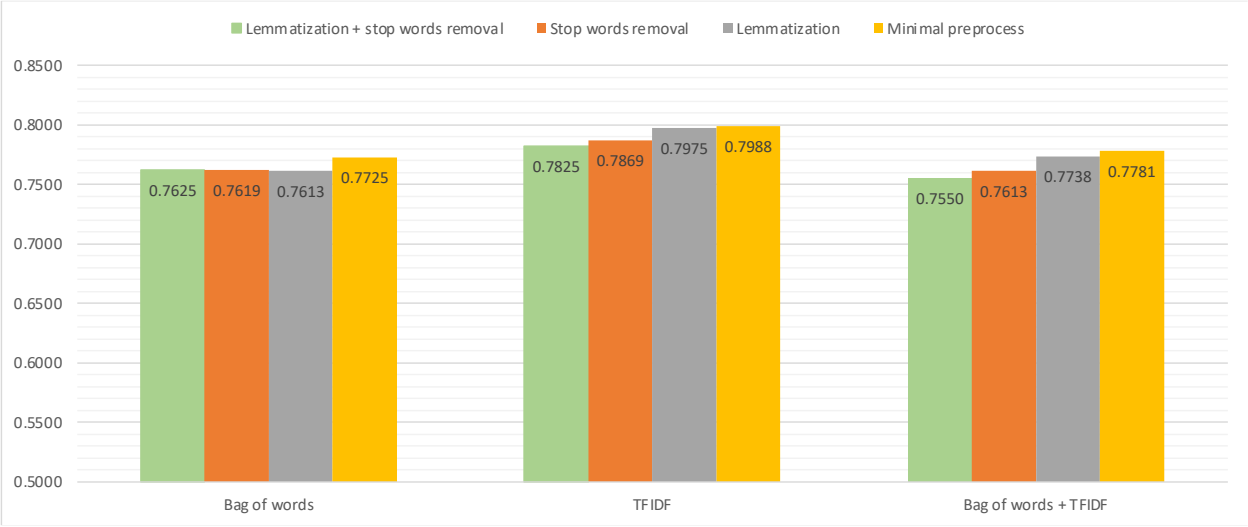
ממצאי ניסוי מס׳ 1 מעידים כי תכונות סמנטיות עדיפות על תכונות לקסיקליות. בניסוי מס׳ 2 החלטנו לבחון האם התכונות הלקסיקליות טומנות בחובן מידע אחר מהתכונות הסמנטיות, כך שייצוג משולב של תכונות לקסיקליות וסמנטיות ייתן תוצאות טובות יותר מאשר ייצוג המסמך באמצעות תכונות סמנטיות בלבד.

לכן, מטרתו העיקרית של ניסוי מס׳ 2 הייתה לבחון האם ייצוג וקטורי המורכב משרשור של תכונות לקסיקליות ותכונות סמנטיות ישפר את אחוז הדיוק של המודל, בהשוואה לייצוג וקטורי המכיל תכונות סמנטיות בלבד.

נבחנו שלושה סוגים שונים של תכונות לקסיליות במסגרת הניסוי-

1) וקטור תכונות לקסיקליות שנוצר באמצעות מודל Bag Of Words
2) וקטור תכונות לקסיקליות שנוצר באמצעות מודל TFIDF
3) וקטור תכונות משולב -וקטור זה נוצר ע״י שרשור של 1+2 כלומר שיטה משולבת של Bag of Words+TFIDF.

כשלב מקדים לניסוי, בוצעו מס׳ ניתוחים מקדימים כדי לאתר את התנאים האופטימליים עבור יצירת תכונות לקסיקליות. החלטנו שעדיף לעבד את התכונות הסמנטיות והתכונות הלקסיקליות בנפרד , מאחר וייתכן כי שיטת עיבוד מקדים אחת עשויה ליצור תכונות לקסיקליות טובות יותר ושיטת עיבוד מקדים אחרת עשויה ליצור תכונות סמנטיות טובות יותר. בדומה למה שביצענו בניסוי מספר 1, בדקנו מס׳ שיטות עיבוד מקדים וארבעה סוגים של אלגוריתמי למידה בשילוב עם שלושת הסוגים השונים של התכונות הלקסיקליות. הממצאים מוצגים בתרשים מספר 8.



תרשים 8 : אחוז הדיוק הממוצע של המודל כפונקציה של שיטת העיבוד המקדים ושל סוג התכונות הלקסקיליות המייצגות את הטקסט. כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור שיטת עיבוד מסוימת x סוג תכונות מסוים והממוצע חושב על פני ארבעת אלגוריתמי הלמידה השונים.

ניתן לראות כי שילוב בין שיטת עיבוד מקדים מינימלי לכל סוגי התכונות הלקסיקליות הביא ליצירת מודלים בעלי אחוז דיוק ממוצע גבוה יותר. בשל כך החלטנו ששיטת עיבוד מקדים מינימלית למסמכים עדיפה בשילוב עם תכונות לקסיקליות.

לאחר הניסוי הראשוני, עברנו לבצע את הניסוי המרכזי. מאחר ובסופו של דבר מטרתנו בפרוייקט עצמו היא למצוא את השילוב המיטבי בין עיבוד מקדים, התכונות המיוצרות מהטקסט ואלגוריתם למידה כך שיווצר מסווג בעל אחוז הדיוק גבוה ביותר , החלטנו שלא לבודד את המשתנה המרכזי אלא המשכנו את הניסויים עם שתי שיטות לעיבוד מקדים ו3 אלגוריתמי הלמידה. שיערנו כי ייתכן ונראה שאלגוריתם למידה מסוים יצליח להסיק מידע חדש מהוקטורים המשורשרים ולייצר מסווגים מדויק יותר בעוד שעבור אלגוריתם למידה אחר תיפגם היכולת להסיק מידע והוא יצור מסווגים פחות מוצלחים. זאת ועוד, לא הצלחנו לקבוע מניסוי מס׳ 1 איזו שיטת עיבוד מקדים עדיפה בשילוב עם תכונות סמנטיות – Lemmatization או Minimal prerocess . לכן

העדפנו להריץ את ניסוי מס׳ 2 פעמיים כאשר אנחנו מעבדים את המסמכים שלנו בשתי השיטות בטרם יצירת התכונות הסמנטיות.

<u>אופן ביצוע הניסוי:</u>

תחילה כל אחד מהמסכים עבר עיבוד מקדים תוך בחירה של אחת מהשיטות- Minimal preprocess, Lemmatization.
לאחר מכן עבור כל אחד מהמסמכים נבחרה אחד מחמשת המודלים הבאים המייצרים וקטור תכונות סמנטיות: doc2vec, inferSent, MiniLM, MPNet, DistilRoBERTa, Universal Sentence Endoder.
נכנה את וקטור התכונות הסמנטיות הנוצר באמצעות המודל הנבחר בשם הוקטור המקורי.
כעת החל הניסוי המרכזי- נבחר סוג אחד של תכונות לקסיקליות ששורשר לוקטור התכונות הסמנטיות:

1. "הסוג הריק" -אי שרשור של וקטור נוסף לוקטור המקורי- זוהי למעשה קבוצת הביקורת.
2. שרשור של וקטור מסוג Bag of words המייצג את אותו המסמך לוקטור המקורי.
3. שרשור של וקטור מסוג TFIDF המייצג את אותו המסמך לוקטור המקורי.
4. שרשור של וקטור משולב Bag of words+TFIDF המייצג את אותו המסמך לוקטור המקורי.

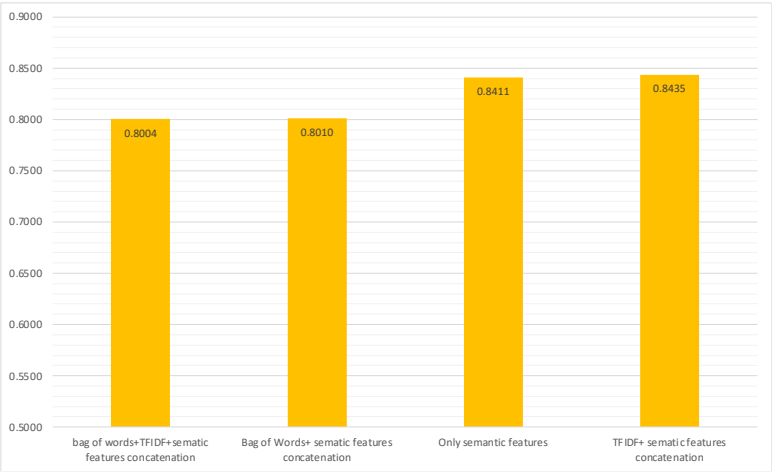
כתוצאה מפעולת השרשור נוצר וקטור אחד המייצג את המסמך והוא מורכב מ2 תתי וקטורים (באמצעות הדבקה של הוקטור המקורי לוקטור התכונות הלקסיקליות בקצהו כך שנוצר וקטור בעל יותר מימדים).
חשוב לציין כי הבחירה של שיטת העיבוד המקדים משפיעה רק בעת יצירת תכונות סמנטיות. התכונות הלקסיקליות נוצרות בנפרד, בשילוב עם עיבוד מקדים מינימלי כפי שהוסבר קודם.

לאחר יצירת התכונות מהטקסט, נבחר אחד משלושת אלגוריתמי הלמידה: MLP, SVM, Random Forest.
אלגוריתם הלמידה אומן בכל פעם על 4 קבוצות דוגמאות במסגרת שיטת תוקף צולב כך שבכל שלב בניסוי נוצרו חמישה מסווגים שונים כתוצאה מהקומבינציה המשולשת ואחוז הדיוק של הקומבינציה נקבע כממוצע על חמשת המסווגים. במסגרת ניסוי זה נוסו כל הקומבינציות האפשריות של שיטת עיבוד מקדים, סוגי תכונות- משולבות ולא משולבות, אלגוריתם למידה.

<u>תוצאות ניסוי מס׳ 2</u>

מאחר והמטרה המרכזית של הניסוי היא לבחון את ההשפעה של שרשור סוגים שונים של תכונות לקסיקליות לתכונות סמנטיות על אחוז הדיוק של המודל, נציג מס׳ גרפים שמציגים את התוצאות הממוצעות של הניסויים, כך שנוכל לבחון את השפעתו גורם זה. את התוצאות הגולמיות של הניסוי ניתן למצוא בנספחים.

<u>השפעת סוג התכונות הלקסיקליות בייצוגים משולבים על אחוז הדיוק הממוצע</u>

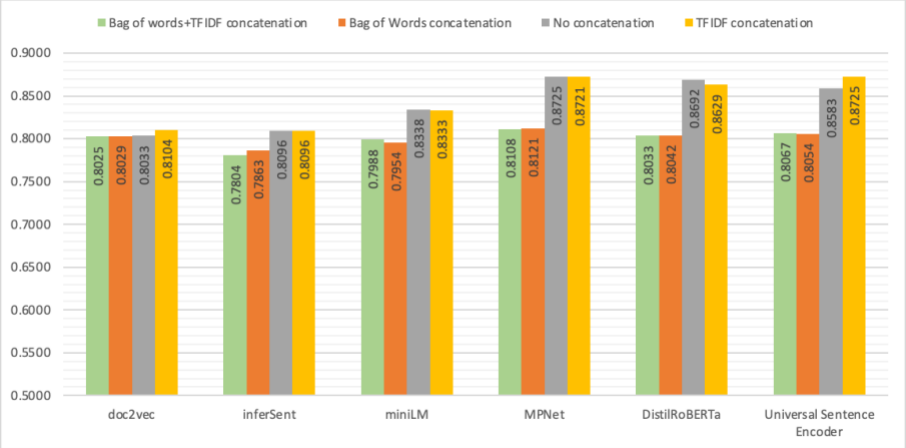


תרשים 9: אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות הלקסיקליות ששורשרו במסגרת ייצוגים משולבים.
** **Only semantic features**– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בתכונות סמנטיות בלבד כדי לייצג את הטקסט, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*
** **Bag of words+ semantic features concatenation**– מודלים משולבים של Bag of words+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את Bag of words, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*
** **TFIDF + semantic features concatenation** – מודלים משולבים של TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*
** **Bag of words+ semantic features concatenation** – מודלים משולבים של Bag of words+TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכילים את Bag of words+TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*

מתוצאות הניסוי ניתן להבחין כי מודלים אשר השתמשו בייצוג של תכונות סמנטיות + וקטורי TFIDF היו בעלי אחוז הדיוק הממוצע הגבוה ביותר. עם זאת בהשוואה לקבוצת הביקורת- שימוש בתכונות סמנטיות בלבד, השיפור באחוז הדיוק הממוצע הוא מועט ולכן לא ניתן להסיק על סמך גרף זה האם באמת הייצוג המשולב עדיף.

הבחנה נוספת היא שמודלים שהשתמשו בייצוגים המשולבים : תכונות סמנטיות + Bag of words ותכונות סמנטיות+ bag of words+TFIDF היו בעלי אחוזי דיוק ממוצע נמוכים במיוחד בהשוואה לקבוצת הביקורת. מכאן ניתן להסיק שייצוגים משולבים אלו אינם עדיפים.
סיבה אפשרית לכך שמודל משולב TFIDF עבד טוב יותר היא שציון הTFIDF מעריך עבור כל מילה עד כמה היא רלוונטית במאגר מסמכים ולכן טומן בחובו מידע חשוב יותר בהשוואה למודל Bag of words , שרק סופר את כמות הפעמים שכל מילה מופיעה. בנוסף, מאחר ומודל Bag of words מייצר תכונות שאינן מנורמלות ועשויות להכיל מספרים גדולים מאוד, ייתכן שהוא "הורס" את הייצוגים המבוזרים של התכונות הסמנטיות ולכן הוא פחות מתאים לשיטת השרשור.

<u>אינטראקציה בין סוג התכונות הסמנטיות לסוג התכונות הלקסיקליות</u>

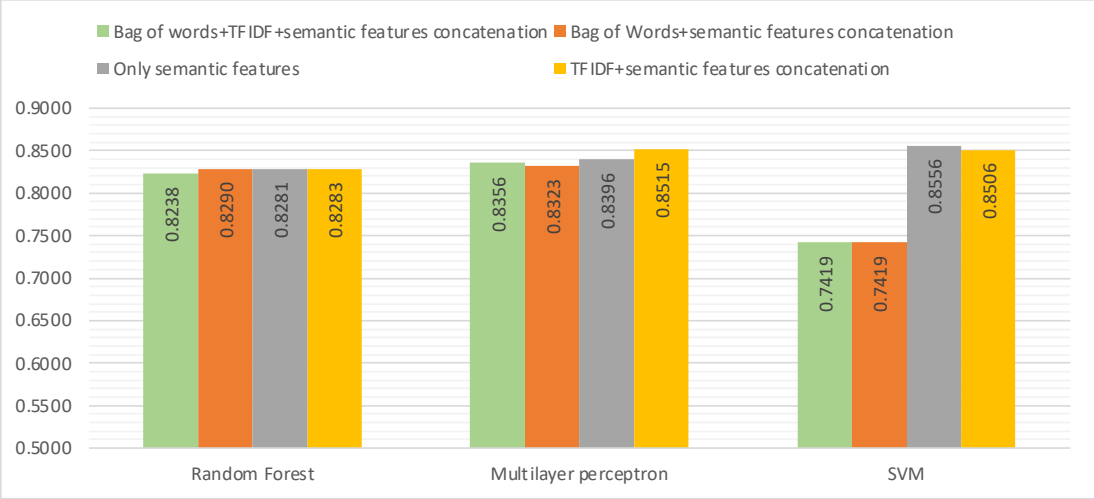


תרשים 10 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות הסמנטיות וסוג התכונות הלקסיקליות במסגרת ייצוגים משולבים.
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל סוג מסוים של תכונות סמנטיות x סוג מסוים של תכונות לקסיקליות והממוצע חושב על פני אלגוריתמי הלמידה השונים ושיטות העיבוד המקדים השונות. בנוסף עבור כל סוג של תכונה סמנטית חושב אחוז הדיוק הממוצע של שימוש רק בתכונות סמנטיות מסוג זה, ללא ייצוג משולב. גרפים אלו כונו בשם "No concatenation" מאחר ולא התבצע שרשור של תכונות לקסיקליות לסמנטיות.

מגרף זה ניתן לראות, בדומה לגרף הקודם, כי שרשור של תכונות לקסיקליות מסוג Bag of words ומסוג Bag of words+TFIDF לתכונות סמנטיות פוגם באחוז הדיוק הממוצע , בהשוואה לשימוש בתכונות סמנטיות בלבד. הפעם ניתן לראות, כי עבור אף סוג של תכונות סמנטיות, השרשור הזה אינו יעיל.

עוד ניתן לראות כי ייצוג משולב של תכונות סמנטיות ותכונות לקסיקליות מיטבי רק עבור תכונות סמנטיות מסוימות.
כך למשל, ניתן לראות כי שימוש בייצוג משולב של Universal Sentence Encoder+tfidf הביא לעליה בכ-2% באחוז הדיוק הממוצע בהשוואה לשימוש בייצוג ע"י תכונות סמנטיות שנוצרו ע"י Universal Sentence Encoder בלבד. בדומה ניתן לראות שיפור גם עבור ייצוג משולב של doc2vec+tfidf בהשוואה ל doc2vec בלבד. לעומת זאת, קיימים סוגים של תכונות סמנטיות שעבורם השילוב יחד עם תכונות לקסיקליות פגם באחוז הדיוק הממוצע. כך למשל, ניתן לראות שעבור תכונות סמנטיות שנוצרו ע"י DistilRoBERTa ו MPNet , שרשור של תכונות מסוג TFIDF הביאו לירידה מועטה באחוז הדיוק הממוצע.
עבור המודלים miniLM ו inferSent , אין הבדל בממוצע בין ייצוג משולב עם TFIDF לבין ייצוג ללא שרשור של וקטורים נוספים כלל.

אינטראקציה בין סוג התכונות הלקסיקליות בייצוגים משולבים לבין סוג אלגוריתם הלמידה



תרשים 12 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות הלקסיקליות ששורשרו במסגרת ייצוגים משולבים וסוג אלגוריתם הלמידה.

** **Only semantic features**– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בתכונות סמנטיות בלבד כדי לייצג את הטקסט, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

** **Bag of words+ semantic features concatenation** – מודלים משולבים של Bag of words+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את Bag of words, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

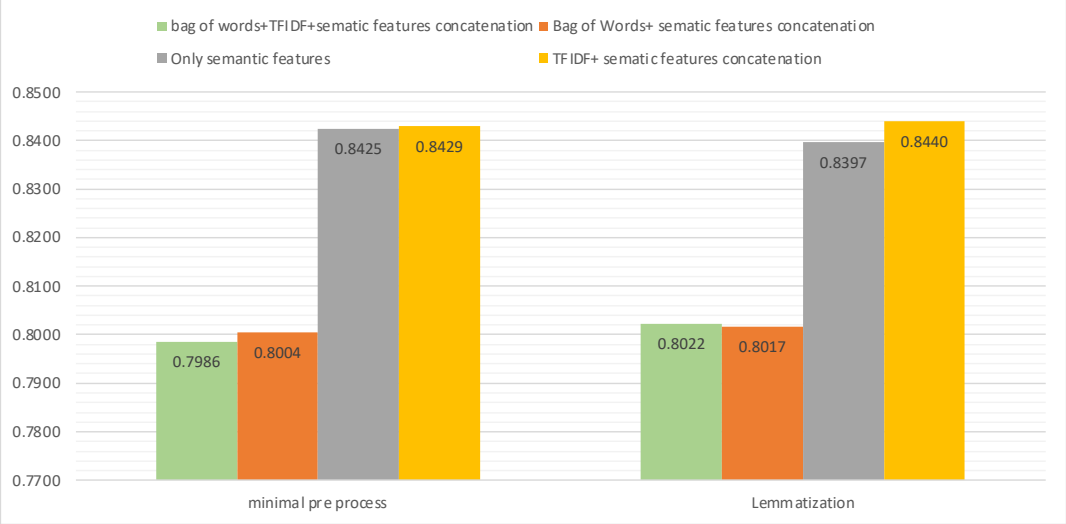
** **TFIDF + semantic features concatenation** – מודלים משולבים של TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

** **Bag of words+ semantic features concatenation** – מודלים משולבים של Bag of words+TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכילים את Bag of words+TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד והופרד בין אלגוריתמי למידה.*

מן התוצאות ניתן לראות כי כפי ששיערנו, היה שוני בין היכולת של אלגוריתמי הלמידה להפיק מידע מהסוגים השונים של התכונות הלקסיקליות בתוך הייצוגים המשולבים.
כאשר מתבוננים באלגוריתם Random Forest לדוגמה, ניתן לראות כי ההשפעה של הייצוגים המשולבים היא מועטה, בהשוואה לשימוש בתכונות סמנטיות בלבד. בנוסף, לא היו כמעט הבדלים בין סוגים שונים של תכונות לקסיקליות.
עבור אלגוריתם MLP, ניתן להבחין כי שימוש בייצוגים משולבים של תכונות סמנטיות+ TFIDF הביא לאחוז דיוק ממוצע גבוה ביותר- שיפור ממוצע של כ2% בהשוואה לייצוג באמצעות תכונות סמנטיות בלבד.
בנוסף, ייצוגים משולבים שכללו תכונות לקסיקליות מסוג Bag of words ו Bag of words+tfidf פגמו מעט באחוז הדיוק הממוצע , בהשוואה לשימוש בתכונות סמנטיות בלבד.

בניגוד לשני האלגוריתמים הקודמים, ניתן לראות כי עבור אלגוריתם SVM ,שימוש בייצוגים משולבים מסוג Bag of words ו Bag of words+tfidf פגמו מאד באחוז הדיוק הממוצע של המודלים שנוצרו. סיבה אפשרית לכך היא שמודל SVM רגיש מאוד לנרמול הנתונים, ומודל Bag of words מייצר נתונים שאינם מנורמלים. לכן ברגע שאנחנו משרשרים לוקטורים מנורמלים מהמשפחות הסמנטיות וקטורים לא מנורמלים , הדבר פוגע ביכולת של SVM לייצר מסווג נומרי שיודע להפריד בין הדוגמאות בצורה טובה. תמיכה בטענה זו היא ששימוש בייצוגים משולבים עם וקטורים מסוג TFIDF פגמה באחוז הדיוק הרבה פחות בהשוואה לייצוגים שכללו את Bag of words.
זאת ועוד, ניתן לראות כי עבור מודל SVM, ייצוג באמצעות תכונות סמנטיות בלבד הוא המיטבי ביותר , וכל הייצוגים המשולבים פגעו באחוז הדיוק הממוצע.

אינטראקציה בין סוג התכונות הלקסיקליות בייצוגים משולבים לבין שיטת עיבוד מקדים



תרשים 12 : אחוז הדיוק ה ממוצע של המודל כפונקציה של סוג התכונות הלקסיקליות ששורשרו במסגרת ייצוגים משולבים וסוג שיטת עיבוד מקדים שעברו התכונות הסמנטיות.

** **Only semantic features**– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בתכונות סמנטיות בלבד כדי לייצג את הטקסט, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות ועל פני כל אלגוריתמי הלמידה והייתה הפרדה בין שיטות עיבוד שונות.*

** **Bag of words+ semantic features concatenation**– מודלים משולבים של Bag of words+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בייצוג משולב המכיל את Bag of words, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, ועל פני כל אלגוריתמי הלמידה והייתה הפרדה בין שיטות עיבוד שונות.*

** **TFIDF + semantic features concatenation** – מודלים משולבים של TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בייצוג משולב המכיל את TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות ועל פני כל אלגוריתמי הלמידה והייתה הפרדה בין שיטות עיבוד שונות.*

** **Bag of words+ semantic features concatenation** – מודלים משולבים של Bag of words+TFIDF+ סמנטיות. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בייצוג משולב המכילים את Bag of words+TFIDF, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות ואלגוריתמי למידה והייתה הפרדה בין שיטות עיבוד שונות.*

***

מן התוצאות ניתן לראות קיומו של קשר חלש בין שני המשתנים, שכן התוצאות שהתקבלו דומות כתוצאה משימוש בייצוגים משולבים עם תכונות לקסיקליות גם כאשר מפרידים בין שתי שיטות העיבוד המקדים. מה שכן ניתן להבחין בו שדווקא כאשר המסמכים עובדו באמצעות שיטת "Lemmatization" היה שיפור גדול יותר באחוז הדיוק הממוצע עבור שימוש בייצוג משולב של תכונות סמנטיות +TFIDF. ייתכן, כי חלק מהמידע הסמנטי של המשפט אובד כאשר מבצעים Lemmatization והייצוג המשולב עם TFIDF מסייע להשיב את המידע האבוד.

קומבינציות אופטימליות שנמצאו בניסוי מספר 2

מהתבוננות בתוצאות הגולמיות, מצאנו מס' מודלים שקיבלו תוצאות גבוהות בניסוי ה1 ושופרו עוד יותר בעקבות שרשור של וקטורי TFIDF :

- הקומבינציה של שיטת עיבוד מקדים מינימלי + תכונות שנוצרו ע"י MPNet+ אלגוריתם למידה MLP הוערכה עם אחוז דיוק של כ-86%. הקומבינציה של שיטת עיבוד מקדים מינמלי + ייצוג משולב של MPNet+TFIDF + אלגוריתם למידה הוערכה עם אחוז דיוק של כ-89%. היה במקרה זהו שיפור משמעותי של 3% שקרה כלל הנראה בעקבות הוספת וקטורי TFIDF לייצוג של המסמכים.
- הקומבינציה של שיטת עיבוד מקדים מסוג "Lemmatization" + תכוות שנוצרו ע"י Universal Sentence Encoder+ אלגוריתם למידה SVM הוערכה עם אחוז דיוק של כ-87.75%. הקומבינציה של שיטת עיבוד מקדים מסוג "Lemmatization" + ייצוג משולב של Universal Sentence Encoder+TFIDF+ אלגוריתם למידה SVM הוערכה עם אחוז דיוק של כ-89.75%. היה במקרה זה שיפור של 2% שקרה כלל הנראה בעקבות הוספת וקטורי TFIDF לייצוג של המסמכים. <u>תוצאות אלו גם היו התוצאות הגבוהות ביותר שקיבלנו עבור מסווג עד כה.</u>

מממצאי ניסוי מספר 2 עולות המסקנות המרכזיות הבאות:

- ייצוגים משולבים של תכונות סמנטיות עם תכונות לקסילקיות מסוג Bag of words ומסוג Bag of words+TFIDF אינם ייצוגים מוצלחים עבור המסמכים במאגר הנתונים שלנו מאחר והן פגמו ביכולות הסיווג של המודלים שנוצרו , בהשוואה לשימוש בתכונות סמנטיות בלבד. לכן החלטנו שלא להמשיך עם ייצוגים משולבים אלו לניסוי מס' 3.
- ייצוגים משולבים של תכונות סמנטיות עם תכונות לקסיקליות מסוג TFIDF דווקא עשויים להניב שיפור ביכולות הסיווג של חלק מהמודלים. התוצאה האופטימלית שהתקבלה התקרבה לכ90% (89.75%).
- היו גם מעט מקרים בהן ייצוגים משולבים עם תכונות לקסיקליות מסוג TFIDF פגמו באחוז הדיוק הממוצע, בהשוואה לשימוש תכונות סמנטיות בלבד. לפיכך, החלטנו שבניסוי מס' 3 נבחן גם ייצוגים משולבים של תכונות סמנטיות+לקסיקליות מסוג TFIDF וגם תכונות סמנטיות ללא כל תוספת.

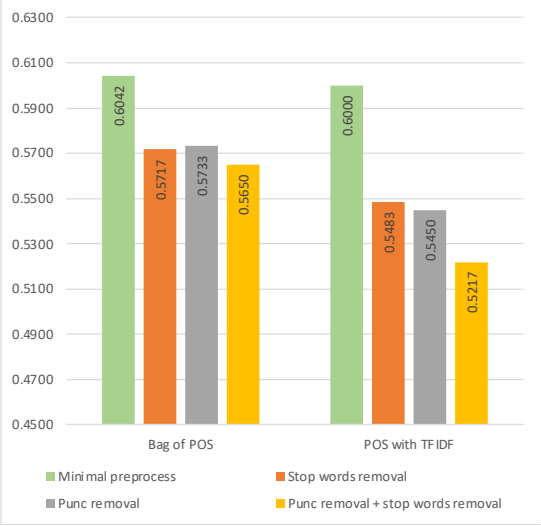
ניסוי מס׳ 3

בניסוי מספר 3 בחנו האם המבנה התחבירי של המסמכים השונים טומן בחובו מידע שעשוי לשפר את יכולות הסיווג של המודל שלנו.
במסגרת ניסוי זה, החלטנו לנסות לנצל את המידע התחבירי שניתן להפיק מחלקי הדיבר של המסמכים (part of speech) וליצור ממנו וקטור תכונות תחביריות.
מטרתו העיקרית של ניסוי מס׳ 2 הייתה לבחון האם ייצוג וקטורי המורכב משרשור של תכונות תחביריות לתכונות לקסיליות ו/או תכונות סמנטיות ישפר את אחוז הדיוק של המודל, בהשוואה לייצוג וקטורי המכיל את התכונות המקוריות בלבד.
בעיקר עניין אותנו לנסות לשפר ייצוגים המורכבים מתכונות סמנטיות בלבד או ייצוגים משולבים של תכונות סמנטיות + לקסיקליות(שהם בעלי אחוז דיוק גבוה יותר). עם זאת, מאחר ורצינו גם לדעת האם יש יתרון כלשהו בשימוש בתכונות תחביריות, החלטנו גם לבדוק ייצוגים משולבים של תכונות לקסיקליות ותחביריות ולבדוק האם ניתן לשפר ייצוג המורכב מתכונות לקסיקליות בלבד.

בשלב ראשוני של הניסוי, עבור כל מסמך במאגר הנתונים- ביצענו תיוג חלקי דיבר (part of speech tagging) ועל בסיס ניתוח זה יצרנו תכונות מהמסמך.
משימת תיוג חלקי הדיבר (part of speech tagging or in short POS tagging) היא משימה בעולם עיבוד השפה הטבעית, בה מאמנים מודל לתייג כל מילה בטקסט לחלק הדיבר שלה (למשל פועל, שם עצם, תואר ועוד). אנחנו השתמשנו במסגרת הפרויקט במודל מאומן מראש שביצע את התיוג, שפורסם ע״י ספריית Spacy.
כעת, היינו צריכות להחליט אילו וקטורים נייצר על בסיס ניתוח תחבירי זה.
החלטנו ליצור שני וקטורים מבוססי ספירת מופעים אותם נבחן במסגרת הניסוי:

- **"Bag of POS"**- שיטה זו הוצעה במאמר [41]. במסגרת שיטה זו עבור כל מסמך- החלפנו כל מילה בתיוג חלק הדיבר שלה. לאחר מכן הכנסנו את כל המסמכים למודל "Bag of words" שייצר עבור כל מסמך וקטור ספירת מופעים לחלקי הדיבר שלו (כלומר ספר כמה פעלים, שמות עצם תארים וכו׳ היו בכל מסמך). השתמשנו במימוש של sklearn לBag of words, אך העברנו אליו רק את חלקי הדיבר של כל מסמך.
- **"POS with TFIDF"**- במסגרת שיטה זו עבור כל מסמך- החלפנו כל מילה בתיוג חלק הדיבר שלה. לאחר מכן הכנסנו את כל המסמכים למודל "TFIDF" שייצר עבור כל מסמך וקטור שבו עבור כל חלק דיבר חושב ציון הTFIDF שלו, בדיוק כמו מודל TFIDF שפועל על מילים. ההשראה לשיטה זו התקבלה מהשיטה הקודמת- חשבנו לנסות גם את מודל הTFIDF כיוון שניתן לייצר את הוקטורים באותה קלות כמו עם שיטת Bag of words. השתמשנו במימוש של sklearn לTFIDF, אך העברנו אליו רק את חלקי הדיבר של כל מסמך.

בשלב הבא רצינו לבחון איזה עיבוד מקדים יהיה מיטיבי בשילוב עם התכונות התחביריות שלנו , כפי שביצענו בניסוי מס׳ 2.
בדומה לניסוי 2 , גם כאן ערכנו בדיקה מקדימה לאיתור התנאים האופטימליים עבור יצירת התכונות התחביריות. במסגרת הבדיקה, ניסינו את שיטות העיבוד המקדים הבאות: Minimal preprocess, stop words removal ושתי שיטות נוספות- punctuation removal ושיטה משולבת stop words removal punctuation removal. הסיבה שלא בחנו את שיטת הLemmatization, מאחר והיא פחות רלוונטית לניתוח תחבירי שבו גם כך לא מסתכלים על המילה עצמה או צורת השורש שלה, אלא מסתכלים על חלקי דיבר. לעומת זאת, ראינו כי בניתוח תחבירי גם לסימני פיסוק יש חלק דיבר מסוג סימן פיסוק ולכן תהינו האם להסרת סימני הפיסוק עשויה להיות השפעה ביצירת התכונות התחביריות. את הממצאים ניתן לראות בתרשים 13.



תרשים 13: אחוז הדיוק הממוצע של המודל כפונקציה של שיטת העיבוד המקדים ושל סוג התכונות התחביריות המייצגות את הטקסט. כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל עבור שיטת עיבוד מסוימת x סוג תכונות מסוים והממוצע חושב על פני 3 אלגוריתמי הלמידה השונים: SVM,MLP,Ranom Forest.

ניתן לראות כי שילוב בין שיטת עיבוד מקדים מינימלי לכל סוגי התכונות התחביריות הביא ליצירת מודלים בעלי אחוז דיוק ממוצע גבוה יותר. בשל כך החלטנו ששיטת עיבוד מקדים מינימלית למסמכים עדיפה בשילוב עם תכונות לקסיקליות.
הבחנה נוספת היא ששימוש בתכונות תחביריות בלבד כייצוג עבור המסמכים השונים אינו מוצלח, ומוביל ליצירת מודלים בעלי אחוז דיוק ממוצע נמוך יותר. תוצאות אלו אינן מפתיעות שכן התכונות התחביריות שיצרנו מתעלמות לגמרי מהמשמעות הסמנטית של המסמכים ולא מתייחסות כלל למילים שבמסמכים אלא רק לחלקי הדיבר.

לאחר הבדיקה המקדימה עברנו לבצע את הניסוי המרכזי. בדומה לניסוי מס' 2, גם פה לא בודדנו את המשתנה המרכזי שלנו בניסוי - השפעת שרשור של סוגים שונים של תכונות תחביריות לתכונות סמנטיות ו/או לקסיקליות על אחוז דיוק המודל, בהשוואה לשימוש בתכונות המקוריות בלבד. הסיבות לכך דומות לסיבות שתיארנו בניסוי מס' 2.

<u>אופן ביצוע הניסוי:</u>

תחילה כל אחד מהמסכים עבר עיבוד מקדים תוך בחירה של אחת מהשיטות- Minimal preprocess, Lemmatization.
לאחר מכן נבחר ייצוג למסמך באמצעות יצירת תכונות לקסיליות ו/או סמנטיות:

- ייצוג באמצעות תכונות לקסיקליות בלבד- נבחנו המודלים Bag of Words וTFIDF.
- ייצוג באמצעות תכונות סמנטיות בלבד- נבחנו המודלים הבאים: doc2vec, inferSent, MiniLM, MPNet, DistilRoBERTa, Universal Sentence Endoder.
- ייצוג משולב של תכונות לקסיקליות+תכונות סמנטיות- בעקבות ממצאי ניסוי מס' 2 החלטנו לבחון ייצוגים משולבים של תכונות סמנטיות ולקסיליות מסוג TFIDF. הייצוגים המשולבים שנבחנו היו: doc2vec+TFIDF,inferSent+TFIDF,MiniLM+TFIDF, MPNet+TFIDF DistilRoBERTa+TFIDF, Universal Sentenc Encoder+TFIDF.

נכנה את וקטור התכונות שנוצר לאחר בחירת הייצוג בשם הוקטור המקורי.

כעת החל הניסוי המרכזי- נבחר סוג אחד של תכונות תחביריות ששורשר לוקטור המקורי:

1. "הסוג הריק"- אי שרשור של וקטור נוסף לוקטור שכבר נוצר- זוהי למעשה קבוצת הביקורת.
2. שרשור של וקטור מסוג Bag of POS המייצג את אותו המסמך לוקטור המקורי.
3. שרשור של וקטור מסוג POS with TFIDF המייצג את אותו המסמך לוקטור המקורי.

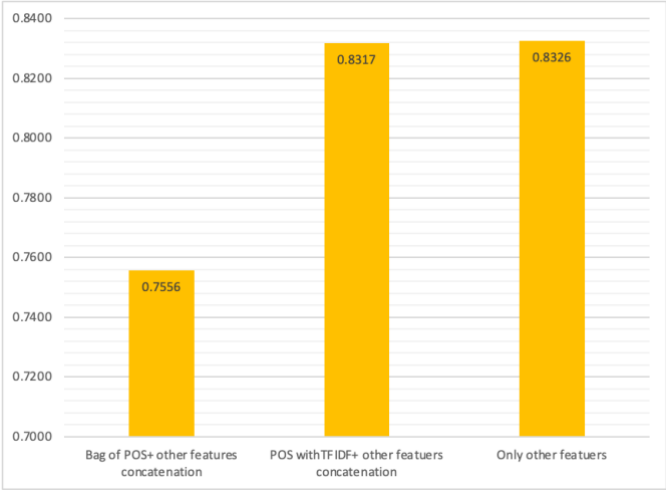
כתוצאה מפעולת השרשור נוצר וקטור אחד המייצג את המסמך והוא מורכב מ2 תתי וקטורים.
חשוב לציין שהעיבוד המקדים במקרה זה ישפיע רק על החלק המקורי בכל וקטור, מאחר והוחלט שעבור יצירת התכונות התחביריות נשתמש בעיבוד מקדים מינימלי.

לאחר יצירת התכונות מהטקסט, נבחר אחד משלושת אלגוריתמי הלמידה : MLP, SVM, Random Forest.
אלגוריתם הלמידה אומן בכך פעם על 4 קבוצות דוגמאות במסגרת שיטת תוקף צולב כך שבכל שלב בניסוי נוצרו חמישה מסווגים שונים כתוצאה מהקומבינציה המשולשת ואחוז הדיוק של הקומבינציה נקבע כממוצע על חמשת המסווגים. במסגרת ניסוי זה נוסו כל הקומבינציות האפשריות של שיטת עיבוד מקדים, סוגי תכונות- משולבות ולא משולבות, אלגוריתם למידה.

תוצאות ניסוי מס' 3

מאחר והמטרה המרכזית של הניסוי היא לבחון את השפעת שרשור וקטורי תכונות תחביריות על אחוז הדיוק של המודל, נציג מס' גרפים שמציגים את התוצאות הממוצעות של הניסויים, כך שנוכל לבחון את השפעתו גורם זה.
(את התוצאות הגולמיות של הניסוי ניתן למצוא בקבצים המצורפים להגשה).

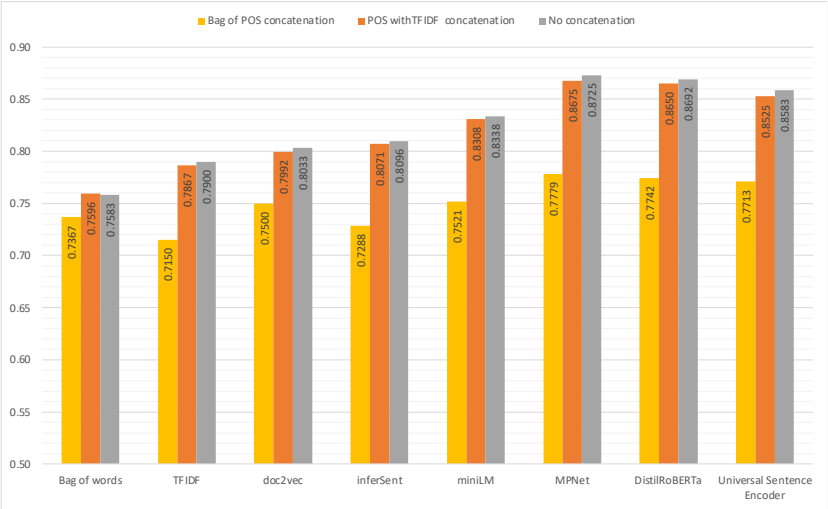
השפעת סוג התכונות התחביריות בייצוגים משולבים על אחוז הדיוק הממוצע



תרשים 14 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות התחביריות ששורשרו במסגרת ייצוגים משולבים.
Only other features *– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע על כל המודלים שהשתמשו בתכונות סמנטיות ו/או לקסיקליות כדי לייצג את הטקסט , כאשר הממוצע על פני כל סוגי התכונות הסמנטיות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*
Bag of POS+ other features concatenation *– מודלים משולבים של Bag of POS+ סמנטיות ו/או לקסיקליות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את Bag of POS, כאשר הממוצע על פני כל סוגי התכונות הסמנטיות ו/או הלקסיקליות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*
* ***POS with TFIDF + other features concatenation*** *– מודלים משולבים של POS with TFIDF + סמנטיות ו/או לקסיקליות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את POS with TFIDF , כאשר הממוצע על פני כל סוגי התכונות הסמנטיות ו/או לקסיקליות, שיטות לעיבוד מקדים ואלגוריתמי למידה.*

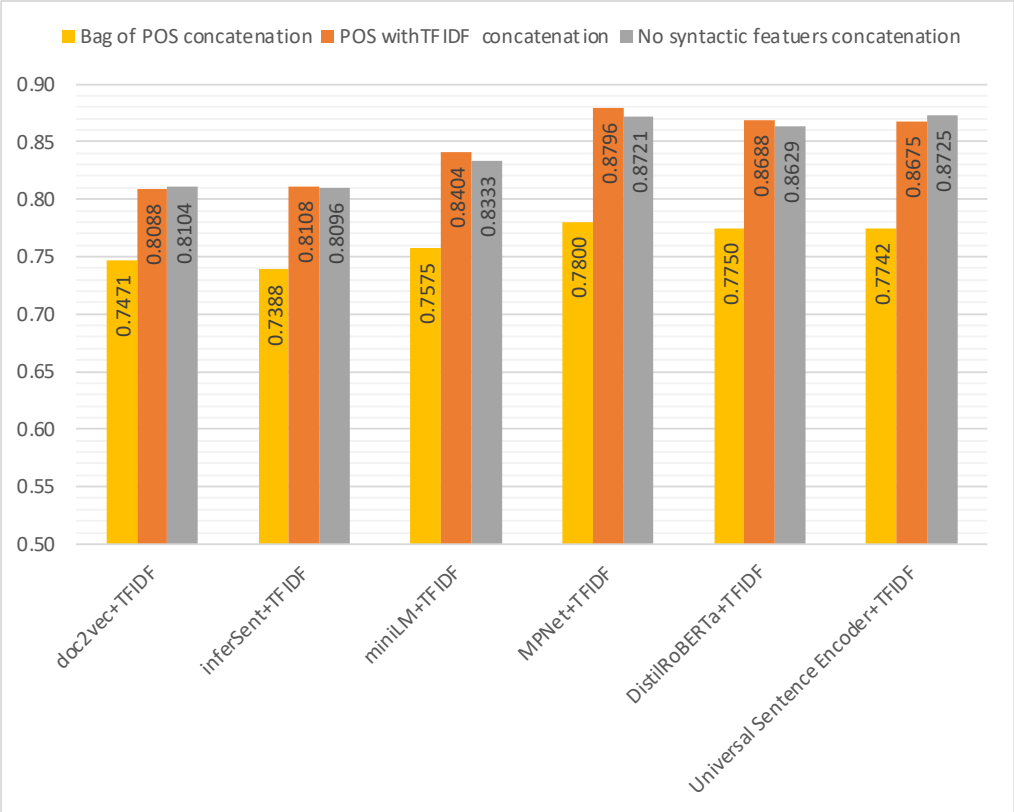
מתוצאות הניסוי ניתן להבחין כי המודלים אשר השתמשו בייצוג של תכונות סמנטיות ו/או לקסיקליות בלבד היו בעלי אחוז הדיוק הגבוה ביותר. כמו כן, המודלים אשר השתמשו בייצוג של תכונות סמנטיות ו/או לקסיקליות עם POS with TFIDF גם כן הגיעו לאחוז דיוק די גבוה.
לעומת זאת שילוב של תכונות סמנטיות ו/או לקסיקליות יחד עם תכונות מסוג Bag of POS הוביל למודלים בעלי אחוז דיוק נמוך למדי לעומת שני האחרים. מכאן נסיק שייצוגים משולבים אלו אינם עדיפים.
סיבה אפשרית לכך היא ש POS with TFIDF מעריך עבור כל חלק דיבר עד כמה הוא רלוונטי בהשוואה לשאר חלקי הדיבר והוא יכול גם לאתר חלקי דיבר יותר משמעותיים שעשויים להוות אינדיקציה לזיהוי פרסום קריטי.
לעומת זאת, Bag of POS מייצר ווקטור אשר סופר את מספרי חלקי הדיבר המופיעים במסמך, ספירה שיכולה להיות פחות רלוונטית מאחר ויש מעט סוגים של חלקי דיבר , לכן הם ככל הנראה חוזרים על עצמם פעמים רבות.
בנוסף, ייתכן שבדומה לניסוי 2, מודל Bag of POS אינו מתאים לייצוג משולב מאחר שאינו מנורמל ועשוי לייצר וקטורים שמכילים מספרים גבוהים מאוד, ואילו רוב המודלים בהם השתמשנו מייצרים וקטורים מנורמלים.
הדבר יכול לייצר תופעה בה למספרים הגבוהים יש יותר השפעה מהמספרים הקטנים , וכך לייצר מסווגים פחות מדויקים.

אינטראקציה בין סוג התכונות התחבירית לבין סוג התכונות הסמנטיות והלקסיקליות בייצוגים משולבים



תרשים 15 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות התחביריות וסוג התכונות האחרות (לקסיליות או סמנטיות) במסגרת ייצוגים משולבים כפולים בלבד.
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל סוג מסוים של תכונות לקסיליות או סמנטיות x סוג מסוים של תכונות תחביריות והממוצע חושב על פני אלגוריתמי הלמידה השונים ושיטות העיבוד המקדים השונות. בנוסף עבור כל סוג של תכונות סמנטיות/לקסיליות חושב אחוז הדיוק הממוצע של שימוש רק בתכונות מסוג זה, ללא ייצוג משולב עם תכונות תחביריות . גרפים אלו כונו בשם "No concatenation" מאחר ולא התבצע שרשור של תכונות תחביריות לתכונות אחרות.

מגרף זה ניתן להסיק כי בדומה לגרף הקודם שרשור של תכונות סמנטיות/לקסיקליות עם Bag of POS פוגם באחוז הדיוק הממוצע. ניתן לראות גם כי גם שימוש בייצוג המשולב Bag of POS+ Bag of Words פגם באחוז הדיוק הממוצע בהשוואה לשימוש בBag of Words בלבד. מכאן ניתן להסיק, שללא קשר לנרמול, המודל Bag of POS כולל בתוכו מידע פחות רלוונטי בהשוואה למודל POS with TFIDF.
כמו כן ניתן להבחין כי אחוזי הדיוק הגבוהים ביותר התקבלו עבור מודלים בהם לא השתמשנו בייצוגים משולבים כלל אלא בתכונות סמנטיות או לקסיקליות בלבד(קבוצת הביקורת שלנו למעשה).ייצוגים משולבים של שילוב תכונות סמנטיות/לקסיקליות + POS with TFIDF הניבו תוצאות מעט יותר נמוכות (ירידה בכ- 0.5% באחוז הדיוק הממוצע).
לפיכך, אנו מסיקות שאין יתרון כלשהו בשימוש בייצוגים משולבים של תכונות לקסיליות+תחביריות או תכונות סמנטיות + תחביריות.



תרשים 16 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות התחביריות וסוג התכונות המשולבות (לקסיקליות+סמנטיות) במסגרת ייצוגים משולבים משולשים בלבד.
כל עמודה בגרף מייצגת את אחוז הדיוק הממוצע שהתקבל תכונות לקסיקליות מסוג TFIDF x סוג של תכונות סמנטיות x סוג מסוים של תכונות תחביריות והממוצע חושב על פני אלגוריתמי הלמידה השונים ושיטות העיבוד המקדים השונות. בנוסף עבור כל סוג של תכונות משולבות חושב אחוז הדיוק הממוצע של שימוש רק בתכונות מסוג זה, ללא ייצוג משולב עם תכונות תחביריות . גרפים אלו כונו בשם "No syntactic featuers concatenation" מאחר ולא התבצע שרשור של תכונות תחביריות לתכונות אחרות.

ניתן להבחין בגרף זה , בדומה לקודמיו, שייצוג משולב הכולל תכונות תחביריות מסוג Bag Of POS מוריד משמעותית את אחוז הדיוק הממוצע.
עם זאת , מגרף זה נראה כי יש יתרון בייצוג משולב הכולל 3 סוגי תכונות : סמנטיות + תחביריות מסוג POS WITH TFIDF + לקסיקליות מסוג TFIDF. ככל הנראה, יש אינטראקציה בין שלוש סוגי התכונות שמביאה למודלים עם אחוז דיוק גבוה יותר, כיוון שראינו בעמוד הקודם שייצוג משולב שכלל רק תכונות תחביריות + תכונות סמנטיות או לקסיקליות לא הניב תוצאות טובות יותר.
בנוסף, נשים לב שעבור ייצוגים משולבים של MPNet+TFIDF , DstilRoberta+TFIDF, ו- MiniLM+TFIDF , שרשור של וקטורי POS with TFIDF הביא לשיפור באחוז הדיוק הממוצע (עלייה של חצי אחוז עד אחוז), בהשוואה לשימוש בייצוג המקורי בלבד.

אינטראקציה בין סוג התכונות התחביריות ששורשרו לבין סוג אלגוריתם הלמידה

תרשים 17 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות התחביריות ששורשרו במסגרת ייצוגים משולבים וסוג אלגוריתם הלמידה. המושג "other features" מתאר תכונות סמנטיות/תכונות לקסיקליות ותכונות משולבות של לקסיקליות+סמנטיות.

** **Only other features**– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בתכונות סמנטיות ו/או תכונות לקסיקליות בלבד כדי לייצג את הטקסט, כאשר הממוצע על פני כל סוגי התכונות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

** **Bag of POS+ other features concatenation**– מודלים משולבים של Bag of words+ תכונות אחרות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את Bag of words, כאשר הממוצע על פני כל סוגי התכונות האחרות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

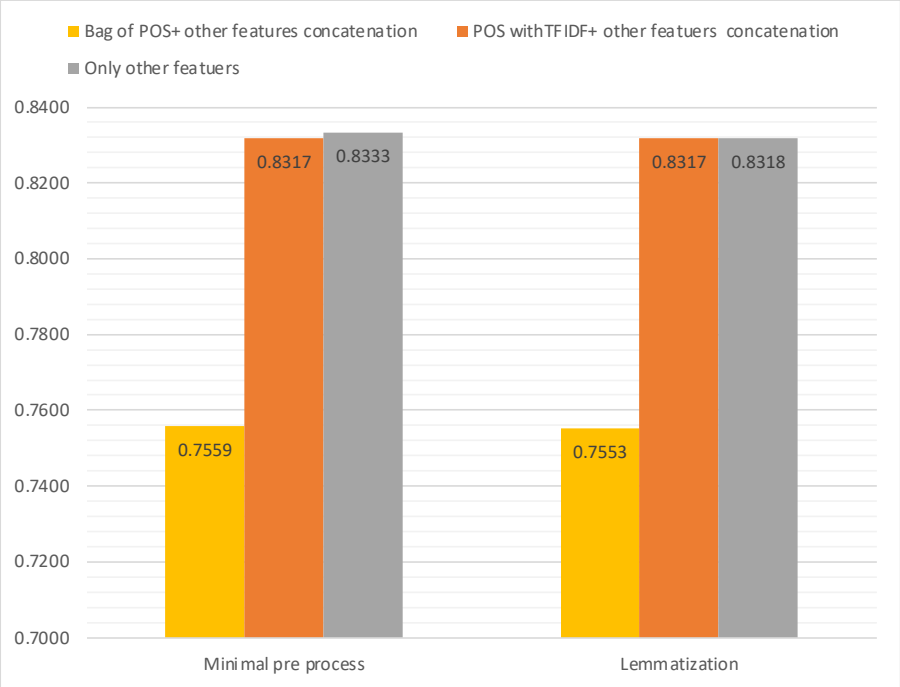
** **POS with TFIDF + other features concatenation** – מודלים משולבים של TFIDF+ תכונות אחרות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את TFIDF, כאשר הממוצע על פני כל סוגי התכונות האחרות, שיטות לעיבוד מקדים והופרד בין אלגוריתמי למידה.*

ראשית, בדומה לגרפים קודמים , ייצוג משולב הכולל תכונות תחביריות מסוג Bag of POS הביא לאחוז הדיוק הנמוך ביותר. ניתן לראות כי שימוש בייצוג המשולב הזה, בשילוב עם הרצת האלגוריתם SVM, פגם בתוצאות באופן משמעותי. ככל הנראה הירידה הייתה משמעותית מאוד בעקבות רגישותו של אלגוריתם SVM לחוסר נרמול.

כמו כן, נבחין כי בממוצע אחוז הדיוק הגבוה ביותר התקבל שילוב בין אלגוריתם הלמידה MLP לייצוג משולב הכולל תכונות תחביריות מסוג POS with TFIDF. בדומה לניסוי מס׳ 2, אפשר לראות כי האלגוריתם MLP מצליח להפיק מידע חדש מהתכונות התחביריות שמסייע לו למצוא מסווגים טובים יותר , בהשוואה לשימוש בתכונות סמנטיות ו/ או לקסיקליות בלבד.

לעומת אלגוריתם MLP, ניתן לראות כי שימוש בייצוגים משולבים הכוללים תכונות תחביריות פגם באחוזי הדיוק כאשר שילבנו תכונות אלו יחד עם אלגוריתמי הלמידה SVM, Random Forest.

אינטראקציה בין סוג התכונות התחביריות ששורשרו לבין סוג שיטת העיבוד המקדים:



תרשים 18 : אחוז הדיוק הממוצע של המודל כפונקציה של סוג התכונות התחביריות ששורשרו במסגרת ייצוגים משולבים וסוג שיטת העיבוד המקדים לתכונות האחרות. המושג "other features" מתאר תכונות סמנטיות/תכונות לקסיקליות ותכונות משולבות של לקסיקליות+סמנטיות.

** **Only other features**– קבוצת הביקורת. פה חושב אחוז הדיוק הממוצע של כל המודלים שהשתמשו בתכונות סמנטיות ו/או תכונות לקסיקליות בלבד כדי לייצג את הטקסט, כאשר הממוצע על פני כל סוגי התכונות, אלגוריתמי למידה והופרד בין שיטות עיבוד מקדים.*

** **Bag of POS+ other features concatenation**– מודלים משולבים של Bag of words+ תכונות אחרות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את Bag of words, כאשר הממוצע על פני כל סוגי התכונות האחרות, אלגוריתמי למידה והופרד בין שיטות עיבוד מקדים.*

** **POS with TFIDF + other features concatenation** – מודלים משולבים של TFIDF+ תכונות אחרות. פה חושב אחוז הדיוק הממוצע של כל המודלים בייצוג משולב המכיל את TFIDF, כאשר הממוצע על פני כל סוגי התכונות האחרות, אלגוריתמי למידה והופרד בין שיטות עיבוד מקדים.*

ניתן לראות כי קיבלנו תוצאות כמעט זהות עבור שיטת עיבוד מקדים מינימלי ושיטת עיבוד מקדים מסוג Lemmatization . מכאן ניתן להסיק שאין אינטראקציה חזקה בין שני המשתנים.
נוסף על כך, ניתן להבחין כי בשימוש עם שתי שיטות העיבוד המקדים, נפגם אחוז הדיוק הממוצע כאשר השתמשנו בייצוגים משולבים הכוללים תכונות תחביריות, ובעיקר בייצוגים משולבים הכוללים תכונות תחביריות מסוג Bag of POS.

<u>קומבינציות אופטימליות שנמצאו בניסוי מספר 3</u>
מהתבוננות בתוצאות הגולמיות, מצאנו מס' מודלים שקיבלו תוצאות גבוהות בניסויים הקודמים ושופרו עוד יותר בעקבות שרשור של תכונות תחביריות מסוג POS with TFIDF:

- הקומבינציה של שיטת עיבוד מקדים מסוג Lemmatization + תכונות שנוצרו ע"י MPNet + tfidf + אלגוריתם למידה MLP הוערכה עם דיוק של כ- 88%. הקומבינציה של Lemmatization + תכונות שנוצרו ע"י MPNet + tfidf + POS with TFIDF + אלגוריתם למידה MLP הוערכה עם דיוק של כ-89.25%. כלומר קיבלנו שיפור של 1.25% באחוז הדיוק.
- הקומבינציה של שיטת עיבוד מקדים מינימלית + ,תכונות שנוצרו ע"י DistilRoBERTa + TFIDF + אלגוריתם למידה Rando Forest הוערכה עם אחוז דיוק של כ-84.75%. הקומבינציה של שיטת עיבוד מקדים מינימלית + ,תכונו שנוצרו ע"י Roberta + TFIDF + POS with TFIDF +אלגוריתם למידה Rando Forest הוערכה עם אחוז דיוק של כ-87.75%. כלומר קיבלנו שיפור משמעותי של 3% באחוז הדיוק

מממצאי ניסוי מספר 3 עולות המסקנות המרכזיות הבאות:

- ייצוגים משולבים של תכונות סמנטיות ואו לקסיקליות עם תכונות תחביריות מסוג Bag of POS אינם ייצוגים מוצלחים עבור המסמכים במאגר הנתונים שלנו מאחר והן פגמו ביכולות הסיווג של המודלים שנוצרו.
- ייצוגים משולבים הכוללים תכונות תחביריות מסוג POS with TFIDF דווקא עשויים להניב שיפור במקרים מסוימים, בעיקר בשימוש בייצוגים הכוללים את שלוש סוגי התכונות. ראינו כי שימוש ייצוגים משולבים הכוללים תכונות לקסיקליות מסוג TFIDF, תכונות סמנטיות ותכונות תחביריות מסוג POS with TFIDF שיפרו את אחוז הדיוק במס' מקרים בהשוואה לשימוש בייצוגים הכוללים תכונות לקסיקליות מסוג TFIDF ותכונות סמנטיות בלבד.
- נראה כי יש קשר בין ניסוי מס' 2 לניסוי מס' 3- בשני הניסויים ראינו כי מודל TFIDF ייצר תכונות המייצגות את המסמכים שלנו בצורה טובה יותר ומשפרות את אחוז דיוק, במיוחד בהשוואה לייצוגים משולבים עם תכונות שיוצרו ע"י Bag of words. הבחנה נוספת שראינו בשני הניסויים היא שקומבינציה בין ייצוגים המשולבים לבין אלגוריתם MLP הייתה מיטבית, ונראה כי אלגוריתם MLP מצליח להסיק מידע נוסף מהתכונות המשורשרות ולמצוא מסווגים אופטימליים יותר לבעיה.

<u>קביעת המסווג האופטימלי לאור ממצאי הניסויים</u>

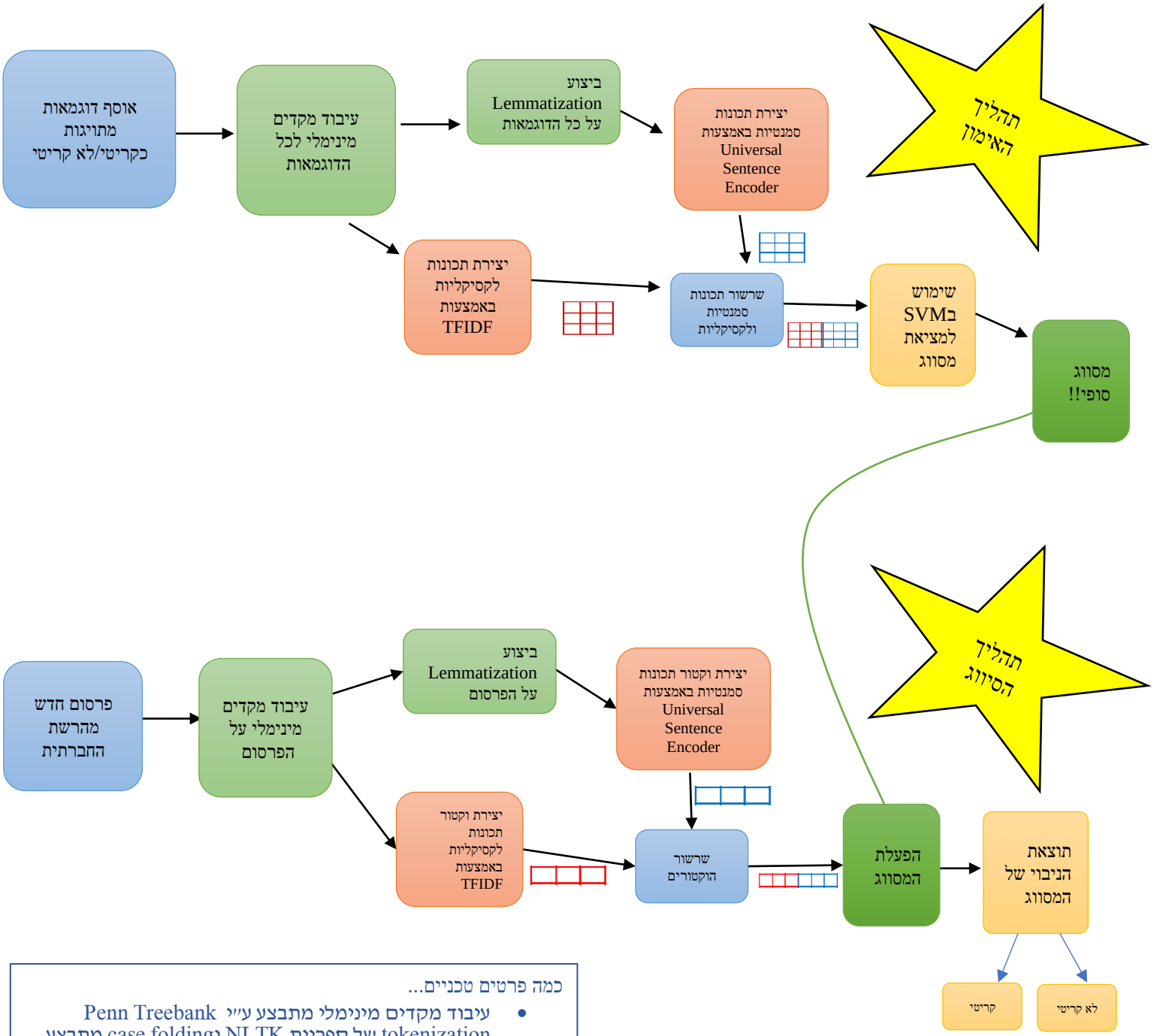
בתום כל הניסויים , הסתכלנו על כל התוצאות שהתקבלו לאורך כל שלושת הניסויים על מנת לאתר את המודל שהראה את אחוז הדיוק הגבוה ביותר. זאת במטרה לבחור אותו כמסווג אופטימלי עבור הבעיה שניסינו לפתור במסגרת הפרוייקט- סיווג של פרסומים מהרשת החברתית בנושא אלימות במשפחה לקריטיים/ לא קריטיים.
המודל הסופי שהחלטנו להשתמש בו הוא מודל ייצוג משולב של תכונות לקסיקליות + סמנטיות.
התכונות הלקסיקליות נוצרות ע"י מודל TFIDF , כאשר בטרם יצירתן כל מסמך עבר עיבוד מקדים מסוג Minimal preprocess.
התכונות הסמנטיות נוצרות ע"י מודל Universal Sentence Encoder , כאשר בטרם יצירתן כל מסמך עבר בנוסף עיבוד מקדים מסוג Lematization.
אלגוריתם הלמידה שנבחר ליצירת מסווג הוא SVM.

אחוז הדיוק שקיבל מודל זה המורכב משילוב של שיטות העיבוד המקדים, הייצוג המשולב ואלגוריתם הלמידה היה 89.75% (כאשר אחוז הדיוק חושב ע"י שיטת תוקף צולב שתיארנו קודם).

עבור יצירת המסווג הסופי, אימנו את אלגוריתם הלמידה SVM על כל 400 הדוגמאות במאגר הנתונים. המסווג זמין לשימוש בקובץ finalClassifier.py והוא מצפה לקבל כקלט מהמקלדת פרסום יחיד מהרשת החברתית כאשר הוא פולט עבורו האם מדובר בפרסום קריטי של אלימות למשפחה או שמדובר בפרסום שאינו קריטי.

בעמוד הבא נציג תרשים ובו מתואר כיצד פועל המסווג הסופי שבחרנו לפתרון.

תיאור המערכת ששימשה לפתרון:



כמה פרטים טכניים...

- עיבוד מקדים מינימלי מתבצע ע"י Penn Treebank tokenization של ספריית NLTK ו case folding מתבצע ע"י פונקציית ()lower של שפת python.
- Lemmatization מתבצע ע"י "WordNetLemmatizer".
- מודל TFIDF המייצר תכונות לקסיקליות מומש ע"י ספריית sklearn.
- מודל Universal Sentence Encoder אומן ע"י Google ופורסם בTensorFlow Hub .
- אלגוריתם הלמידה SVM מומש ע"י ספריית sklearn.

סיכום הממצאים ודיון

מטרתנו בפרוייקט זה הייתה לייצר מסווג המסוגל להבחין בין פרסום קריטי העוסק באלימות במשפחה לפרסום שאינו קריטי, זאת במטרה ליצור כלי אשר יוכל לשמש ארגוני סיוע לאלימות במשפחה לאתר מקרים מסוכנים. משימת איסוף הדוגמאות הייתה מאתגרת מאוד מאחר ופרסומים רבים היו שנויים במחלוקת , כאשר תיארו סיפור אישי קשה העוסק באלימות במשפחה מצד אחד אך לא היה סימנים ברורים האם אותו אדם נמצא במצוקה גם היום. בנוסף, בחרנו לעבור על כל הפרסומים באופן ידני ולבחור רק את הפרסומים המשמעותיים ביותר, משימה נוספת שלקחה זמן רב.
לאחר שהיה בידינו מאגר דוגמאות מוכן, התחלנו ללמוד רבות על עולם עיבוד השפה הטבעית ובפרט על משימת סיווג הטקסט. במהלך תהליך הלמידה שלנו, החלטנו שמעניין אותו להתמקד בפרוייקט בשלושה שלבים מרכזיים במשימת סיווג הטקסט שעשויים להשפיע על המודל שיווצר- **העיבוד המקדים של הטקסט** יכול לסייע לנקות רעש מהטקסט והוא גם מכין את הטקסט לשלבים הבאים למשל באמצעות שבירה לאסימונים, **התכונות באמצעותן מייצגים את הטקסט** ישפיעו על יכולת הניבוי של המסווג הסופי לקבל החלטה וכל **אלגוריתם למידה** ילמד מהדוגמאות שיקבל בדרך שונה ולכן ימצא מסווג שונה.

הניסויים שערכנו הראו לנו כי יש קשר בין שלושת המשתנים וכל אחד מהמשתנים עשוי להשפיע על השני. למשל ראינו כי יש אינטראקציה חזקה בין הדרך שמעבדים את הטקסט לבין הדרך בה מייצרים תכונות מהטקסט. בין היתר, ראינו כי עיבוד מקדים לטקסט הכולל הסרת "stop words" השפיע לרעה על מס' מודלים המייצרים תכונות מהטקסט , כך שהתכונות שנוצרו היה פחות אינפורמטיביות ויכולת הניבוי של המסווגים שנוצרו נפגמו. לעומת זאת בשימוש בשיטות עיבוד מקדים מסוג עיבוד מקדים מינימלי או Lemmatization , אותם מודלים ליצירת תכונות הצליחו לייצר תכונות אינפורמטיביות יותר ששיפרו את יכולות הניבוי של המסווגים שנוצרו.
זאת ועוד, ראינו גם כי יש אינטראקציה חזקה בין סוג התכונות שמשתמשים בהן כדי לייצג את הטקסט לבין אלגוריתם הלמידה שמשתמשים בו למציאת מסווג. אלגוריתם למידה כמו SVM למשל הצליח ללמוד בצורה הכי טובה כאשר השתמשנו בתכונות סמנטיות המייצגות כל אחת מהדוגמאות כוקטור נומרי שיש לו משמעות במרחב, ולכן היה קל יותר לאלגוריתם SVM למצוא מישור שהפריד בין הדוגמאות. עם זאת אלגוריתם זה נכשל בתהליך הלמידה שלו כאשר סיפקנו לו תכונות שנוצרו ממודלים כמו "Bag of words". לעומתו, אלגוריתמים כמו MLP דווקא כן הצליחו ללמוד בצורה טובה יותר תכונות כמו אלו שייצר "Bag of words".

עוד ראינו כי יש יתרון קטן בשימוש בייצוגים משולבים, במיוחד בשימוש עם אלגוריתם MLP . ראינו כי לייצוג משולב של תכונות לקסיקליות מסוג TFIDF ביחד עם תכונות סמנטיות היה יתרון במס' מקרים על שימוש בתכונות סמנטיות בלבד, במיוחד ראינו עליה של כ2-3% באחוז הדיוק בשילוב של ייצוג משולב עם אלגוריתם MLP.
נוסף על כך, ראינו כי ייצוג משולב הכולל תכונות תחביריות מסוג POS with TFIDF עשוי להיות יעיל בעיקר בייצוג משולש המכיל תכונות לקסיקליות+ תחביריות+ סמנטיות. גם במקרה זה ראינו שיפור בעיקר בשילוב עם אלגוריתם MLP. לעומת זאת, לא היה נראה יתרון בשימוש של ייצוג משולב של תכונות תחביריות + לקסיקליות או תכונות תחביריות + סמנטיות.
ברור מהתוצאות כי יש קשר בין ניסוי מס' 2 ל3, ונראה שתכונות המיוצרות באמצעות מודל TFIDF, בין אם מדובר בתכונות לקסיקליות או תחביריות, טומנות בחובן מידע חדש שמשפר את יכולת הניבוי של המסווג. כמו כן , ברור כי בעיקר לאלגוריתם הלמידה MLP שמורכב מרשת עצבית יש את היכולת להשתמש במידע החדש הזה.

מכלל הניסויים בפרוייקט הגענו למסקנה כי שיטות העיבוד המקדים שהיו המתאימות ביותר למסמכים שלנו היו Minimal preprocess וLemmatization , התכונות הסמנטיות שייצגו את המידע בצורה הטובה ביותר יוצרו ע"י המודלים DistillRoBERTa, MPNet ו Universal Sentence Encoder, התכונות הלקסיקליות והתחביריות הטובות ביותר יוצרו ע"י מודל TFIDF, ואלגוריתמי הלמידה שהניבו תוצאות מיטיביות היו SVM בשימוש עם תכונות סמנטיות ואלגוריתם MLP בשימוש עם ייצוגים משולבים.

המסווג האופטימלי אותו איתרנו בפרוייקט הנוכחי הגיע ל89.75% דיוק. תוצאות אלו התקבלו בשיטת התוקף הצולב שיושמה במחקר. אחוז הדיוק אליו הגענו במחקר הוא בהחלט ברמה מספקת, במיוחד לאור תוצאות דומות בתחום ובהתחשב בהיקף המוגבל של המחקר שהיה ביכולתנו לבצע.

הסתייגויות ,הערות וכיווני מחקר עתידיים:

1. במהלך הפרוייקט השתמשנו באלגוריתמי למידה המוצעים ע״י ספריית sklearn עם פרמטרי ברירת המחדל. לפרמטרים של אלגוריתמי הלמידה השונים עשויה להיות השפעה על התוצאות, במיוחד במודלים כמו MLP או SVM לכן ייתכן כי עם כוונון נכון של פרמטרים אלו יתקבלו תוצאות שונות.

2. במסגרת הפרוייקט השתמשנו במאגר נתונים קטן יחסית של כ-400 דוגמאות. סביר להניח שהתרחשה תופעה של over-fitting ואם נריץ את המסווג הנבחר על קבוצת מבחן חדשה נקבל אחוז דיוק נמוך יותר מזה שקיבלנו באמצעות תוקף צולב. זאת ועוד, הדוגמאות השליליות במאגר הנתונים שלנו היו קשורות לאלימות במשפחה, מערכות יחסים , דכאון ונושאים קרובים. כדי שהמסווג יוכל לעבוד על כל פרסום אפשרי מהרשת החברתית, יש להכניס דוגמאות שליליות גם מעולמות תוכן שאינם קשורים כלל למצוקה כלשהי.

3. ככיוון מחקר עתידי, אנו ממליצות על דרך אפשרית ליצירת מאגר נתונים באופן אוטומטי, כאשר הסיווג מתבצע באופן אוטומטי ע״י היוריסטיקות מבוססות אוצר מילים או ניתוח רגש (sentiment analysis). דרך זו הוצעה במאמר [49] ליצירת מאגר נתונים למשימת סיווג של פרסומים אובדניים ברשת החברתית אך אנחנו חושבות שהיא עשויה להתאים גם במקרה של אלימות במשפחה.

4. לאחר הניסוי ה1, ראינו כי האלגוריתם SVM לא מצליח לסווג בצורה טובה תכונות שנוצרו ע״י מודל Bag of words. סיבה אפשרית לכך הייתה שSVM אינו רגיש לנרמול. בחרנו שלא לנרמל את המודל Bag of words בניסויים הבאים מאחר ורצינו להמשיך לבחון את מודל Bag of words כפי שהוגדר כהיסטוגרמה של מילים. מבחינתנו הוא הווה מעיין קבוצת ביקורת ביחס למודלים האחרים כיוון שמודל זה הוא מודל פשוט ומוכר לייצוג של טקסטים. ייתכן כי הפעלת נרמול על מודל Bag of words ולאחר מכן הרצת SVM הייתה יוצרת מסווג עם אחוז דיוק גבוה יותר.

5. במסגרת הפרוייקט בחרנו להתעלם מפרסומים ״ניטרליים״ שלא ידענו איזה סיווג לתת להם, אך ב״עולם האמיתי״ צריך להחליט איזה סיווג לתת לפרסומים כאלה מאחר ויש רבים מהם ברשת החברתית. בנוסף, הסיווג ״קריטי״ לפרסום מסוים הוא גם לא מדויק מספיק לדעתנו מאחר וראינו כי יש סיטואציות קריטיות במסגרתו הקורבן מפחד כי בן אדם אחר יפגע בחייו (אירוע פלילי) ויש סיטואציות קריטיות במסגרתו הקורבן במצב נפשי קשה ועשוי לפגוע בו. לכן אנחנו מציעות ככיוון מחקר עתידי לשנות את הסיווג הבינארי לסיווג הכולל מס׳ מחלקות: פרסומים בהם הקורבן שרוי מצב נפשי קשה, פרסומים המתארים סיטואציה פלילית בה הקורבן יכול להיפגע ע״י בן זוגו, פרסומים ניטרליים המתארים סיפור אישי אך לא ברור מצבו של הקורבן, פרסומים הקשורים באלימות במשפחה ואינם מתארים סיטואציה קריטית ופרסומים שאינם קשורים לאלימות במשפחה כלל. אנחנו חושבות שסיווג כזה יותר שימושי עבור ארגוני סיוע וחירום, מאחר וניתן לפצל מקרים בין ארגוני חירום שונים (משטרה למשל יכולה לטפל באירועים פלילים וארגונים אחרים במצב נפשי). בנוסף, ניתן לתת את הפרסומים שיסווגו כניטרליים למעבר ידני ע״י מומחה שיוכל לקבוע על פי הידע שלו האם יש מקום לתת סיוע לקורבן.

6. במסגרת הפרוייקט בחרנו פרסומים יחסית קצרים שמכילים מספר משפטים, מאחר והקלט למקודדים שהשתמשנו בהם הוא קצר ובדר״כ עד כ-500 מילים (למשל במקודדים מספריית sentence transformers או Universal Sentence Encoder). היו מעט פרסומים שנחתכו כאשר השתמשנו במקודדים אלו. כיוון מחקר עתידי שאנו מציעות הוא מציאת דרך להתמודד עם פרסומים ארוכים יותר, בין היתר כיוון אפשרי יכול להיות חלוקה של הפרסום למס׳ סגמנטים , יצירת שיכון עבור כל אחד מהחלקים ולאחר מכן ביצוע ממוצע על כל השיכונים. ניתן גם לשמור על ״חלון״ שיהיה משותף בין הסגמנטים השונים ולבדוק האם זה מביא לייצוג משופר יותר.

7. במסגרת הפרוייקט בדקנו מס׳ מועט של שיטות עיבוד מקדים למסמכים. ככיוון אפשרי למחקר אנו מציעות עוד מס׳ שיטות שאנחנו חושבות שעשויות להשפיע לטובה על התוצאות. ראשית כל, הבחנו כי הרבה מהפרסומים כוללים שגיאות כתיב שיוצרים הבדלים בין מילים , למשל ראינו פרסומים בהם משתמשים במילה dose במקום בdoes.בנוסף יש שימוש רב במילות סלנג וקיצורים במקום בצורה הסטנדרטית והפורמלית של המילה. הבדלים אלו עשויים לפגום במיוחד במודלים מהמשפחה הלקסיקלית שמסתמכים על המילים כתכונות. אנו מציעות לבחון שימוש בכלים שעשויים לעזור בסטנדרטיזציה של המילים ולבחון איך ישפיע תיקון שגיאות כתיב והפיכת קיצורים למילה המלאה על דיוק המודל. בנוסף, במסגרת הפרוייקט השתמשנו ברשימת ״stop words״ אחת בלבד, לכן אנו מציעות לבחון רשימות נוספות ואפילו ליצור רשימת ״stop words״ שעשויה יותר להתאים לאוצר המילים הנפוץ

בפרסומים העוסקים באלימות במשפחה. בנוסף ייתכן כי גם שימוש בשיטות tokenization וlemmatizaion שונות מאלו שהשתמשנו בהן יביא לתוצאות שונות.

8. שיטות עיבוד מקדים נוספות שניתן לבחון במחקרי המשך הן השפעת הניקוד של המסמכים, והשפעתן של אותיות גדולות. במסגרת הפרוייקט ביצענו case folding והפכנו את כל האותיות במסמכים לקטנות כדרך לייצר סטנדרטיזציה בין מילים אך ייתכן ויש לאותיות הגדולות משמעות כלשהי שעשויה לשפר את יכולות הניבוי של המסווג. בנוסף, לא הסרנו ניקוד מהמסמכים וייתכן והם משפיעים לרעה על יכולות הניבוי של המסווג , לכן אנחנו חושבות שיכול להיות מעניין לבחון זאת.

9. כיווני מחקר נוספים יכולים להיות בהקשר של ייצוג הטקסט ויצירת תכונות. אפשר לבחון מודלים נוספים ליצירת שיכוני מילים כמו word2vec,GloVe,ElMo . בנוסף אפשר לבחון מודלי שפה נוספים כמו GPT-3 ולנסות להשתמש בembeddings שהם מייצרים. רעיון נוסף הוא בחינה של ייצוגים משולבים הכוללים שרשור של מס׳ סוגים של תכונות סמנטיות, כמו למשל doc2vec+inferSent.

בנימה אישית

הפרוייקט תרם לנו רבות הן מבחינה מקצועית והן מבחינה אישית.
מבחינה מקצועית, בזכות הפרוייקט נחשפנו לעולם העיבוד השפה הטבעית ולאתגרים הרבים שיש בניתוח מידע טקסטואלי. בנוסף למדנו רבות על אלגוריתמי למידה שונים וגם על מודלים מתחום הלמידה עמוקה.
התנסינו במחקר, בקריאת מאמרים והתמודדות עם אתגרים בלתי צפויים בזמן שעושים מחקר.
מבחינה אישית, נחשפנו לנתונים רבים וקשים הקשורים לתופעת האלימות במשפחה ולהיקפה. קראנו סיפורים אישיים רבים על נשים שעוברות התעללות פיזית ונפשית ממושכת מצד בן זוגם ורק מחפשות אדם שיעזור להן וגם קראנו סיפורי גבורה ונדהמנו מהכוח והחוזק שקיים באנשים שעברו חוויות כה קשות.
הנושא של הפרוייקט והרצון לנסות ולו לסייע במעט כדי למגר את תופעת אלימות המשפחה יצרו אצלנו מוטיבציה רבה במהלך הכנת הפרוייקט , מאחר וידענו את ההשפעה שיכולה להיות אפילו משיפור קטן באחוזי הדיוק של המסווג.
בשל כל הסיבות הללו, אנחנו שמחות מאוד שקיבלנו את ההזדמנות להשתתף בקורס זה ולראות כיצד ניתן להשתמש בכלים מעולם למידת המכונה על מנת לתרום לחברה ואף להציל חיים.

Bibliography

[1] "What is domestic abuse?," United Nations, [Online]. Available: https://www.un.org/en/coronavirus/what-is-domestic-abuse. [Accessed 27 09 2021].

[2] "Violence against women prevalence estimates, 2018: global, regional and national prevalence estimates for intimate partner violence against women and global and regional prevalence estimates for non-partner sexual violence against women," World Health Organization, Geneva, 2021.

[3] "Global Study on Homicide 2019," United Nations Office on Drugs and Crime , 2019, p. 10.

[4] Black, M.C., Basile, K.C., Breiding, M.J., Smith, S.G., Walters, M.L., Merrick, M.T., Chen, J., & Stevens, M.R. (2011). The National Intimate Partner and Sexual Violence Survey (NISVS): 2010 Summary Report. Atlanta, GA: National Center for Injury Prevention and Control, Centers for Disease Control and Prevention.

[5] "Artificial neural network," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed 21 10 2021].

[6] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory to algorithms. Cambridge: Cambridge University Press, 2019.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. Cambridge, Mass: The MIT Press, 2017.

[8] "Recurrent neural network," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network. [Accessed: 21-Oct-2021].

[9] Krishni, "A high-level introduction to lstms," *Medium*, 19-Apr-2019. [Online]. Available: https://medium.datadriveninvestor.com/a-high-level-introduction-to-lstms-34f81bfa262d. [Accessed: 21-Oct-2021].

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, pages 5998– 6008, 2017 1, 3, 5.

[11] "Transformer," *Wikipedia*, [Online]. Available: https://he.wikipedia.org/wiki/Transformer. [Accessed: 21-Oct-2021].

[12] A. Raviv and M. Erlihson, Machine and Deep learning in Hebrew, in Proc, 2021.

[13] U. Upadhyay, "Knowledge distillation," *Medium*, 04-Jun-2020. [Online]. Available: https://medium.com/neuralmachine/knowledge-distillation-dc241d7c2322. [Accessed: 21-Oct-2021].

[14] "Embeddings | machine learning crash course | google developers," *Google*. [Online]. Available: https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture. [Accessed: 21-Oct-2021].

[15] D. Jurafsky and J. Martin, "Speech and Language Processing (3rd ed. draft)," *Speech and Language Processing*, 2021. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/. [Accessed: 21-Oct-2021].

[16] A. Aponyi, "What are sentence embeddings and their applications?," *TAUS*, 10-Feb-2021. [Online]. Available: https://blog.taus.net/what-are-sentence-embeddings-and-their-applications. [Accessed: 21-Oct-2021].

[17] L. Torrey and J. Shavlik. Transfer learning. IGI Global, 2009

[18] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, "Pre-trained models for Natural Language Processing: A Survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[19] O. Press, "Neural language models explained," *Neural Language Models Explained –*, 2017. [Online]. Available: https://ofir.io/Neural-Language-Modeling-From-Scratch/. [Accessed: 21-Oct-2021].

[20] "Language model," *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Language_model#Benchmarks. [Accessed: 21-Oct-2021].

[21] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. arXiv preprint arXiv: 1602.02410, 2016.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv: 1810.04805, 2018.

[23] "Natural language inference," *NLP*. [Online]. Available: http://nlpprogress.com/english/natural_language_inference.html. [Accessed: 21-Oct-2021].

[24] "The Stanford NLP Group," *The Stanford Natural Language Processing Group*. [Online]. Available: https://nlp.stanford.edu/projects/snli/. [Accessed: 21-Oct-2021].

[25] S. Subramani, H. Wang, H. Q. Vu, and G. Li, ''Domestic violence crisis identification from facebook posts based on deep learning,'' IEEE Access, vol. 6, p. 54 075–54 085, 2018

[26] K. Ganesan, "What are stop words?," *Opinosis Analytics*, 04-Aug-2020. [Online]. Available: https://www.opinosis-analytics.com/knowledge-base/stop-words-explained/#.YXE4R9lBy7N. [Accessed: 21-Oct-2021].

[27] *Dropping common terms: Stop words*. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html. [Accessed: 21-Oct-2021].

[28] H. Saif, M. Fernandez, Y. He, and H. Alani, ''On stopwords, filtering and data sparsity for sentiment analysis of Twitter,'' in Proc. 9th Lang. Resour. Eval. Conf. (LREC), Reykjavik, Iceland, 2014, pp. 80–81

[29] S. Prabhakaran, "Lemmatization approaches with examples in Python," *Machine Learning Plus*, 13-Oct-2021. [Online]. Available: https://www.machinelearningplus.com/nlp/lemmatization-examples-python/#wordnetlemmatizer. [Accessed: 21-Oct-2021].

[30] S. Chandran, "Introduction to text representations for Language processing-part 1," *Medium*, 06-Jun-2021. [Online]. Available: https://towardsdatascience.com/introduction-to-text-representations-for-language-processing-part-1-dc6e8068b8a4. [Accessed: 21-Oct-2021].

[31] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," arXiv preprint 27 arXiv: 1607.04606, 2016

[32] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In Proceedings of The 31st International Conference on Machine Learning, pages 1188–1196, 2014.

[33] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), September 2017, pp. 670–680.

[34] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. GuajardoCespedes, S. Yuan, and C. Tar, "Universal sentence encoder," arXiv preprint arXiv: 1803.11175, 2018.

[35]] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. arXiv preprint arXiv: 2004.09297, 2020.

[36] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," arXiv preprint arXiv: 2002.10957, 2020.

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv: 1907.11692, 2019.

[38] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS, 2019.

[39] E. Kouloumpis, T. Wilson, and J. Moore, ''Twitter sentiment analysis: The good the bad and the omg!'' in Proc. 5th Int. AAAI Conf. Weblogs Social Media, 2011, pp. 538–541.

[40] A. Al-Tubaity, A. Alqarni, and A. Alnafessah, "Do Words with Certain Part of Speech Tags Improve the Performance of Arabic Text Classifcation?" in Proceedings of the the 2nd International Conference, pp. 155–161, Lakeland, FL, USA, April 2018.

[41] Robinson, T., 2016. Disaster tweet classification using parts-of-speech tags: a domain adaptation approach. Ph.D. thesis. Kansas State University.

[42] "1.17. neural network models (supervised)," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html. [Accessed: 21-Oct-2021].

[43]"A practical explanation of a naive Bayes classifier," *MonkeyLearn Blog*, 25-May-2017. [Online]. Available: https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/. [Accessed: 21-Oct-2021].

[44] "1.11. ensemble methods," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html. [Accessed: 21-Oct-2021].

[45] H. Saif, Y. He, and H. Alani, ''Semantic sentiment analysis of twitter,'' in Proc. Semantic Web-ISWC, 2012, pp. 508–524,2012.

[46] Nothman, J., Qin, H. and Yurchak, R., 2018, July. Stop Word Lists in Free Open-source Software Packages. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS) (pp. 7-12).

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, 2015

[48] C. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, 2005.

[49] Ríssola EA, Bahrainian SA, Crestani F. (2020) A Dataset for Research on Depression in Social Media. In Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 338–342.