FAKE NEWS DETECETIONS

כותבים:

אלון בריל, ת"ז: 206910192

עמרי מלכה, ת"ז: 313540288

תאריך הגשה: 30.4.2021

מנחה:

ד"ר דינה גורן-בר

הבעייתיות בפייק-ניוז

פייק-ניוז הוא מונח המשמש לחדשות מפוברקות. פייק-ניוז הן המצאה – שקר שנוצר מכלום, שקר אשר מקבל מראה של חדשות אמתיות עם מטרה להטעות אנשים. חשוב לזכור, המידע בידיעה הוא שקרי, אבל זה נראה נכון.

ידיעות שקריות אלו מקבלות במה יותר נרחבת במדיה החברתית מאשר חדשות אמיתיות בגלל שהמידע בהן לרוב מעניין יותר (מעניין אך לא אמיתי), השילוב של מהירות התפוצה ברשתות חברתיות והמידע השקרי שבידיעות הוא שילוב מסוכן ביותר.

הסכנות בפייק-ניוז:

* פייק-ניוז משפיע על הדעה הפוליטית שלנו.
* פייק-ניוז גורם לנו לצרוך פחות מידע.
* פייק-ניוז גורם לנו להאמין למידע רפואי שקרי ומסוכן.
* פייק-ניוז מסוכן לחיינו.



הגדרת הבעיה

הבעיה שהפרויקט שלנו בא לפתור היא בעיית זיהוי פייק-ניוז לפי תוכן ידיעה.

הדרך שבה אנו ניגש לפתרון הבעיה היא באמצעות שימוש בלמידת מכונה ועיבוד שפה טבעית לסיווג טקסט של ידיעות.

# לקוחות פוטנציאלים

אפליקציות ואתרי מדיה חברתית יוכלו ליישם את האלגוריתם שלנו על מנת לחסום פייק-ניוז או להזהיר את המשתמשים שלהם מפניהם.

# מצב השוק כרגע

חברת פייסבוק פועלת לקראת חסימת פייק-ניוז ע"י מערכת דיווחים מהמשתמשים.

חברת טוויטר פיתחה מערכת שמשלבת למידת מכונה לזיהוי ציוצים שקריים בנושא הקורונה.

מטרות הפרויקט

מטרת הפרויקט היא להצליח לסווג בין חדשות אמיתיות לפייק-ניוז לפי טקסט של ידיעה.

# אבני דרך

1. נחקור את מאגר המידע Liar-liar (על מאגר מידע זה נאמן את המודלים שלנו).
2. נלמד אלגוריתמים רלוונטיים של עיבוד שפה טבעית.
3. נבנה מודל חיזוי ראשוני.
4. Feature Engineering.
5. נבנה מודל חיזוי משופר.
6. ננתח את התוצאות מ4 ו5.
7. נחזור על צעדים 4 עד 6 עד שנקבל תוצאות מספקות.

סקירת ספרות

# Fake news detection using machine learning by Simon Lorent

<https://matheo.uliege.be/bitstream/2268.2/8416/1/s134450_fake_news_detection_using_machine_learning.pdf>

תזה שנכתבה ע"י סימון לורנט על סיווג פייק-ניוז. בתזה זו סימון מציג טכניקות שונות בלמידת מכונה ועיבוד שפה טבעית שבעזרתן הוא בנה ואימן מודלים לסיווג פייק-ניוז.

סימון השתמש בשני מאגרי מידע - Fake news ו- Liar-liar. אחוזי דיוק הסיווג שאליהם סימון הגיע היו 94% דיוק על Fake news ו- 61% דיוק על Liar-liar.

# Identifying fake news: The LIAR dataset and its limitations by Yingzhao Ouyang

<https://towardsdatascience.com/identifying-fake-news-the-liar-dataset-713eca8af6ac>

במאמר זה הכותב ניסה לסווג פייק-ניוז ממאגר המידע Liar-liar תוך שימוש בטכניקות של עיבוד שפה טבעית. הוא השיג תוצאת דיוק של 58%.

# ‘Fake news’: Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions by Jonas De Keersmaecker

<https://www.sciencedirect.com/science/article/abs/pii/S0160289617301617>

במחקר זה גילו שחשיפה לפייק-ניוז משפיע על דעתו של קורא גם לאחר שהוסבר לו שהידיעה שקרא שקרית. מתגלית זו ניתן להבין שנשקפת סכנה **לכולנו** בפופולריות של פייק-ניוז ברשתות החברתיות.

# החידוש שלנו

ישנם שתי מאגרי מידע עיקריים לפייק-ניוז- Fake news ו- Liar-liar. Fake news מכיל פייק-ניוז רק מפוסטים ממדיה חברתית, על מאגר מידע זה הצליחו להשיג תוצאות סיווג טובות.

לעומת זאת, Liar-liar מכיל פייק-ניוז מפוסטים ממדיה חברתית, רדיו, עיתונים, ומקורות אחרים וכתוצאה מכך, נחשב קשה ללמידת מכונה.

עדיין לא הצליחו להשיג תוצאות סיווג טובות על Liar-liar.

אנחנו ננסה להשיג תוצאות סיווג טובות על Liar-liar מכיוון שהוא משקף את הפייק-ניוז בעולם האמיתי.

ניתוח מתחרים

|  |  |  |
| --- | --- | --- |
| אחוז דיוק | אלגוריתם | כותב |
| 0.61 | Attention mechanism | Simon Lorent |
| 0.58 | Topic modeling & basic NLP | Yingzhao Ouyang |

האתגר בפרויקט שלנו הוא להשיג אחוז דיוק יותר טוב מזה של Simon Lorent שהשיג את אחוז הדיוק הגבוה ביותר עד כה.

הפתרון

מימשנו שני פתרונות שונים, ההבדל בין הפתרונות הוא הדרך בא אנו ממירים את הטקסט שבמסמך לייצוג מספרי.

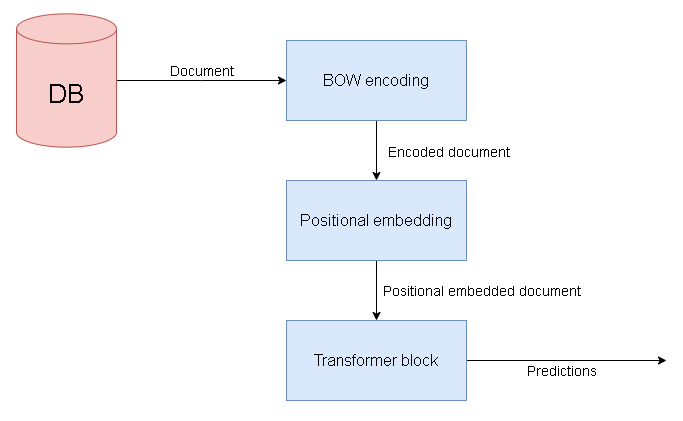
בפתרון הראשון מיצרים מילון מילים שבו מופיעות המילים שבקורפוס ולכל מילה מספר ייחודי, ממירים את המילים במסמך למספרים המתאימים במילון, שיטה זו נקראת Bag-of-words או בקיצור (BOW).

בפתרון השני מייבאים מודל טרנספורמר מאומן שנקרא Bert. מזינים לתוכו את המסמכים בקורפוס ומוצאים מתוך המודל את הייצוג הפנימי של המסמכים ש Bert למד - מערך וקטורים ש Bert למד לכל מילה במסמך.

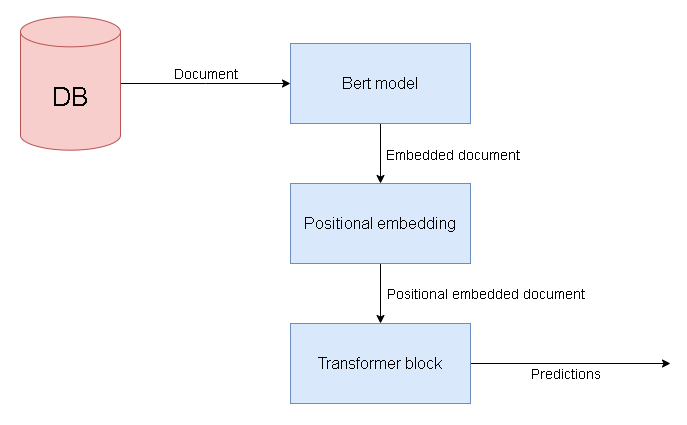
קראנו לשני הפתרונות על שם הדרך שבא המרנו את המסמכים לייצוג מספרי.

להלן דיאגרמות של הפתרונות:

# Bag of Words



# Bert



מימוש הפתרון Bag-of-words

1. מנקים את קורפוס האימון והמבחן.

1.א. מוחקים תווים מיוחדים וממירים את כל האותיות לאותיות קטנות.

1.ב. מוחקים stop words.

1.ג. מחזירים כל מילה לשורש שלה.

2. מחשבים ערכי TF-IDF לכל מילה בקורפוס.

3. שומרים מילים עם ערכי TF-IDF גבוהים.

4. יוצרים מילון של המילים שקיבלנו בסעיף 3 כך שכל מילה משויכת למספר ייחודי.

5. מקודדים את המסמכים - כל מילה מוחלפת במספר שאליו היא משויכת במילון.

6. ממירים את תוויות הסיווג לייצוג One-hot.

7. לכל מסמך, לכל מיקום מילה במסמך נחשב ערך מיקום בעזרת נוסחת Positional encoding.

8. לכל מסמך, מזינים את המסמך המקודד לשכבת נוירונים.

9. לכל מסמך, מזינים את ערכי המיקומים של המסמך לשכבת נוירונים.

10. לכל מסמך, מחברים את שתי שכבות הנוירונים שקיבלנו בסעיפים 8 ו- 9.

11. בונים מודל Transformer.

12. מאמנים את המודל.

12.א. מפצלים את מסמכי האימון ל Batches.

12.ב. לכל Batch בקורפוס האימון, לכל מסמך ב Batch, מזינים למודל את השכבה שקיבלנו בסעיף 10.

12.ג. משווים את חיזויי המודל אל מול תוויות הסיווג.

12.ד. מתקנים את משקולות המודל בטכניקת Back propagation.

12.ה. חוזרים על צעדים א' עד ד' למספר איפוקים שקבענו מראש

13. נבחן את המודל אל מול קורפוס המבחן.

מימוש הפתרון Bert

1. מייבאים מודל טרנספורמר מאומן Bert.

2. מזינים את הקורפוס ל- Bert.

3. מקבלים מBert את הייצוג הפנימי של כל מסמך בקורפוס.

4. ממירים את תוויות הסיווג לייצוג One-hot.

5. לכל מסמך, לכל מיקום מילה במסמך נחשב ערך מיקום בעזרת נוסחת Positional encoding.

6. לכל מסמך, מחברים את הייצוג הפנימי של המסמך עם ערכי המיקום שלו.

7. לכל מסמך, מזינים את התוצאה שקיבלנו עבורו בסעיף 6 לשכבת נוירונים.

8. בונים מודל Transformer.

9. מאמנים את המודל.

9.א. מפצלים את מסמכי האימון ל Batches.

9.ב. לכל Batch בקורפוס האימון, לכל מסמך ב Batch, מזינים למודל את השכבה שקיבלנו בסעיף 7.

9.ג. משווים את חיזויי המודל אל מול תוויות הסיווג.

9.ד. מתקנים את משקולות המודל בטכניקת Back propagation.

9.ה. חוזרים על צעדים א' עד ד' למספר איפוקים שקבענו מראש

10. נבחן את המודל אל מול קורפוס המבחן.

# המודלים השונים

בנינו, אימנו ובחנו שישה מודלים שונים, שלושה לפי פתרון BOW ושלושה לפי פתרון Bert.

להלן תיאור ההבדלים בין המודלים:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification | Labels | Data | Input type | Model |
| Multiclass | All train labels | All train documents | BOW encodings | BOW, Model1 |
| Binary | Only ‘true’, ‘false’ labels from train | Train documents labeled as ‘true’, ‘false’ | BOW encodings | BOW, Model2 |
| Binary | All train labels merged to ‘true’ and ‘false’ (\*) | All train documents | BOW encodings | BOW, Model3 |
| Multiclass | All train labels | All train documents | Bert embeddings, M.S.L = 128 (\*\*) | Bert, Model1 |
| Multiclass | All train labels | All train documents | Bert embeddings, M.S.L = 512 (\*\*) | Bert, Model2 |
| Binary | Only ‘true’, ‘false’ labels from train | Train documents labeled as ‘true’, ‘false’ | Bert embeddings, M.S.L = 512 (\*\*) | Bert, Model3 |

(\*) – מיזוג התוויות, הפכנו את התוויות "שקר מוחלט", "לא נכונה" ו- "בקושי אמת" לתווית "לא נכונה" ואת התוויות "חצי אמת", "כמעט אמת" ו- "אמת" לתווית "אמת". המטרה במיזוג התוויות הייתה ללמוד סיווג בינארי על כל המסמכים בקורפוס.

(\*\*) – M.S.L = Max sequence length, כשמזינים את המסמכים ל Bert פרמטר זה מעיד על אורך המסמכים (במילים) שאותם Bert מקבל. הוצאנו שני ייצוגים פנימיים של המסמכים, ייצוגים ש Bert למד עבור M.S.L = 128 וייצוגים ש Bert למד עבור M.S.L = 512. המטרה בשני הייצוגים הייתה לבדוק האם מסמכים ארוכים מידי מהווים רעש לתהליך הלמידה.

הערה – חשוב לציין ש Simon Lorent המתחרה הראשי שלנו, עבד עם אותו Data ו- Labels שהמודלים BOW, Model2 ו- Bert, Model 3 עבדו.

תוצאות ומדדים

# מבנה מאגר המידע

מאגר המידע שלנו מפוצל לקבוצות אימון ומבחן ולכל קבוצה יש תוויות סיווג מוכנות.

השתמשנו בשתי עמודות מתוך מאגר המידע - עמודת התוויות, ועמודת הטקסט.

* עמודת התוויות – עמודה המכילה שש סיווגים שונים לידיעה.
* עמודת הטקסט – עמודה המכילה את הידיעות עצמן.

סיווגים אפשריים לידיעה הם : שקר מוחלט / לא נכונה / בקושי אמת / חצי אמת / כמעט אמת / אמת.

כל הידיעות נאספו מאתר אמריקאי בשם: POLITIFACT.COM אשר מדרג את נכונות ההצהרות/ידיעות אשר סיפקו אנשי ציבור אמריקאים.

# בדיקת הצלחת המודל

חשוב לנו להגיע לאחוז דיוק גבוה (Accuracy) על מנת להצליח בסיווג של ידיעה, בנוסף לא נרצה לטעות בסיווג ידיעה מזויפת (Recall, low FN) (על חשבון סווג ידיעה נכונה כידיעה מזויפת).

# אופן הבדיקה

לאחר שאימנו את המודל על קבוצת האימון בטכניקת Cross-validation נחשב את חיזויי המודל על קבוצת המבחן ונבדוק מהם אחוזי ה- Accuracy וה- Recall.

# תוצאות

|  |  |  |
| --- | --- | --- |
| Recall | Accuracy | Model |
| 0.013 | 0.293 | BOW, Model1 |
| 0.695 | 0.695 | BOW, Model2 |
| 0.653 | 0.653 | BOW, Model3 |
| 0.000 | 0.201 | Bert, Model1 |
| 0.161 | 0.164 | Bert, Model2 |
| 0.409 | 0.490 | Bert, Model3 |

סיכום ומסקנות

# ניתוח התוצאות

המדד העיקרי שלנו להצלחת הפרויקט הוא להשיג אחוז דיוק (Accuracy) יותר גבוה מזה של סימון לורנט שקיבל אחוז דיוק של 0.61. הצלחנו, במודל BOW, Model2 קיבלנו אחוז דיוק של 0.69, המסקנה שאנחנו מסיקים היא שמודל הטרנספורמר מתאים יותר לסיווג פייק ניוז מאשר מודל ה- LSTM שאיתו סימון סיווג.

התוצאות שקיבלנו בעזרת הייצוגים הפנימיים של Bert הן נמוכות, ישנן שתי סיבות אפשריות לתוצאה זאת:

* אורכי המסמכים שלנו ארוכים יותר מאשר האורך המקסימלי ש Bert יכול לקבל (512 מילים) והמודל לא קיבל מספיק מידע (שלא כמו בפתרון BOW).
* מודל Bert מאומן לסיווג סנטימנטלי, ייתכן שהייצוגים הפנימיים שלו לא מספקים מידע איכותי לסיווג פייק ניוז כמו קידוד One-hot שביצענו בפתרון BOW.

לבסוף, המודלים לא מצליחים לבצע סיווג Multiclass, אחוזי הדיוק וה- Recall מאוד נמוכים.

# המלצות להמשך פיתוח

* לנסות לשלב עוד פיצ'רים ממאגר המידע.
* ניתן לחבר את עמודת הטקסט עם עמודת הכותרת וללמוד מהעמודה המשולבת.
* שימוש בייצוגים פנימיים של המסמכים ממודלים מאומנים נוספים.
* יצירה ושימוש בפיצ'ר המתייחס ל sentiment של ידיעה, ניתן לקבל סיווג סנטימנטלי ממודל מאומן כמו Hug-face.
* לנסות גדלי וקטורים שונים לייצוג מילה בעזרת Bert.

רשימת מקורות

Keras transformer model:

<https://keras.io/examples/nlp/text_classification_with_transformer/>

Bert contextualized embeddings:

<https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b>

extracting words with TF-IDF :

<https://kavita-ganesan.com/extracting-keywords-from-text-tfidf/#.YIVJ7pAzaUk>