­­­­­­­­

**חילוץ ישויות בעזרת אלגוריתמים של למידת מכונה**

Text, chat or text message

Description automatically generated

איור – [23]

מגיש: יונתן טויב

בהנחיית ד"ר מיה הרמן

1. תוכן עניינים

[1 מבוא 4](#_Toc78990563)

[2 עיבוד שפה טבעית 5](#_Toc78990564)

[3 אפיון בעיית חילוץ ישויות מטקסט 6](#_Toc78990565)

[4 קשיים 11](#_Toc78990566)

[4.1 תלות בשפה 12](#_Toc78990567)

[4.2 תלות בתחום 13](#_Toc78990568)

[5 שיטות מדידה 14](#_Toc78990569)

[5.1 השוואה מדויקת 14](#_Toc78990570)

[5.2 השוואה מקלה 16](#_Toc78990571)

[6 דרכי פתרון מבוססות חוקות 17](#_Toc78990572)

[7 דרכי פתרון בעזרת למידה מבוססת פיצ'רים 18](#_Toc78990573)

[7.1 אלגוריתם CRF 19](#_Toc78990574)

[8 ייצוג מידע טקסטואלי 20](#_Toc78990575)

[8.1 ייצוג מבוסס תדירות 21](#_Toc78990576)

[8.1.1 Count Vector 23](#_Toc78990577)

[8.1.2 TF-IDF 24](#_Toc78990578)

[8.2 distributed representations 26](#_Toc78990579)

[8.2.1 ייצוג מבוסס מילים 28](#_Toc78990580)

[8.2.2 ייצוג מבוסס אותיות 34](#_Toc78990581)

[8.2.3 טוקנים 35](#_Toc78990582)

[9 דרכי פתרון מבוססות רשתות נוירונים 36](#_Toc78990583)

[9.1 מבוא 36](#_Toc78990584)

[10 ארכיטקטורת CNN 38](#_Toc78990585)

[11 ארכיטקטורת RNN 39](#_Toc78990586)

[12 ארכיטקטורת LSTM 42](#_Toc78990587)

[13 ארכיטקטורת Bidirectional LSTM 44](#_Toc78990588)

[14 מנגנון Attention 45](#_Toc78990589)

[14.1 Self-attention 46](#_Toc78990590)

[14.2 multi head attention 49](#_Toc78990591)

[14.3 Transformer 50](#_Toc78990592)

[15 BERT 53](#_Toc78990593)

[16 שילובים מתקדמים 54](#_Toc78990594)

[16.1 RNN + CRF 54](#_Toc78990595)

[16.2 Bidirectional LSTM+CRF 54](#_Toc78990596)

[16.3 BERT+CRF 57](#_Toc78990597)

[17 כיוון מחקר- מודל רב שפה 58](#_Toc78990598)

[18 סיכום 58](#_Toc78990599)

[19 ביבליוגרפיה 59](#_Toc78990600)

# מבוא

חילוץ ישויות היא תת בעיה בעיבוד שפה טבעית ( NLP ) העוסקת בחילוץ וסיווג ישויות שהוגדרו מראש מתוך טקסט נתון. לאורך השנים נבדקו ונוצרו דרכים רבות להתמודדות עם בעיה זו.

בשנים האחרונות נוצרה התקדמות גדולה בעולמות למידת המכונה ובפרט ברשתות העמוקות וכחלק מזה תחום עיבוד השפה הטבעית הגיע להישגים רבים הכוללים התפתחות משמעותית במשימת חילוץ הישויות.

בסמינר זה נכיר ונעמיק בחלק מהפתרונות החשובים שניסו להתמודד עם בעיה זו עד לימינו ולדרכים המודרניות שמשתמשים בהן. בנוסף יוצגו כיווני מחקר עכשוויים.

הסמינר יעסוק בקשיים ההופכים בעיה זו למורכבת ובדכים בהן ניתן להתמודד עם קשיים אלו.

# עיבוד שפה טבעית

עיבוד שפה טבעית הוא תחום מחקר בעולם הבינה המלאכותית והבלשנות העוסק בבעיות הנדרשות להתמודד עם שפה אנושית, כפי שזו מדוברת על ידי אנשים. המשמעות היא שאין פורמט מוגדר שבו מגיעים הנתונים.

מכאן שהמודל צריך למצוא דרך "להבין" את השפה, יש שימושים רבים מאד לכך כדוגמת:

* Part of speech – סיווג חלקי טקסט שונים - ראה הסבר בהמשך
* תרגום של טקסט ושל מלל – בעיה מרכזית וקשה ששנים מנסים למצוא לה פתרון
* הוספת כתוביות טקסט לתוכן אודיו – למשל לצורך סיכומי דיונים, הרצאות או עזרה בתקשורת עם כבדי שמיעה.
* מילות מפתח בטקסט – סיכום קצר של מידע – מאפשר חיפוש מתקדם

חלק מהבעיות בעולם זה משמשות כתת משימות בבעיות אחרת ( חלק מ PIPELINE ), דוגמאות טובות לכך הם משימות כמו:

* זיהוי חלקי דיבור ( part of speech ) – סיווג חלקי הטקסט השונים לקטגוריות מורפולוגיות כגון שמות עצם, שמות תואר, פעלים, מילות חיבור ועוד. למשימות כדוגמת חילוץ מילות מפתח, הוצעו פתרונות רבים שמתחילים מהגדרת מועמדים למילות מפתח ומתבססים על צמצום האפשרויות בשלב זה. במקרים רבים, סינון המועמדים מתבסס על סיווג ראשוני של המילים לשתי קטגוריות בלבד, שמות עצם ושמות תואר. לכל משפט נבדקת כמות שמות העצם ושמות התואר שהוא מכיל.
* פירוק (Tokenization): פירוק הטקסט ליחידות קטנות שיקראו טוקנים.

# **אפיון בעיית חילוץ ישויות מטקסט**

חילוץ ישויות היא תת בעיה בעיבוד שפה טבעית. תחילה נגדיר קטגוריות שמעניינות אותנו כדוגמת שמות אנשים, מקומות, ארגונים ועוד. כעת בהינתן קלט טקסטואלי, נרצה לחלץ ממנו את הישויות שהגדרנו ולסווג אותן בהתאם.

חילוץ ישויות משמש כחלק ראשוני מדרכי פתרון של בעיות רבות בתחום עיבוד שפה טבעית ( NLP ).

מספר דוגמאות לשימוש:

* דמיון בין מסמכים, ניתן להשתמש בחילוץ ישויות ( כחלק ראשוני בתהליך) כדי להגדיר וקטור מייצג למשפט \ מסמך, ואז להגדיר דמיון בין מסמכים באמצעות זווית \ מרחק אוקלידי בין הווקטורים המייצגים של מסמכים אלו – ראה פירוט בהמשך
* ניתן לסכם מסמך \ כתבה \ ביקורת בעזרת חילוץ ישויות, מתוך הנחה שכל המידע ה"חשוב" שמעניין אותנו יחולץ בעזרתו
* מערכת שאלות ותשובות

ועוד

דוגמאות תלויות תחום \ שפה - בהם נתמקד בעבודה:

* דיווחי מודיעין על איומים ברשת משמשים חברות ברחבי העולם על בסיס יומיומי עבור אבטחה. איומי סייבר מיועדים גם למשתמשי קצה וגם לעסקים ללא קשר לגודל שלהם וכתוצאה מכך כמות גדולה מאד של אנליסטים ואנשים פרטיים משקיעים זמן רב בניתוח מידע ואיומים. ניתוח כמויות עצומות של מידע דורש זמן ומאמץ רבים, וכתוצאה מכך גם כסף רב. ניתן לנקוט בגישת חילוץ ישויות לבעיה זו. נגדיר את הקטגוריות החשובות מבחינתנו, כדוגמת תוכנות זדוניות, דומיין, כתובת, HASH, IP ועוד. לחילוץ ישויות משמעותיות באמצעות מתודולוגיה זו עשוי להיות ערך גבוה עבור חברות אבטחת סייבר.
* טקסט בשפה הרוסית בבסיסי נתונים מסוג:
  + Gareev – 97 מסמכים שנאספו מספריית האינטרנט של "Yandex News" בנושא עסקים
  + Person-1000 – נתונים טקסטואליים ברוסית שבהם סומנו שמות אנשים. בסיס נתונים זה מכיל חומרים משירותי החדשות המקוונים הרוסיים.
  + FactRuEval 2016 – בסיס נתונים המכיל טקסטים חדשותיים ואנאליסטים ברוסית. מקור בסיס הנתונים הוא אתרים פרטיים ו wikinews. נושאי הטקסטים הם חברתיים ופוליטיים
* טקסט בשפה הפורטוגזית HAREM ו MiniHAREM המכילים ישויות מתויגות בקטגוריות כדוגמת: מיקום, שמות אנשים, ארגונים, ערך, תאריך, כותרת, אירוע ועוד

ישנן אפשרויות רבות להגדרת בעיית חילוץ ישויות, מכיוון שהשאלה הראשונה היא איזה קטגוריות נבחר. בדרך כלל נחלק את התשובה לשתי אפשרויות:

* קטגוריות גנריות: כדוגמת מיקום, שמות אנשים
* קטגוריות תלויות תחום: כדוגמת חלבונים, אנזימים וגנים

יש מספר רב של דרכים להתמודד עם הבעיה, ולאורך השנים עם התפתחות עולם הרשתות העמוקות, גם לבעיה זו הוצגו פתרונות רבים מתחום זה שהראו שיפורים משמעותיים. באופן כללי ניתן לחלק את דרכי ההתמודדות עם הבעיה ל 4 אפשרויות מרכזיות:

* גישות מבוססות חוקים
* גישות מבוססות מידע לא מתויג
* גישות מבוססות הגדרת פיצ'רים
* גישות מבוססות רשתות נוירונים

באופן פורמאלי יותר, בהינתן רצף של טוקנים המיוצגים על ידי ווקטורים מתאימים, נרצה להחזיר מערך של שלשות המגדירים לכל מופע של ישות שנמצאה בתוך הרצף, את האינדקס של הווקטור הראשון של הטוקן השייך לה, האינדקס של הווקטור האחרון ואת הישות עצמה. הסיבה לשימוש באינדקסים במקום שימוש פשוט בטוקן, היא שיש מצבים בהם אותה ישות מתפרסת על מספר טוקנים. לדוגמא, נניח ונגדיר כל טוקן כמילה בטקסט ( נושא ייצוג הטקסט יוסבר באופן מפורט בהמשך העבודה ) במשפט הבא: I was born in Trenton, New Jersey" נצפה לקבל שני אובייקטים שהקטגוריה שלהם היא מיקום, ולא לשלושה. למרות שהטוקנים עצמם מוגדרים כמילה במשפט, עדיין היינו מצפים שהמילים “New”,” Jersey” יאוחדו לישות אחת.

דוגמה להמחשה:

Text

Description automatically generated

איור - [4]

עד שנת 2005 עיקר מסדי הנתונים הייעודיים עבור משימות חילוץ הישויות – הכוונה למסדי נתונים מתויגים – הגיעו ממאמרי חדשות עם מספר מצומצם של קטגוריות לסיווג.

בשנים שלאחר מכן חל שינוי משמעותי, גם בתחומים מהם הגיע המידע וגם במספר הקטגוריות. להלן מספר דוגמאות למסדי נתונים רלוונטיים לשפה האנגלית [4]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| שם | שנת | מקור הטקסט | כמות קטגוריות | כתובת |
| MUC-6 | 1995 | עיתוןWall Street | 7 | <https://catalog.ldc.upenn.edu/LDC2003T13> |
| MUC-7 | 1997 | חדשות New York Times | 7 | <https://catalog.ldc.upenn.edu/LDC2001T02> |
| OntoNotes | 2007 -2012 | מגזינים, חדשות, אתרי אינטרנט ועוד | 18 | <https://catalog.ldc.upenn.edu/LDC2013T19> |
| BBN | 2005 | עיתון Wall Street | 64 | <https://catalog.ldc.upenn.edu/LDC2005T33> |
| WikiGold | 2009 | וויקיפדיה | 4 | https://figshare.com/articles/Learning\_multilingual\_named\_entity\_recognition\_from\_Wikipedia/5462500 |
| WikiFiger | 2012 | וויקיפדיה | 112 | <https://github.com/xiaoling/figer> |
| HYENA | 2012 | וויקיפדיה | 505 | https://www.mpi-inf.mpg.de/departments/databases-andinformation-systems/research/yago-naga/hyena |
| BC5CDR | 2015 | PubMed ( תחומי מדעי החיים, מדעי הרפואה ועוד ) | 3 | http://bioc.sourceforge.net/ |
| DFKI | 2018 | חדשות עסקיות ומדיה חברתית | 7 | https://dfki-lt-re-group.bitbucket.io/product-corpus/ |
| Gillick | 2016 | מגזינים, חדשות, אתרי אינטרנט ועוד | 89 | <https://arxiv.org/e-print/1412.1820v2> |
| NNE | 2019 | Newswire ( סוכנות חדשות ) | 114 | https://github.com/nickyringland/nested\_named\_entities |

בניגוד לרוב בסיסי הנתונים בטבלה, BC5CDR חריגה מכיוון שהיא מכילה תיוג למידע טקסטואלי שמוגדר בתחום ספציפי. כמובן יש בסיסי נתונים נוספים כאלה. במקרים רבים נושא המידע המתויג הוא חלק משמעותי מהיתרון שיש לחברה מסחרית, ולכן בהרבה מקרים או שאין בכלל מידע מתויג בתחומים רבים, או שקיים מידע אבל הוא אינו חשוף לציבור.

כיום קיימים הרבה שירותים המכילים מודלים מאומנים אשר ניתן להשתמש בהם. ניתן חלק אותם לשתי קבוצות:

* כלים שפותחו על ידי האקדמיה: StanfordCoreNLP, OSU Twitter NLP, Illinois NLP, Gimli ועוד
* כלים שפותחו על ידי התעשייה: spaCy, NLTK, OpenNLP, AllenNLP ועוד

ניתן להשתמש בכלים אלה כמו שהם, וניתן להיעזר בהם כדי לייצר מודלים מתקדמים יותר , גם עבור נושאים שונים מאלו שעבורם נוצרו. חלק מכלים אלה מגיעים לתוצאות מיטביות ( state of the art ) כפי שיפורט בהמשך.

# קשיים

חילוץ ישויות הוא בעיה מורכבת. להלן נפרט חלק מהבעיות המרכזיות:

* יותר מקטגוריה אחת רלוונטית. לדוגמא: נניח שאנחנו עובדים על קורות חיים. נרצה לתייג הן מקומות לימוד והן מקומות מגורים. כעת נניח כי כתוב: "למדתי בתל אביב" האם תל אביב זה מקום או אוניברסיטה?
* עלול להיות מקרה של חפיפה, למשל: "Ron and Daniel Smith" במקרה זה, בהגדרת ישות של שם נצפה לקבל גם את “Ron Smith” וגם “Daniel Smith”.
* יש מספר דרכים להעריך איכות מודלים בתחום זה, חלקן יוצגו בהמשך, ומכאן שלא תמיד קיים מודל "נכון" יותר.
* סיווג הערך מסתמך בצורה משמעותית על התוכן שסובב אותו, זוהי בעיה מורכבת ובה נתמקד. בעיה זו יוצרת 2 בעיות מרכזיות:

## תלות בשפה

הבעיות העיקריות הנובעות מתלות בשפה הן:

* + ייצוג מידע. מידע טקסטואלי לא יכול להיכנס לאף מודל, נעסוק בנושא זה בצורה מעמיקה יותר בהמשך, אבל נדגיש שהשלב הראשון יהיה לייצר ייצוג מספרי לטקסט, ובמעבר בין השפות יש צורך ביצירת ייצוג חדש.
  + מודל מבוסס חוקים ידרוש ידע מעמיק במבנה השפה. גם בשפות שמגיעות ממשפחות דומות, כמו למשל עברית וערבית ששתיהן שייכות לשפות השמיות, יש הבדלים מהותיים. מעבר בין שפות ממשפחות שונות דורשים חוקים דקדוקיים שונים מאד, ולכן לא ניתן לצפות שמודל שעובד עבור שפה מסוימת יעבוד גם עבור שפה אחרת. גרוע מכך, יצירת מודל למספר שפות ביחד עלולה לייצר ביטול הדדי וקריסת המודל עבור שניהם.
  + מודל לחילוץ ישויות מנסה להבין מהי ישות בהסתמך על המילים הסובבות אותה. קיימות מילים שאין להן תרגום בין שפה לשפה. באופן כללי יותר, קטגוריה של חילוץ ישויות מסתמכת באופן משמעותי מאד על תלות סמנטית ועל התוכן העוטף את המילה, שני המאפיינים האלו משתנים בין שפות שונות.
  + אם ננסה להגדיר ווקטור מייצג לכל מילה, כפי שיפורט בהמשך, באמצעות פיצ'רים שנגדיר כמשמעותיים, עלולים להתגלות הבדלים מהותיים בין שפות שונות. דוגמה טובה היא שבשפה האנגלית נצפה שמילה שמתחילה באות גדולה נותנת לנו מידע רב, וניתן לצפות שמודל חילוץ ישויות ייתן משקל רב לנושא זה בקטגוריות כמו מיקום או שם, אבל בשפה העברית לא קיימת לכך מקבילה.

## תלות בתחום

הבעיות העיקריות הנובעות מתלות בתחום הן:

* + מודל מבוסס חוקים:  
    יידרש ידע מעמיק בתחום הספציפי. למשל, מודל שמצליח להבחין בכתבות ספורט, יכול להיעזר בישויות של ענפי ספורט ובעזרת חוקים שיוגדרו שיותאמו להם, אך זה לא יתרום בניסיון לקטלג טוקנים כמחלות במאמרים רפואיים.
  + מודל סטטיסטי:

המודל "יבין את העולם" לפי המידע שהוא הכירת כלומר המידע עליו הוא אומן.

בתחומים מסוימים ובבסיסי נתונים מסוימים יש מבנה טקסטואלי מסוים יחסית אחיד, ולכן מודל סטטיסטי שאומן על קורות חיים "יבין" את העולם כפי שהוא מוצג בקורות החיים, אשר מכילים מבנה יחסית קבוע. כאשר ננסה לקחת את אותו המודל לתחום אחר, למשל סיווג שירים לפי סגנונות, המודל עלול לפספס בגדול.

# שיטות מדידה

ניתן למדוד בעזרת השוואת הפלט של המודל אל מול סיווג אנושי. ניתן לחלק את שיטות ההשוואה לשתיים מרכזיות: השוואה מדויקת, השוואה מקלה ( relaxed match ).

## השוואה מדויקת

המודל נדרש להצליח להבחין בצורה נכונה גם בסיווג וגם בגבול הערך.

כלומר המודל צריך להבין שבמשפט “I live in New York” –

סיווג - "New York" היא מקום

גבול הערך - "New York" ולא רק " New " או "York" או בכלל “in New York”.

בהינתן מידע לצורכי אימון, נקרא לערכים ולקטגוריות שלו, שהן הנכונות, המידע האמיתי.

נתייחס לזה על פי החלוקה הבאה:

* + FP - False Positive – ישות שחזרה מהמודל אבל לא הייתה קיימת במידע האמיתי.
  + FN - False Negative – ישות שלא חזרה מהמודל אבל הייתה קיימת במידע האמיתי.
  + TP - True Positive – ישות שחזרה מהמודל והייתה קיימת במידע האמיתי.

בעזרת האפיונים האלו מגדירים את המדדים הסטטיסטיים הבאים:

להלן דוגמא מעולמות הסיווג למשמעות שלהם ולהבדל ביניהם:

נניח שנרצה לסווג 100 מיילים לספאם ולא ספאם, ונניח שהחלוקה האמיתית היא:

30 מיילים - לא ספאם.

70 מיילים - כן ספאם.

כעת בהינתן מודל המסווג את 100 המיילים נשתמש במדדים כדי להעריך את איכות המודל.

נניח והמודל סיווג 60 מיילים כספאם ( כולם באמת ספאם ) ו 40 מיילים כלא ספאם.

– כמות הפעמים שהמודל סיווג נכון מייל כספאם בייחס לכמות כל מיילים שהם ספאם – כלומר

הבעיה במדד מסוג זה, שאם למשל המודל יחזה תמיד לכל מייל שהוא ספאם, מדד ה יהיה תמיד 100% הצלחה בעוד שברור שמודל הוא מאד לא מוצלח. כלומר המדד הגבוה אינו משקף במקרה זה את איכות המודל.

– כמות הפעמים שהמודל סיווג נכון לספאם בייחס לכל המיילים במאגר – כלומר

מדד נוסף חשוב אשר משלב אותם נקרא balanced F-score:

מכאן ניתן להגדיר שני מדדים נוספים:

* macro-averaged F-score – נחשב מדדי בצורה בלתי תלויה על כל הקטגוריות. כעת ניקח את התוצאות ונחשב את הממוצע.
* micro-averaged F-score – נסכום את ה FP של כל הקטגוריות ונקרא לזה FP, נסכום את ה TP של כל הקטגוריות ונקרא לזה TP, נסכום את ה FN של כל הקטגוריות ונקרא לזה FN, כעת נשתמש בתוצאות אלו כדי להוציא F-score

## השוואה מקלה

בשיטת השוואה זו, ישות תסווג לקטגוריה הנכונה, גם אם הגבולות שלה אינם מדויקים, אבל רק במידה שיש חפיפה עם הגבולות האמיתיים של הישות.

לדוגמא:

“Yossi Benayoun lives in Israel”

אם המודל יציג ש “Yossi Benayoun lives” יקוטלג כ Person.

יש כאן טעות בגבול, מכיוון ש “lives” לא היה אמור להיות כאן, אבל יש חפיפה עם הגבולות האמיתיים ולכן יתקבל בהשוואה המקלה.

בנוסף ישות שגבולותיה הוגדרו נכון, אבל סווגה כקטגוריה לא נכונה, תתקבל בשיטת מדידה זו כהצלחה.

כלומר “Yossi Benayoun” סווג לקטגוריה אחרת, אבל כן הוגדר ביחד אותה ישות.

# דרכי פתרון מבוססות חוקות

פתרון אפשרי של הבעיה הינו הגדרה ידנית של אוסף של חוקים. פתרון זה מתרכז בעיקר בהגדרת דקדוק וחילוץ תחבירי של תבניות הקשורות למבנה השפה. מכאן שצורה זה תלוית תחום מכיוון שיש צורך באחידות מסוימת בדפוסים התחביריים.

Quimbaya [9] - הציע גישה מבוססת מילון לבעיית חילוץ הישויות ברשומות בריאות אלקטרוניות. הוכח בניסויים כי גישה זאת משפרת את ה Recall אך משפיעה בצורה מוגבלת על ה Precision.

לאורך השנים נוצרו הרבה מערכות המבוססות בעיקר על כלליים סמנטיים ותחביריים לזיהוי ישויות.

מערכות מבוססות חוקים עובדות טוב מאד כאשר הלקסיקון שלם.

עם זאת, בשל כללים ספציפיים לתחום ומילונים לא שלמים, נצפים לרוב לדיוק גבוה ב Precision ודיוק נמוך ב Recall כלומר המודל נוטה לטעויות מסוג FN ביחס ל FP.

בנוסף לא ניתן להעביר את המערכת והחוקים שנוצרו עבורה לתחום אחר ובפרט לשפה אחרת ויהיה צורך לבנות הכול מחדש בכל פעם.

# דרכי פתרון בעזרת למידה מבוססת פיצ'רים

גישה זו מבוססת על שימוש בשיטות תיוג מפוקחות. חילוץ ישויות יוגדר כסיווג רב קטגוריאלי או תיוג רצפים. בהינתן מידע טקסטואלי נגדיר את הפיצ'רים הרלוונטיים, ובעזרתם נמיר את הטקסט לווקטור. כעת נוכל להשתמש במודלים מפוקחים של למידת מכונה. למעשה אלגוריתמים אלו ילמדו את דפוסי ההתנהגות של המידע הטקסטואלי אל מול התיוגים, וכך בהינתן מידע חדש, יחזו את התיוגים הרלוונטיים.

בחירת הפיצ'רים היא נקודה קריטית בתהליך, וכך לדוגמא ניתן להגדיר כל מילה על ידי ערכים בוליאניים או מספריים. פיצ'רים ברמת מילה, ברמת המסמך, ברמת אוסף מסמכים, למשל כלל המידע, נמצאים בשימוש רחב במערכות לחילוץ ישויות. בהמשך העבודה אתעמק יותר בנושא זה. מערכות רבות המירו כך את הטקסטים לווקטורים והחילו על התוצר שיטות חילוץ ישויות מפוקחות, לדוגמה:

* Hidden Markov Models (HMM)
* עצי החלטה
* Maximum Entropy Models
* Support Vector Machines (SVM)
* Conditional Random Fields (CRF)

Bikel [10] - הציע את המערכת חילוץ הישויות הראשונה המבוססת על HMM שנקראה IdentiFinder ושימשה עבור חילוץ שמות, תאריכים, ביטויי זמן וכמויות. בנוסף Szarvas [11] - פיתח מערכת חילוץ ישויות רב לשונית המבוססת בין השאר על עצי החלטה.

שיטות מסוג זה השיגו תוצאות טובות.

## אלגוריתם CRF

CRF - Conditional Random Field - הוא מודל הסתברותי המשויך למודלים של למידת המכונה המפוקחים. נעשה בו שימוש רחב במקרים רבים כמו עיבוד תמונה, ביואינפורמטיקה, עיבוד שפה טבעית ועוד. ניתן להשתמש ב CRF בצורה עצמאית כדי לפתור את בעיית חילוץ הישויות. CRF מתחשב בהקשר.

מודל ה CRF יאומן על מנת לחזות וקטור בהינתן משפט

צורת החישוב הינה [2]:

את הניקוד ( SCORE ) מחשבים כך:

– סיכוי של מעבר מקטגוריה לקטגוריה – כלומר סיכוי המעבר מהקטגוריה שאליה חזינו את התיוג למילה הקודמת לקטגוריה הזו.

- מייצג עד כמה המודל "בטוח" שהמילה במקום ה i שייכת לקטגוריה .

בשלב האימון ממקסמים את ה LOG של הסיכוי של הסיווג לקטגוריה הנכונה -

לאורך השנים נוצרו פתרונות מתקדמים יותר המבוססים על רעיון זה, למשל Krishnan and Manning [12] הציעו גישה דו שלבית המתבססת על שני מסווגי CRF כך שהמסווג השני מקבל את הפלט של המסווג הראשון.

נדגיש כי מערכות חילוץ ישויות המבוססות CRF היו בשימוש רב במספר רב של תחומים, כגון: מידע רפואי וציוצים.

# ייצוג מידע טקסטואלי

אחד מהדברים המשותפים לכל המודלים המוזכרים בעבודה זו בנושא בינה מלאכותית הוא העובדה כי מודלים אלו אינם יודעים להתמודד אם אותיות ומילים אלא רק עם מערכי מספרים. מכאן שבכל משימה בעולם עיבוד שפה טבעית יהיה עלינו להמיר את הטקסט שמגיע כקלט למבנה מתמטי מסוג מטריצה, או במקרה הפרטי ווקטור, של פיצ'רים שאיתם נעבוד. הפיצ'רים יכולים לייצג מספר דברים כדוגמת מילים ובסעיף זה נתייחס לחלק מהאפשרויות העיקריות.

נושא ההמרה לפיצ'רים קריטי גם במודלים "רגילים" של למידת מכונה וגם ברשתות עמוקות.

ניתן לחלק את הנושא לשתי שיטות עיקריות:

- קידוד מבוסס תדירות ( Frequency based Embedding )

- קידוד מבוסס שיטות סטטיסטיות

## ייצוג מבוסס תדירות

תחילה יש צורך להגדיר מילים בצורה וקטורית, אלגוריתם בשם one-hot encoding יגדיר ייצוג חד-חד ערכי לכל מילה. האלגוריתם בודק את כמות המילים הקיימות במאגר המידע, ומייצר ווקטור בגודל זה לכל מילה. כל מילה תוגדר אחרת על ידי ווקטור שבו כל הערכים הם 0 למעט ערך אחד שיהיה 1.

למשל, בהינתן מאגר המילים בגודל 10.,תהי המילה "מסמך" באינדקס מספר 9, והמילה "אדום" באינדקס מספר 10. אז צורת הייצוג שלהם תהיה:

מסמך –

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

אדום -

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

היתרון הגדול בצורה זו הוא הפשטות שלה.

אבל לצורת ייצוג זאת ישנם מספר חסרונות מובהקים:

* מאגרי מידע בעולם האמיתי נוטים להיות גדולים מאד, מה שעלול לייצר ווקטורים עצומים שידרשו משאבים גדולים וזמן תפעול משמעותי. במקרה של מאגר מילים בגודל 100k ומשפטים בגודל של חמש מילים, כל משפט עדיין ייוצג על ידי ווקטור בגודל 100 אלף למרות שממדו "האמיתי", כלומר החלק הלא מאופס בווקטור, יהיה לכל היותר בגודל 5.
* צורת ייצוג זו מאבדת את המידע שניתן לחלץ מהסדר בו מופיעות המילים במשפט.
* למילים מעצם היותן חלק מטקסט נרחב יש משמעות מסוימת, בשיטה הזו היא לא מיוצגת כלל.

כלומר כל הווקטורים המייצגים את המילים ניצבים זה לזה למרות שבהסתמך על המשמעות הטקסטואלית.

כאשר משתמשים בשיטה זו לבד, מקובל בדרך כלל להעביר את המידע מספר תהליכים ראשוניים כדי לנסות לצמצמם הטיות שונות של המילה. דוגמא טובה היא שבמקרים מסוימים לא נרצה להפריד בין "למדתי" ל"למד", הרי בדוגמא שהייתה כאן אין כל קשר בין המילים, אך בשיטות כמו ה tokenize שהוצגו לעיל ניתן לנסות לפתור את זה.

עבור ייצוג מקטעים של טקסט ניתן להשתמש בצורת ייצוג הנקראת B.O.G. ( bag of words ).

בשיטה זו יוגדר ווקטור מייצג למשפט \ מסמך בגודל קבוע ובהתבסס על המילים המופיעות בו.

יש מספר שיטות להגדרה זאת כדוגמת count vector שיוצג בסעיף הבא.

### Count Vector

הווקטור ייוצג בהתאם למספר ההופעות של מילה.

דוגמא:

נניח ונתונים ארבעת המשפטים הבאים:

1. היום יום ראשון ולכן יצאתי לעבודה
2. היום יום שישי ולכן נשארתי בבית
3. היום יום חג ולכן לא יצאתי לעבודה
4. היום אין חג וגם לא יום שישי וגם לא יום שבת ולכן יצאתי לעבודה

כל מאגר המילים שלנו כרגע הוא:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| וגם | שבת | אין | לא | חג | בבית | נשארתי | שישי | לעבודה | יצאתי | ולכן | ראשון | יום | היום |

ובעזרת אלגוריתם bag of words נייצג את ארבעת המשפטים באופן הבא:

1. [1,1,1,1,1,1,0,0,0,0,0,0,0,0]
2. [1,1,0,1,0,0,1,1,1,0,0,0,0,0]
3. [1,1,0,1,1,1,0,0,0,1,1,0,0,0]
4. [1,2,0,1,1,1,0,0,0,1,2,1,1,2]

### TF-IDF

בעיה משמעותית שעלולה להיווצר מהגישה המוצגת בסעיף הקודם היא תיעדוף יתר למילים חסרות משמעות מיוחדת שמופיעות הרבה בשפה כדוגמת “and” , “the” , “to” ועוד. לכן הגיוני לרצות לנרמל את הייצוג כך שתהיה התחשבות בדוגמאות כאלה.

בהינתן מסמך ומילה המופיעה בו ניתן להגדיר:

*כאשר:*

TF – מספר הפעמים שהמילה מופיעה במסמך לחלק למספר המילים המופיעות במסמך.

t - מספר הפעמים שהמילה מופיעה במסמך

T - מספר המילים במסמך

N - מספר המסמכים

n – מספר המסמכים שהטוקן t נמצא בהם

כמו כן נגדיר:

לצורך הדוגמא, נגדיר שני מסמכים A, B המורכבים ממשפט בודד והם יהיו כל המסמכים שיגדירו את הקורפוס:

A - “Italy towns want to pay you $33K to move in”

B - “I moved to a tiny village in Japan to experience a fading way of life”

מכאן ש:

TF(to, A) =

TF(to, B) =

IDF(to) = = = 0

TF\_IDF(to,A) =

TF\_IDF(to,B) =

TF(Japan, B) =

IDF(Japan) =

TF\_IDF(Japan) =

למעשה מה שניתן לראות מהדוגמה האחרונה זה שהמילה to נענשה על כמות ההופעות שלה בדוגמאות השונות, בעוד שהמילה "Japan" למרות שהופיעה רק פעם אחת, קיבלה ציון גבוה יותר.

אינטואיטיבית, ניתן להסתכל על דוגמה זו ובאמת להבין שהמילה to לא נותנת הרבה מידע.

הייצוג הסופי יהיה בעזרת אלגוריתם B.O.G כפי שהוצג בסעיף הקודם.

## distributed representations

כשבני אדם מדברים, בלי קשר לנושא השיחה, המילים עצמן אוגרות מידע רב. אנו יודעים למשל מה הקשר בין "הלך" ל"הלכתי". ואנו יודעים שאם נשתמש במילה "עני" או במילה "רש" זה הינו הך, אבל המודל לא אוגר בתוכו את הידע הזה סתם ככה. אם במשימה שעליה נאמן את המודל יהיו מספיק דוגמאות כאלו, המודל עשוי להבין הקשרים כאלו, אך בפועל, מכיוון שמילים מסוימות נמצאות בשימוש רב יותר מאחרות באופן משמעותי, הסיכוי לכך נמוך.

בניגוד ל one-hot encoding, בצורת ייצוג זאת אנו רוצים שעצם הייצוג יאגור בתוכו מידע, כך שהקשר בין מילים יתורגם למיקומם במרחב.

המפתח לגישה זו הוא שימוש בייצוג צפוף. כלומר ישנה הקפדה על ממד מסוים קבוע וקטן ( בדרך כלל בהרבה ) מכמות המילים.

יש מספר דרכים להתמודד עם בעיה זו:

הגדרה ישירה – ניתן להגדיר בצורה ידנית מהם הפיצ'רים המגדירים מילה \ משפט. למשל האם המילה מתחילה באות גדולה, האם אורך המילה גדול מ 3 אותיות ועוד. כל מילה תיוצג על ידי הווקטור המתאים עבורה, וכל הטקסט יוגדר כמטריצה כך שכל ווקטור מציג את תכונות המילה. לחילופין ניתן להשתמש בשכבה נוספת שתחשב ממוצא לכל הווקטורים. אפשרות נוספת היא שימוש בווקטור המחושב באמצעות סכום הווקטורים.

שיטות סטטיסטיות:

חלק גדול מההתקדמות הגודלה של משימות מעולם ה NLP בעת המודרנית הוא המודלים שנוצרו לאורך השנים ( pre trained models ) לייצוג מידע טקסטואלי. שילוב של אלגוריתמים מתוחכמים, כמויות מידע גדולות מאד וכח חישובי גדול מאד יצרו מודלים ששוחררו בחינם לעולם ואפשרו את הבסיס למשימות רבות בעולם זה.

כיום, נהוג בדרך כלל לקחת את אחד המודלים האלו, ולהשתמש בו באופן ישיר, או, אם יש מספיק מידע ספציפי למשימה, לעשות FINE-TUNING כלומר לעדן או למקד את הקידוד כך שיתאים למידע \ משימה הרלוונטית.

הרעיון המרכזי של השיטה הוא שניתן להפיק ייצוגים סמנטיים למילים באמצעות ניתוח דפוסי ההתרחשות שלהם.

You shall know a word by the company it keeps (Firth, J. R)

### ייצוג מבוסס מילים

#### w2v

Word to vector הינה שיטה סטטיסטית ללימוד יעיל של מילים מדאטה גדול של טקסטים.

עוברים על הטקסט בעזרת חלון בגודל קבוע, כל פעם מגדירים מילה אחת, ומסתכלים על המילים הסובבות אותה, זאת אמרת:

Graphical user interface

Description automatically generated

איור – [17]

כלומר, נרצה לאמן כל מילה מול המילים הסובבות אותה, ובעצם לבצע פעולת חיזוי וסיווג בעזרת רשת נוירונים קצרה של מילה מול מילים אחרות.

ההגדרה מתייחסת לשני אלגוריתמים שונים אך דומים:

* Continuous Bag Of Words ( CBOW ) - המודל לומד לייצג את המילה על ידי ניסיון לחזות אותה בעזרת המילים הסובבות אותה.
* Skip Gram – מודל המנסה לחזות את המילים הסובבות את המילה מתוך המילה עצמה.

בשני המקרים, מגדירים גודל של חלון שאיתו עובדים. לכל מילה מוגדר ווקטור מתאים המאותחל באופן רנדומלי, וכעת, עם התקדמות החלון, המודל יחזה, ובהסתמך על תוצאות האמת, יבצע התאמות, כלומר המודל יתקן את ערכי הווקטור של כל מילה. בסופו של התהליך, אם כמות המידע מספיק גדולה, הייצוג של המילה יהיה משמעותי.

Diagram

Description automatically generated

איור [13]

ניתן להראות את פונקציית המחיר כך, כאשר את הסיכוי עצמו מחשבים באמצעות softmax והמטרה להגיע למקסימום:

מסיבות חישוביות לא בודקים את פונקציית ה softmax אל מול כל המילים במאגר, אחת הדרכים האפשריות להתמודד עם כך היא בחירה קבוצת נציגים רנדומלית כל פעם בנוסף למילים שבאמת סובבות את המילה.

לאחר סיום התהליך לכל מילה יהיה ווקטור מייצג.

מה המובן של ייצוג משמעותי:

מבדיקות שנעשו לאחר שימוש במודל זה, הוצג כי אם לוקחים את הווקטור המייצג של המילה "KING", מחסרים ממנו את הווקטור המייצג של המילה "MAN" ומחברים לתוצאה את הווקטור המייצג של המילה "WOMAN" – מתקבל ווקטור קרוב מאד לווקטור המייצג של המילה "QUEEN".

נקודה נוספת ומשמעותית היא שניתן למצוא קירבה בין מילים על ידי זוויות, וכך לראות עד כמה המילים דומות.

דמיון =

ככל שהזווית קטנה יותר הדמיון גדול יותר .

ניתן גם להיעזר במדידות במיון בדרכים שונות כגון מרחק ווקטורי, אבל הזוויות נחשבות כאינדיקציה טובה מאד עבור דמיון.

בנוסף הייצוג במרחב שומר על קשרים בין ערכים שונים לדוגמא:

Chart, scatter chart

Description automatically generated

איור – [18]

#### GloVe

Glove - Global Vectors

בניגוד ל w2v בשיטה זו אין הסתמכות רק על סטטיסטיקה לוקאלית, כלומר היחס של מילה רק למילים שסביבה אלא יש שימוש גם בסטטיסטיקה גלובאלית.

כלומר המודל משתמש במטריצת הגורמים הגלובאלית ( global matrix factorization ) ובהקשרים מקומיים המופקים באמצעות חלון המגדיר את מידת הלוקאליות (local context window )

עבור מאגר מילים המכיל V מילים שונות, מטריצת המילים המשותפות תהיה בגודל V\*V, כך שבשורה ה i ובעמודה ה j יוצג כמה פעמים המילה i הופיע באותו חלון עם המילה j.

דוגמה למטריצת המילים המשותפות ( co-occurrence matrix ) עבור המשפט בעת שימוש בחלון בגודל 1

“the cat sat on the chair”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chair | On | sat | Cat | the |  |
| 1 | 1 | 0 | 1 | 0 | The |
| 0 | 0 | 1 | 0 | 1 | Cat |
| 0 | 1 | 0 | 1 | 0 | Sat |
| 0 | 0 | 1 | 0 | 1 | On |
| 0 | 0 | 0 | 0 | 1 | Chair |

חישוב דמיון של מילים יכול להיעשות באמצעות שלוש מילים בכל פעם:

Graphical user interface, text

Description automatically generated

איור [14]

חישוב הסיכוי ( ) נעשה על ידי לקיחת כמות ההופעות המשותפות של k ו word באותו חלון וחלוקתה בכמות ההופעות הכלליות של k בכל הקורפוס.

המטרה של של GloVe הוא ללמוד ייצוג ווקטורי של מילים כך שהמכפלה הפנימית שלהם ( dot product ) תהיה שווה ללוג של פונקציית הסיכוי של המילים בתוך מטריצת המילים המשותפות.

על פי ספריית glove עבור המילה morning הווקטור המייצג שלה יהיה:

[

0.1210, 0.1012, -0.1806, 0.2005, 0.3468, -1.0604, -1.1647, 0.6439,

-0.3352, -0.7904, -0.5596, -1.0967, -0.1495, 0.3683, 0.5777, 0.2775,

-1.3737, -0.2331, -1.2181, -0.4640, 0.9180, 1.3100, 0.5937, 0.0471,

0.6540, -1.0776, 0.3306, 1.0118, 0.2792, 0.1584, 3.1537, 0.4544,

-0.1560, 0.3337, 0.0330, -0.4056, 0.4131, -0.2861, 0.3155, 0.3254,

-0.4461, 0.3640, -0.5311, -0.3834, 0.6394, -0.0799, 0.0463, -0.2036,

0.4200, 0.2557

]

### ייצוג מבוסס אותיות

יש מספר בעיות עם ייצוג ברמת מילה, העיקריות שבהן:

* כיצד יש להתייחס למילים שאינן נמצאות בקורפוס האימון
* מאבדים מידע של חלקי מילים

ניתן להשתמש ב CNN וב RNN עבור ייצוג ברמת אות כדוגמת:

Diagram

Description automatically generated  
איור [4]

RNN – כפי שיפורט בהמשך מתמודד בצורה מוצלחת עם סדרות, וניתן להתייחס לכל רצף אותיות כסדרה

CNN - נהוג להשתמש ברשת קונבולוציה חד ממדית ( 1D-CNN ) לטובת ייצוג מילים ברמת האותיות.

הרעיון המרכזי הוא לסרוק בעזרת מספר סורקים כל מילה – אות אחרי אות. כל סורק יתמקד במספר אותיות בכל פעם ויחלץ מהן מידע רלוונטי. בסופו של התהליך מילה תיוצג בעזרת המידע שחולץ מכל הסורקים.

### טוקנים

במקום לחלק למילים, או לאותיות, ניתן לקחת משהו חלקי.

WordPiece הוא אלגוריתם לייצוג טוקנים. הרעיון המרכזי שלו הוא לייצר מילון, שיאותחל מכל התווים הבודדים בקורפוס, ואז בצורה איטרטיבית להוסיף לו את שילובי התווים השכיחים ביותר.

דוגמא, יהי הקלט :

White House prepares fightback in Covid disinformation war!

לדוגמא בעזרת BertTokenizer ( מספריית transformers והמודל של bert-base-cased ) הטקסט יחולק בצורה הבאה:

['White', 'House', 'prepares', 'fight', '##back', 'in', 'Co', '##vid', 'di', '##sin', '##formation', 'war', '!']

לכל טוקן יש ווקטור מייצג משלו.

"##" – משמעותו שהוא חלק ממילה והמשך של טוקן אחר מהמילה.

המילה " disinformation " מחלוקת לשלושה טוקנים שונים.

לכל טוקן מתוך השלושה האלה יש ווקטור מייצג מתאים.

נשמור על ההפרדה, משמע בדוגמא למעלה יש קלט מסוג משפט עם שמונה מילים, ומייצגים אותו באמצעות מטריצה שאורכה כגודל הווקטור המייצג של כל טוקן ואורכה בגודל 13, ואת זה נעביר למודל.

אפשרות נוספת היא לקחת את הממוצע של שלושת הווקטורים האלה, ולהגיד כי זהו הווקטור המייצג של המילה.

# דרכי פתרון מבוססות רשתות נוירונים

## מבוא

רשתות נוירונים הן אוסף של מודלים מתמטיים שצברו פופולריות רבה בשנים האחרונות. [5]

ההשראה לתהליך מקורה בתהליכים ביולוגיים או קוגניטיביים, אך בפועל רשתות נוירונים מלאכותיות משמשות כהכללה ושכלול של התהליך הבסיסי של לימוד מכונה, המקביל לתהליך החישובי המתבצע ברשת חד שכבתית. הכללה זו מאפשרת תחכום מתמטי וחישובי שקשה להשיג במודלים אחרים.

דוגמה לרשת מסוג fully-connected.

Diagram

Description automatically generatedאיור – [19]

דוגמא לנוירון בתהליך הפעפוע קדימה ואחורה.

Diagram

Description automatically generated

אחד מהיתרונות של רשתות נוירונים הוא העובדה כי העמקת הרשת מאפשרת מציאת קשרים מורכבים מאד, ויש בכוחה לחבר יחד פונקציות לינאריות שונות לכדי פונקציה מורכבת הרבה יותר.

דברים שבאלגוריתמים אחרים מצריכים ייצור של פיצ'רים נוספים, רשתות נוירונים ימצאו לבד את הקשרים.

דוגמא קלאסית היא חישוב פונקציית ה XOR באמצעות מכונה לומדת.

כלומר:

|  |  |  |
| --- | --- | --- |
| Y | X2 | X1 |
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |

באלגוריתמים לומדים "רגילים" כדוגמת logistic classification לא נוכל להגיע ל 100% הצלחה אלא אם נוסיף פיצ'ר נוסף המקשר בין פיצ'רים אחרים, כי עבור XOR לא קיימים W1, W2 המקיימים Y=XOR(X1,X2)=W1\*X1+W2\*X2 אבל ברשת רב שכבתית, למעשה ברשת בעלת שכבה חבויה בודדת, נוכל בקלות.

באופן כללי יותר, רשת נוירונים יכולה לבטא בעזרת שרשור של פונקציות פשוטות, כל פונקציה שהיא.

# ארכיטקטורת CNN

CNN - Convolutional Neural Network

CNN היא ארכיטקטורה של רשתות נוירונים המפורסמת בשימוש הרב הנעשה בה בעולמות עיבוד התמונה. ניתן להסתכל על תמונה כמערך דו-ממדי של פיקסלים לתמונות שחור-לבן או תלת-ממדי לתמונות עם צבע.

היעד הוא להשתמש במספר שכבות כך שיזהו דפוסים דומים כדוגמת קווים, עיקולים, צבעים.

Diagram

Description automatically generatedאיור [9]

CNN מורכב משלושה סוגי שכבות: convolutional, Pooling, fully connected  
שכבות הקונבולוציה מטרתן לסרוק את הקלט ולבדוק לכל איזור ( קבוצת פיקסלים קרובה ) כמה הם שייכים לדפוס שאותה שכבה בודקת. ניתן להשתמש בכמות סורקים וכך להרחיב את מרחב החיפוש.

איור להמחשה [17]

Shape

Description automatically generated

# ארכיטקטורת RNN

ניתן להסתכל על בעיית חילוץ הישויות כבעיית תיוג רצפים. לפיכך, ההתמודדות עם בעיה זו דורשת רשתות המשמרת תלויות טמפורליות, שאינן משתמרות ברשת מסוג fully-connected. אחת מהדרכים לפתרון בעיה זו היא רשת RNN שהיא בעלת יכולת לאחסן בזיכרון קשרים לכל החלקים האחרים ברצף ומאפשרת התמודדות עם בעיית חילוץ הישויות.

A picture containing timeline

Description automatically generated

בניגוד לרש

תות נוירונים "רגילות", כאן נשמור על אותן המשקולות לאורך כל הדרך. המודל יקבל רצף של טוקנים ( X ), כל אחד כווקטור, יעבור טרנספורמציה ליניארית בעזרת מטריצת משקולות מתאימה ( ). פונקציית המצב hn תשמור מידע על כל מה שהיה בסדרה עד הרגע הנוכחי ובכניסה לכל מצב באמצעות טרנספורמציה ליניארית עם מטריצת המשקולות שלו ( ). את התוצאה נעביר נחבר ונוסיף ווקטור משקולות חופשיות ונעביר דרך פונקציית אקטיבציה לא ליניארית, בדרך כלל relu \ tanH \ sigmoid.

את התוצאה נעביר למצב הבא, hn+1, ונוציא גם כפלט, בדרך כלל לאחר טרנספורמציה ליניארית נוספת ומעבר בפונקציית אקטיבציה מסוג SOFTMAX שבעצם ניתן להשתמש בו על מנת לסווג את התוצאה לכל אחד מהקטגוריות שהוגדרו.

Diagram, schematic

Description automatically generated

איור – [20]  
H – ערך המצב - מוגדר על ידי כל השלבים שהגיעו עד כה   
X – ערך הקלט הנוכחי

השגיאה ברשת RNN נמדדת כסכום השגיאות בסדרה, ופעולת עדכון המשקולות נעשית באמצעות אלגוריתם Backpropagation Through Time, אשר מחשב את התיקון של כל המצבים ביחס לכל המצבים שקדמו להם ורק אז מתקן את המשקולות.

*דוגמא לתיקון שגיאות של אחת ממטריצות המשקולות*

החלק המסומן בצהוב הוא בעייתי, מכיוון שבשביל לחשב אותו נשתמש בכלל השרשרת והמשמעות היא שככל שהמרחק בין k ל t יגדל ( משמע המרחק בין המילים ) כך כמות פעולות ההכפלה ילך ויגדל.

ניתן להסתכל על בעיית חילוץ הישויות כבעיית encoder-decoder, בדומה למה שקורה במשימות תרגום.

משמע, יש חלק ראשון שמקודד את המידע, וחלק נוסף שמפענח את המידע המקודד לכדי סיווג:

Graphical user interface, text, application, email

Description automatically generated

איור [4]

# ארכיטקטורת LSTM

מחקרים מצאו כי למרות היכולת התיאורטית של רשתות RNN ללמוד תלויות טמפורליות ארוכות טווח, במציאות רשתות RNN סובלות מבעיות בתהליך תיקון המשקלות בעת הפעפוע לאחור. תופעה זו מכונה בעיית השיפוע הנעלם - “gradient descent”. מצב זה קורה כאשר הנגזרת של השגיאה ביחס למשקולות שואפת ל 0 או לאין סוף ועקב כך הלמידה נהיית לא אפקטיבית.

ארכיטקטורת ה LSTM ( LONG SHORT TERM MEMORY ) עוצבה על מנת להתמודד עם בעיה זו באמצעות שימוש בתאים הנקראים חוסמי זיכרון - memory blocks - בשכבות הביניים. תאים אלו מכילים ארבעה חלקים: שער קלט - input gate, שער פלט - output gate, שער שכחה - forget gate ותא זיכרון - memory cell,

עם התפתחות המחקר במודלים מבוססי RNN, טכניקות תיוג הרצפים השתפרו בצורה מהירה ותרמו רבות למשימות כמו חילוץ ישויות.

תיאור השערים:

forget gate layer

Input gate layer

Candidate

Cell state

output gate layer

Diagram

Description automatically generated

הבדלים בין RNN ל LSTM מבחינת חישוב:

Diagram

Description automatically generated

ניתן לראות בצורה ברורה מהתמונה כי אימון מודל מבוסס LSTM כבד מבחינה חישובית בצורה משמעותית ממודל RNN. לכן פותחו מודלי ביניים כמו GRU שהוא גם מודל רשת סדרתי מבוסס שערים כמו LSTM, אך מכיל פחות שערים ופחות חישובים.

# ארכיטקטורת Bidirectional LSTM

זיהוי נכון של ישות במשפט תלוי בהקשר של המילה. גם החלק שלפני המילה משפיע אבל גם החלק שאחריה. Bidirectional LSTM ( מבוססים על Bidirectional RNN ) עוצבו על מנת לקודד כל אלמנט ברצף, תוך כדי לקיחה לחשבון את החלק הימני והשמאלי של התוכן, מה שהופך את ארכיטקטורה זאת לאחת מהמובילות כדרך פתרון לבעיית חילוץ הישויות. ארכיטקטורה זאת מכילה שני שלבים מרכזיים:

* רשת LSTM אחת שרצה משמאל לימין ומחשבת ייצוג של התוכן השמאלי
* רשת LSTM אחת שרצה מימין לשמאל ומחשבת ייצוג של התוכן השמאלי

הפלט של רשתות אלה מאוחד על מנת לייצר ייצוג מלא של הקלט. ארכיטקטורת Bidirectional LSTM הוכיחה יכולות מרשימות במגוון משימות בעולם עיבוד שפה טבעית ( NLP ) כגון: תרגום, שאלות ותשובות ובמיוחד במשימת חילוץ הישויות.

תמונה להמחשה:

Diagram

Description automatically generated

איור – [22]

# מנגנון Attention

מנגנוני encoder-decoder, כמו RNN לסוגיו השונים השיגו תוצאות מצוינות לאורך השנים ובמשימות רבות הגיעו לתוצאות הטובות ביותר ( state of the art ) בתקופתם. עם זאת, אחת המגבלות הבולטות ביותר שלהם היא הצורך לקודד את המידע שנצבר לווקטור בגודל קבוע ( H ).

דבר זה מגביל את ביצועי הרשת, במיוחד במקרים של רצפי כניסה ארוכים ( טקסט ארוך בקלט ). דבר זה בלט במיוחד לאורך השנים בעולמות של תרגום טקסט.

Attention הוא רעיון המשחרר את ארכיטקטורת ה encoder-decoder מייצוג על ידי גודל קבוע. הדבר נעשה על ידי שמירת הפלטים במצבי הביניים במקודד ( LSTM כדוגמא ) של כל קלט ואימון מודל שלומד לבחור לאיזה חלק בקלט יש לשים לב ( pay attention – ומכאן השם ). בהמשך מחברים את החלקים הרלוונטיים מתוך הקלטים. הדוגמה שנתתי מגיעה מעולמות התרגום אך אותו דבר בדיוק קורה גם בבעיית חילוץ הישויות.

כאשר אנו מגיעים למילה חדשה, במקום לקבל רק ווקטור שמייצג את כל המידע שנאסף עד אותו רגע, כמו ב RNN, המצב החדש יקבל את כל ערכי הפלט של כל מצבי הביניים שעברנו עד רגע זה והמודל ילמד לאיזה חלקים מתוך המידע הזה לשים לב. זה מאפשר לשים דגש על חלקים חשובים ולהתעלם מהשפעת חלקים שלא תורמים ל"הבנת" הטקסט ורק יפגעו בתוצאות.

רעיון מאד דומה עומד בבסיס משימות נוספות בעולמות עיבוד שפה טבעית, עיבוד תמונה, זיהוי קול ועוד.

## Self-attention

לצורך הסבר המנגנון וחשיבותו, ניעזר בדוגמה הבאה:

" The animal didn't cross the street because it was too tired"

נשאלת השאלה, למי שייכת המילה it במשפט זה, לרחוב או לחיה?

למרות שלרוב האנשים הקוראים משפט זה התשובה ברורה, לאלגוריתמים קשה מאד להצליח להבין את נושא השייכות. מנגנון ה self-attention נועד להתמודד עם הקושי הכרוך ביצירת קשרים אלו.

בעיבוד של כל מילה, כלומר כל אלמנט במשפט בקלט, המכניזם של self attention יאפשר לו לבדוק את כל המיקומים האחרים בקלט על מנת לחפש רמזים שיעזרו לו לקודד בצורה מיטבית את המילה. בדיקת כל המיקומים עלולה להיות יקרה וזאת הסיבה לכך שבתוכנות המממשות את המנגנון הזה נוטים להגביל את גודל הסדרה.

בדומה לאופן בו מצבי הביניים בשכבות החבויות עוזרים לרשתות RNN לקודד מידע המייצג את ערכי העבר, self-attention הוא המקביל לכך בעולמות ה transformers שיוצג בסעיפים הבאים.

Chart, radar chart

Description automatically generated

איור [15]

כעת אפרט יותר איך המנגנון עובד.

ראשית נייצר שלושה ווקטורים מכל אחד מהקלטים של המקודד.

לכל מילה, נייצר Query vector, Key vector, Value vector.

זה נעשה על ידי כפל בין הטקסט, בחלוקתו הייצוגית, במקרה זה מילים, לבין שלוש מטריצות מתאימות שמעודכנות כחלק מתהליך האימון. כלומר מעבירים את הקלט בשלוש טרנספורמציות ליניאריות שונות ושומרים את הפלטים בשלוש מטריצות.

כעת מתבצעות מספר פעולות:

* לכל מילת קלט מחושב ציון אל מול כל מילה אחרת ממילות הקלט. משמעות הציון היא עד כמה נתייחס לכל חלק בקלט בזמן קידוד המילה במקום שלה. ישנן מספר דרכים לחשב את הציון, אחת המקובלות שבהן היא חישוב מכפלה פנימית בין ווקטור הquery לווקטור ה key של כל מילה. כך שכאשר אנחנו מחשבים את ה self-attention במילה במיקום הראשון, הציון הראשון יהיה המכפלה הפנימית בין q1 ל k1, הציון השני יהיה המכפלה הפנימית בין q1 ל k2, וכן הלאה.
* נחלק את התוצאות ב 8 – זה מוביל ליציבות הגראדינטים ( 8 הוא שורש של 64 שהוא גודל הווקטור בו נעשה שימוש במאמר attention is all you need )
* את התוצאה נעביר ב Softmax – כך שכל התוצאות יהיו חיוביות ומסתכמות ל 1.
* הכפלת ווקטור ה value בתוצאות ה Softmax
* כעת נסכום את כל התוצאות

Graphical user interface, PowerPoint

Description automatically generated

איור [15]

התוצאה היא ווקטור שניתן לשלוח קדימה ברשתות נוירונים. במציאות, על מנת לשפר ביצועים נמיר הכל לחישובים על ידי מטריצות.

כלומר נייצר טרנספורמציה ליניארית מהקלט וכך נייצר את שלושת המטריצות האלו.

כל החישובים שהוצגו לעיל ניתנים לביטוי באופן הבא:

## multi head attention

המודל מוגדר כך: עבור מספר ראשי attention שונים, כל המטריצות של ה key, value ו query יהיו שונות בין כל ראש של attention ( טרנספורמר משתמש ב 8 ראשי attention ). כולן מאותחלות רנדומלית וכך לאחר האימון כל אחד מהראשים לוקח חלק בתהליך הקידוד של הקלט ומעביר אותו לייצוג אחר בתת מרחב משלו.

כעת יש לנו 8 מטריצות מסוג Z ( כפי שהוצג בחלק של ה self-attention ), כעת הן יאוגדו אחת לשנייה, ויוכפלו במטריצת משקולות חדשה ומתאימה וזהו.

Diagram, engineering drawing

Description automatically generated

איור – [21]

למנגנון ה multi head attention יש מספר יתרונות:

* הוא משפר את יכולת המודל להתמקד במיקומים שונים.

מודל שפה מבוסס self attention מזהה את חוזק הקשר של כל מילה לכל מילה אחרת בחלון ( האזור עליו פועל ה self attention ).  
ב multi head attention כל חלק ( self attention ) מאומן באופן עצמאי והחיבור מצליח למצוא קשרים מורכבים יותר  
המנגנון נותן לשכבת ה attention מספר תתי מרחבים לייצוג.

## Transformer

מנגנון הטרנספורמר מוזכר לראשונה למיטב ידיעתי במאמר attention is all you need [5].

בהתחלה עבור משימות תרגום ובהמשך למשימות NLP רבות אחרות.

Diagram

Description automatically generated

איור [5]

המנגנון הוא מסוג encoder-decoder ועובד בצורה הבראה:

* הקלט הוא טקסט, ממירים לייצוג ווקטורי באמצעות embedding ( בעזרת word piece לדוגמא )
* לכל ווקטור שהומר מקודדים את המיקום ( יש לכך מספר אפשרויות שעלו לאורך השנים, אבל הרעיון הוא שכחלק מהווקטור המייצג יופיע המידע של המיקום )

החלק של ה encoder מורכב ממספר שכבות:

* multi-head attention - כפי שהוצג בסעיף הקודם
* חיבור ונורמליזציה – חיבור של הקלט עם הפלט של ה multi-head attention עבור יצירת חיבור ישיר של הקלט, ושכבת נורמליזציה שמטרתה לנרמל את הממוצע והשונות
* fully connected layer
* חיבור ונורמליזציה

כל זה מרכיב שכבת encoder אחת, במאמר המקורי משתמשים בשש כאלו, אחת אחרי השנייה.

החלק של ה decoder:

* Masked multi-head attention – דומה ל multi-head attention אבל ב decoder יש צורך לדמות את המציאות בה אין את המידע העתידי ולכן מסתירים אותו, לכאן נכנס החלק שכבר תורגם
* חיבור ונורמליזציה
* fully connected layer
* שכבת multi-head attention – שמקבלת גם את המידע מה encoder
* חיבור ונורמליזציה
* fully connected layer

גם החלק הזה קורה 6 פעמים במאמר המקורי, אחת אחרי השנייה.

ולאחר מכן מתבצע:

* fully connected layer
* softmax

# BERT

BERT - Bidirectional Encoder Representations from Transformers [7] הוא מנגנון מבוסס TRANSFORMER.

מטרת המודל היא לייצר ייצוג ווקטורי לטוקנים כך שיאגרו מידע ומשמעות בצורה מיטבית.

המודל מקבל כקלט כמות טקסט גדולה מאד ומחלק אותה לטוקנים ( word peace ) ועוד מספר תגיות ייחודיות עבורו, ומאומן בעזרת שתי משימות:

* זיהוי המשפט הבא.
* חיזוי מילה ( בכל איטרציה מפלטרים מילים אחרות ).

בעזרת משימות אלו, בדומה ל W2V, נוצר הייצוג של הטוקנים

במאמר המקורי שיצא בסוף שנת 2018 [7] הוצגו שתי ארכיטקטורות: מודל בסיסי ומודל גדול. המודל הגדול מגיע לתוצאות הטובות יותר אך כבד חישובית עד כדי כך שלרוב לא משתמשים בו.

המודל המאומן יקבל מידע טקסטואלי, יחלק אותו לטוקנים ( word peace ) ויחליף כל טוקן בווקטור המתאים לו.

כשהמודל יצא הוא נחשב הטוב ביותר במשימות רבות, וגם היום הוא ודומיו, למשל ROBERTA [8] שמגיע מרעיון מאד דומה, הם הפתרון הדיפולטיבי לבדיקה במשימות NLP רבות.

# שילובים מתקדמים

## RNN + CRF

בעוד ש RNN ישמור את המידע על כל הסדרה עד רגע נתון, הוא לא ישמור את המידע על התיוגים שכבר נעשו. אתן דוגמה מעולם אחר, נניח והינו רוצים לקרוא מילה מתמונה, ונניח והינו משתמשים באיזשהו מודל לחזות כל אות. אם האות הראשונה והשנייה היו i, סביר להניח שאם המודל הציע לאות השלישית כמה אפשרויות בסבירות גבוהה, ואחת מהן היא i, כנראה שכדאי להוריד את רמת הביטחון באפשרות זו, מתוך הידע הקיים על השפה. גם ברשתות ששומרות מידע טמפורלי כמו RNN התוספת של שכבת ה CRF מוסיפה מנגנון שלא קיים באופן טבעי - יחס לקטגוריה ולא רק למידע שנשמר - והתוצאות האמפיריות מראות כי השילוב טוב מהותית מכל מודל בנפרד.

## Bidirectional LSTM+CRF

כפי שמוצג במאמר משנת 2015 [2], בו מוצג הפסואוקוד המתאר איך עובד LSTM+CRF.

Text

Description automatically generated

איור [2]

כלומר, כפי שנשמע בדיוק, יש רשת bi-lstm שאת ה output שלה מפעפעים ל crf, ואז את הפעפוע חזרה עושים בצורה הפוכה, כלומר, תחילה מה CRF ואז ל bi-lstm. זוהי שיטה מאד נפוצה שהגיעה לתוצאות טובות מאד יחסית למתחרותיה לתקופה מסוימת.

מאמר מ 2020 [2] בנושא חילוץ ישויות עבור עולמות הסייבר מציג שימוש נוסף בארכיטקטורה המבוססת על עקרון זה:

ווקטורי הקלט נכנסים לשני חלקי הרשת bi-lstm מייצרים פלט משותף. בפלט זה נעשה שימוש ברשת חד שכבתית ואת התוצאה מעבירים לשכבת crf. Crf משתמשים במידע על תיוג השכן כדי לחזות את התגים הנוכחיים לכל שלב.

שיטות אלו השתמשו ב beam decoding על מנת להשיג חילוץ ישויות מיטבי שנחקר בשיטות כגון maximum entropy classifer ו maximum entropy Markov models (MEMMs) אשר מתמקדים במיקום ברמת משפט במקום מיקום ברמת תוקן.

Crf מנצל את שרשרת התיוגים הטובים ביותר בהינתן קלט מסוג משפט ולא מיקום בודד.

תמונה מהמאמר המדגימה את הארכיטקטורה בה השתמשו במאמר זה:

Diagram, engineering drawing

Description automatically generated

איור[2]

במאמר נוסף, משנת 2017 [3] בנושא חילוץ ישויות בשפה הרוסית, מוצגת הארכיטקטורה הבאה כמיטבית:

השפה הרוסית מורכבת ועשירה מאד, לכן ניתן לצפות ששילוב של bi-lstm ו CRF יאפשר שיפור של תוצאות תיוג הישויות.

בדוגמה שהוצגה במאמר זה, הייצוג הוא ברמת האותיות אשר נכנסות לרשת bi-lstm, ואז ווקטור זה ( ייצוג ברמת אותיות ) מתחבר עם ווקטור ברמת המילה ועובר לרשת Bi-LSTM חדשה. הרשת מחשבת את הציון לסיכוי שרצף מסוים שייך לקטגוריה מסוימת.

Diagram

Description automatically generated

איור [3]

## BERT+CRF

הרעיון העומד בבסיס מודל זה הוא לקיחת מודל BERT מבוסס טוקנים ( word-piece ), ואחריו שכבת CRF. לקלט בגודל n טוקנים, BERT יוציא פלט של רצף טוקנים מקודדים עם שכבות ביניים H. המסווג ייקח כל טוקן מקודד ויגדיר לו ממד קטגוריות. כלומר: כך ש k הוא מספר הקטגוריות. תוצאת הניקוד תהיה: – כלומר לכל טוקן יתקבל הניקוד עבור כל קטגוריה. את התוצאות נעביר אל שכבת ה CRF. הפרמטרים של שכבה זו הם המטריצה של מעברי התיוגים: , כך ש מייצג את הציון של מעבר מתיוג 1 לתיוג 2. מטריצה A כוללת 2 מצבים נוספים שהם תחילת וסוף הרצף.

כך שלקלט של רצף ורצף של חיזוי תיוגים , כשכל שייך לאחת מ K הקאגוריות, ציון הרצף מוגדר באמצעות:

*במאמר משנת 2020 [1] בנושא חילוץ ישויות בשפה הפורטוגזית, הודגם כיצד ניתן להגיע לתוצאות הטובות ביותר באמצעות ארכיטקטורה זאת.*

*תמונה להמחשה:*

Diagram

Description automatically generated

איור [1]

# כיוון מחקר- מודל רב שפה

גם היום עם כל החידושים והיכולות החדשות, עדיין מחלקים את המידע לפי השפות שונות ואין מודל אחד שמאחד מספר שפות ביחד. ליתר דיוק, אין מודל רב-שפתי שמגיע או מתקרב לרמת הביצועים של מודלים חד-שפתיים. מודל כזה הוא עדיין אתגר של התחום.

# סיכום

כמו משימות רבות בעולם, בעיבוד השפה הטבעית חל שיפור משמעותי בשנים האחרונות.

בסמינריון זה הוגדרה הבעיה של חילוץ ישויות, הוצגו דרכי מדידה, קשיים ואפשרויות פתרון מרכזיות שעלו לאורך השנים.

בעבודה הוצגה קשת של פתרונות, החל ממערכות מבוססות חוקים שהיו נהוגות בעבר, ועד לרשתות נוירונים מודרניות וצורת השימוש בהן. כן נידונו דרכי הייצוג של המידע, שהשתנו בצורה משמעות בשנים האחרונות.

עדיין נותרו לא מעט אתגרים, כדוגמת מודל רב שפות, והתמודדות עם כמות המידע הגדולה הדרושה כיום ליצירת מודל חדש מאפס.

# ביבליוגרפיה

1. Souza, F., Nogueira, R., & Lotufo, R. (2019). Portuguese named entity recognition using BERT-CRF. *arXiv preprint arXiv:1909.10649*.‏
2. Arkhipov, M. Y., & Burtsev, M. S. (2017, September). Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition. In *Conference on Artificial Intelligence and Natural Language* (pp. 91-103). Springer, Cham.‏
3. Kim, G., Lee, C., Jo, J., & Lim, H. (2020). Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network. *International journal of machine learning and cybernetics*, *11*(10), 2341-2355.‏
4. Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.‏
5. Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.‏
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).‏
7. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.‏
8. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.‏
9. *A. P. Quimbaya, A. S. Múnera, R. A. G. Rivera, J. C. D. Rodríguez, O. M. M. Velandia, A. A. G. Peña, and C. Labbé, “Named entity recognition over electronic health records through a combined dictionary-based approach,” Procedia Comput. Sci., vol. 100, pp. 55–61, 2016.*
10. *A. P. Quimbaya, A. S. Múnera, R. A. G. Rivera, J. C. D. Rodríguez, O. M. M. Velandia, A. A. G. Peña, and C. Labbé, “Named entity recognition over electronic health records through a combined dictionary-based approach,” Procedia Comput. Sci., vol. 100, pp. 55–61, 2016.*
11. *D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, “Nymble: a high-performance learning name-finder,” in ANLC, 1997, pp. 194–201.*
12. *G. Szarvas, R. Farkas, and A. Kocsor, “A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms,” in DS. Springer, 2006, pp. 267–278.*
13. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.‏
14. Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).‏
15. *The Illustrated Transformer - https://jalammar.github.io/illustrated-transformer/*
16. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way -<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
17. Word2Vec (skip-gram model): PART 1 – Intuition – medium
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).‏
19. <https://www.ibm.com/cloud/learn/neural-networks>
20. <https://medium.com/analytics-vidhya/recurrent-neural-network-and-its-variants-de75f9ee063>
21. <https://peltarion.com/blog/data-science/self-attention-video>
22. Build a Custom BiLSTM Model Using TensorFlow: A Step Guide – TensorFlow Tutorial - <https://www.tutorialexample.com/build-a-custom-bilstm-model-using-tensorflow-a-step-guide-tensorflow-tutorial/>
23. How Does Named Entity Recognition Work: NER Methods - medium