סדנה בעיבוד שפה טבעית בעברית – פרויקט סופי

כיף עם ויקיפדיה

מריה צייטלין • 316984368 • unimaria
סיפן נוה • 208647834 • nfun

תוכן עניינים

רקע

ויקיפדיה היא אנציקלופדיה מקוונת המבוססת על תוכן חופשי ומשתמשת בטכנולוגיית ויקי. ויקיפדיה נכתבת ומשופרת מדי יום בידי מתנדבים, ומסתמכת על חוכמת ההמונים. המתנדבים כותבים את הערכים, יוצרים את הגרפיקה המעטרת את האנציקלופדיה, וכן מסייעים בהגהה ובניטור התוכן מפני השחתות. ויקיפדיה נכתבת בכ-300 שפות, וביניהן גם עברית.[1]

ויקינתונים (באנגלית: Wikidata) הוא מסד נתונים חופשי, שיתופי ורב-לשוני. ויקינתונים משמש כמאגר נתונים מרכזי לכל המיזמים של קרן ויקימדיה וביניהם ויקיפדיה.[1] ויקינתונים הוא בסיס נתונים גרפי בפורמט RDF. לכל אובייקט שמור פריט במסד, כאשר הפריט מזוהה על ידי מזהה יחודי בשם QID, ומכיל את שם הפריט, תיאור של הפריט, ורשימה של קביעות. כל קביעה מכילה מאפיין מסוים ואת ערכו. לדוגמה, המאפיין "שיא הגובה" בפריט "אוורסט" יקבל את הערך 8848.

הרעיון לפרויקט עלה תוך כדי עריכה בוויקיפדיה. כאשר משתמש כותב ערך חדש על אישיות בוויקיפדיה, חלק מהפעולות שהוא צריך לבצע הן מילוי מידע על האישיות בוויקינתונים, כדי ליצור באופן אוטומטי את תבנית ה-info-box שנמצאת בצד הערך. הפתרון שמצאנו מאפשר ליצור בקלות הצעה לפריטי ויקינתונים מתוך הערך העוסק באישיות בוויקיפדיה, ובכך להקל על העורך. הדבר נעשה באמצעות הבנה של המשמעות הסמנטית במשפט וחילוץ המידע הבדיד מתוך הטקסט קולח תוך שימור המשמעות, כך שבסופו של דבר מתקבל אוסף של המידע הבדיד שנמצא במשפט. כל שורה במידע הבדיד תתויג לפי מאפיין שניתן להבין ממנו בקלות מה צריך להיות המאפיין בוויקינתונים.

אם נוכל בעתיד גם ליצור ערך בסיסי מתוך המידע הבדיד שמופיע בוויקינתונים, זו תהיה תרומה גדולה לוויקיפדיה. הטמעה של כל אחד משני הכלים יכולה לשפר מאוד את חוויית העורכים.

[1] לקוח מוויקיפדיה

שאלת המחקר

במסגרת הפרויקט עלו שתי מטרות מחקר משלימות. שתי המטרות עוסקות ביצירת מידע סמנטי הניתן להבנה מטקסט בעברית.
השאלה הראשונה נוסחה על מנת לאפשר לנו יצירה של תבנית info-box בויקיפדיה באופן אוטומטי, והכנסת מידע לויקינתונים.

"האם וכיצד ניתן לחלץ מידע בדיד בעל משמעות סמנטית ממשפט בעברית העוסק באישיות, באופן אוטומטי, באמצעות כלי NLP?"

השאלה השניה משלימה את הראשונה ביצירת טקסט קולח מהמידע הבדיד הנמצא בויקינתונים.

"האם וכיצד ניתן ליצור טקסט בעברית המתאר אישיות ממידע בדיד, באופן אוטומטי, באמצעות כלי NLP?"

החלטנו להתמקד במימוש פרויקט העונה על שאלת המחקר הראשונה ולכן הכותרות הבאות המסמך עוסקות בשאלת המחקר הראשונה. לשאלת המחקר השניה נתייחס בקצרה בפסקה ייעודית.

המטרות

משאלת המחקר הראשונה נוכל לגזור את המטרה הבאה: בהינתן טקסט רציף העוסק בדמות, נרצה לחלץ מידע בדיד על הדמות מתוך הנאמר בטקסט. לדוגמה, מהמשפט "אלברט איינשטיין נולד בגרמניה וגר בשוויץ", נרצה לחלץ את המידע הבא:

משאלת המחקר השניה נוכל לגזור את המטרה הבאה: בהינתן מידע בדיד העוסק בדמות, נרצה לפלוט טקסט רציף שיעסוק בדמות וישקף את המידע הבדיד. לדוגמה:

גישות ורעיונות

כדי להשיג את מטרתנו הסופית – חילוץ אינפורמציה בדידה ממשפט, עלינו להשיג מטרת ביניים שכוללת הבנה של המשפט מבחינה סמנטית. במהלך המחקר שביצענו, בדקנו מספר גישות וטכניקות NLP שעשויות לאפשר את השגת המטרה.

Question Answering

השימוש במענה על שאלות להבנה סמנטית של טקסטים מתחיל לתפוס תאוצה במחקר האקדמי בשנים האחרונות.[2] הגישה נחקרה והוכחה ב-NLP בשפה האנגלית תוך בניית מאגר של ייעודי בטכניקה בשם QA-SRL.
QA-SRL היא טכניקה המשתמשת בשאילת שאלות תבניתיות מסביב לפרדיקט של המשפט ובתיוג התשובות שלהן, כדי לאפשר הבנה סמנטית של משפטים.[3] הטכניקה משמשת כבסיס למגוון משימות NLP, שביניהן Open Information Extraction.[4]

לאחר שבדקנו את הגישה, החלטנו שיישום שלה בעברית עשוי להיות מעניין ונוכל להשיג את המטרה שלנו באמצעותה. שאלות שבנויות מעל נשוא המשפט מאפשרות הבנה של יחסים במשפט, ונותנות מבנה אלגנטי של מידע בדיד. לדוגמה, במשפט "אלברט איינשטיין נולד בגרמניה וגר בשוויץ", שאלה כמו "מי נולד?" תוציא את שמה של הדמות שעליה מדברים במשפט – אלברט איינשטיין. נוכל גם לשאול שאלות מתוחכמות יותר כמו "היכן הוא נולד?" ולקבל את התשובה "בגרמניה". שימוש בשאלה זהה לאחר ניתוח מורפולוגי של המשפט היה משאיר אותנו עם התשובה "גרמניה", ומאפשר לנו לחלץ את המידע הבדיד בצורה קנונית.

חיפוש של מאגר מידע של שאלות ותשובות בעברית שנוכל להשתמש בו לביצוע המשימה לא הניב תוצאות. לכן, החלטנו לחפש דרך חדשה לביצוע המשימה, ולהמשיך לחקור את כיוון השאלות והתשובות בעברית במחקר שנעשה במסגרת המאסטר.

שימוש ב-Named entity recognition

לאחר שמציאת מאגר שאלות ותשובות בעברית לא הצליחה, הבנו שעלינו למצוא פתרון חדש שיאפשר לנו לחלץ את המידע, תוך שימוש באמצעים שכבר עומדים לרשות ה-NLP בעברית. החלטנו לבחון את הכיוון של שימוש ב-NER מתוך הנחה שתיוג של מילה כישות יאפשר לנו להבין תובנות מעניינות.

לאחר העברת משפט במודל שמתייג NER, נקבל משפט עם תיוגים של מילים. לדוגמה, במשפט "אלברט איינשטיין נולד בגרמניה וגר בשוויץ", נקבל את התיוגים הבאים:

B-PER	E-PER	O	O^S-GPE	O^O	O^S-GPE
"אלברט	איינשטיין	נולד	בגרמניה	וגר	בשוויץ"

ניתוח תמים של המשפט לאחר תיוגו יוביל להבנה שמדובר באדם בשם אלברט איינשטיין, ובשתי מדינות – גרמניה ושוויץ. באמצעות post-processing נוסף של המשפט שהתקבל לאחר התיוג, נוכל לחלץ את

[2] *About the QA-SRL Project* - https://qasrl.org/
[3] *QA-SRL First Approach - https://dada.cs.washington.edu/qasrl/docs/emnlp2015_hlz.pdf*
[4] *QA-SRL for OpenIE - https://aclanthology.org/N18-1081/*

היחסים הסמנטיים שבטקסט. במקרה הנוכחי, נוכל להבין שהמילה "גרמניה" שתויגה כמדינה מתייחסת לפועל "נולד", ובכך לחלץ מידע בדיד שמגדיר את גרמניה כמדינת לידה.

כעת עמדו בפנינו שתי אפשרויות לביצוע ה-NER. הראשונה היתה להשתמש ב-NEMO[2] שאומן במעבדה של פרופסור רעות צרפתי, והשניה לבצע finetuning של אלף-ברט לביצוע משימת NER, תוך שימוש בקורפוס של NEMO.
האפשרות שנבחרה היא האפשרות השניה – אימון אלף-ברט עצמאי. הבחירה באפשרות הזו אפשרה לנו להוסיף למודל תיוגים נוספים שלא קיימים ב-NER המקורי, כמו תיוג של מקצוע – OCC, שהוא מידע שנרצה לחלץ אודות אישיות מסוימת.

בניית הקורפוס

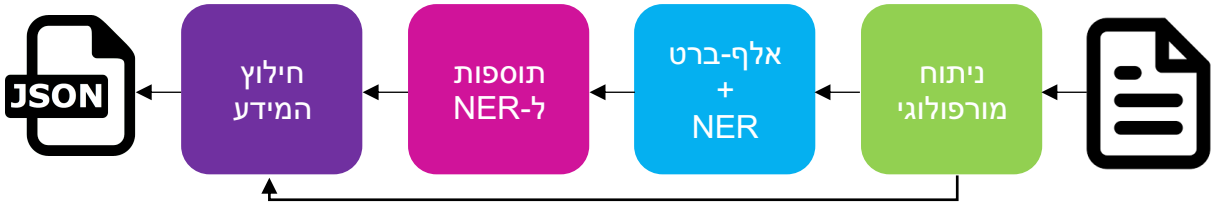
בתחילת הפרויקט רצינו להשתמש במאגר שמכיל שאלות ותשובות בעברית, אך לא מצאנו אחד. לאחר שהחלפנו את הגישה לאחרת, גם סוג הקורפוס שהיינו צריכות להשתמש בו השתנה. הקורפוס שהשתמשנו בו בפרויקט הוא הקורפוס המוכן NEMO-Corpus.[5] הקורפוס מכיל כמה עשרות אלפי משפטים ואת תיוגי ה-NER שלהם. בשביל להעביר את הנתונים באלף-ברט לא היה צריך לבצע עליהם פעולות מיוחדות.

בשלב מאוחר יותר של הפרויקט נדרשה העשרה של המידע בקורפוס המקורי, כדי שהמודל ידע להתמודד גם עם תיוגים שלא מופיעים בו, כמו "מקצוע" (OCC). בשביל ההעשרה השתמשנו בקטגוריות של ויקיפדיה כדי ליצור רשימה של כ-350 מקצועות. כתבנו סקריפט שמשתמש ברשימה ויודע לתייג את המידע המקורי שבקורפוס תוך שימור תיוגים קיימים אם יש כאלו, ותוך שימור המבנה המורפולוגי (לדוגמה, מילה כמו "כ-נהג" צריכה להיות מתויגת בתיוג O^OCC שמשמר את החלוקה של המילה למורפמות).

[5] *NEMO-Corpus on Github* - https://github.com/OnlpLab/NEMO-Corpus

המודל הנבחר

המודל שבחרנו בנוי בצורת פייפליין שהמידע שעובר בו מתחיל כטקסט, ונפלט כמילון שמכיל את המידע המחולץ. מודלי ה-NLP שאנחנו משתמשים בהם כוללים NER, וחלוקה מורפולוגית של משפט באמצעות Stanza. בשלב מתקדם יותר בפייפליין מבוצעת אנליזה באמצעות קוד פייתוני שלא כולל מודל מאומן. הפייפליין מורכב ממבנה בסיסי, ומתוספות אופציונליות. נפרט על כל השלבים בפייפליין.



הקלט: הקלט שהמודל יודע לקבל הוא קלט טקסטואלי שכולל משפט אחד או יותר. השימוש במודל מתבצע דרך API פייתוני שיודע לקבל משפט כקלט inline, או לקבל נתיב לקובץ המכיל את הטקסט. בגלל ההתנהגות של Stanza, המודל מתפקד בצורה טובה יותר כאשר הקלט לא כולל ניקוד, טקסט באנגלית, וסימני פיסוק מוזרים.

ניתוח מורפולוגי: שלב הניתוח המורפולוגי הוא שלב קלאסי ב-NLP בעברית. בשלב הזה השתמשנו ב-Stanza לביצוע ניתוח סינטקטי וחלוקה מורפולוגית של הטקסט. הקלט לשלב היה טקסט רציף. מתוך הפלט ש-Stanza פולט, המידע שמעניין את המערכת שלנו הוא החלוקה למורפמות, חלק הדיבר של המילה, והאב של המילה בעץ הגזירה של המשפט. בהינתן משפט הקלט האהוב עלינו - "אלברט איינשטיין נולד בגרמניה וגר בשוויץ", הפלט של השלב הזה יהיה:

```
אלברט: Info(head=2, pos='PROPN')
איינשטיין: Info(head=0, pos='PROPN')
נולד: Info(head=-1, pos='VERB')
ב: Info(head=4, pos='ADP')
גרמניה: Info(head=2, pos='PROPN')
ו: Info(head=6, pos='CCONJ')
גר: Info(head=2, pos='VERB')
ב: Info(head=8, pos='ADP')
שוויץ: Info(head=6, pos='PROPN')
```

אלף-ברט + NER: השתמשנו באלף-ברט, וביצענו fine-tuning עם המידע של NEMO-Corpus. השלב הזה משמש את המודל לתיוג מילים במשפט כ"מעניינות" תוך התייחסות למשמעות הסמנטית של המילה. לדוגמה, במשפט האהוב עלינו, גרמניה תתוייג כ-GPE שפירושו "ישות גיאו פוליטית" או במילים אחרות - מדינה. במקרה שלנו, מילים שמתויגות כ-Named entity הן מעניינות כי הן מתייחסות לאובייקט בעל שם שקשור לאישיות שהמשפט מדבר עליה. בצורה הזו נוכל ללמוד את שם האישיות, שמות של מדינות שהאדם קשור אליהן, של ארגונים שעבד בהם, של שפות שהוא מדבר ועוד.

מאוחר יותר במהלך העבודה על הפרויקט, הבנו שיש עוד מילים שנרצה שהמודל יתייג ואין להן תיוג ב-NEMO הבסיסי, לדוגמה - שמות של מקצועות. כדי להוסיף תיוג של מקצועות, הרצנו על ה-data של NEMO-Corpus סקריפט שכתבנו שמטרתו לתייג מקצועות בקורפוס באופן אוטומטי בהתאם לדרישות. רשימה של כ-350 מקצועות נגזרה מוויקימילון, והם אלו ששימשו אותנו בתיוג האוטומטי של המידע,

מעל לתיוג המקורי של הקורפוס. באופן כזה, ניתן לשלוט על התיוגים שיצאו מאלף-ברט. אם נרצה מידע נוסף, נוכל פשוט להוסיף תיוגים לקורפוס המקורי, ולקבל אותם "בחינם" בפלט של המודל.

בבדיקת המודל הסקנו שהמידע מתויג בצורה אמינה יותר בשלב השימוש במודל אם מבצעים חלוקה מורפולוגית של המשפט לפני שלב התיוג, ולכן הטקסט נכנס לשלב הזה לאחר פרסור של Stanza. גם טקסט ללא פרסור של Stanza עובד כראוי, ואם לא היינו משתמשות ממילא בניתוח מורפולוגי בפייפליין, אז לא בטוח שהיה הכרחי להוסיף אותו רק בשביל השלב הזה. בפרק "מסקנות" נמצא ניתוח נוסף של השימוש ב-Stanza שנכתב לאחר הערכת המודל.
הפלט של השלב הזה הוא תיוג של כל מילה כישות ניתנת לשיום, או כמילה שאינה ישות כזו. הנה, לדוגמה, התיוג של המשפט האהוב עלינו לאחר חלוקה מורפולוגית, "אלברט איינשטיין נולד ב גרמניה ו גר ב שוויץ" (משמאל לימין):

```
['B-PER', 'E-PER', 'O', 'O', 'S-GPE', 'O^O', 'O', 'O', 'S-GPE']
```

אלברט איינשטיין תויג כשם, וגרמניה ושוויץ כמדינות.

תוספות: בשלב הזה הוספנו חילוץ של שדות שאנחנו לא מעוניינים שהמודל ילמד, ונעדיף שיחולצו בדרך אחרת. דוגמה לשדה כזה היא תאריך. לתאריכים יש מבנה ייחודי שניתן לתפוס עם ביטוי רגולרי, ונעדיף לבצע התאמה מדוייקת על פני למידה, כאשר זו לא נחוצה, כדי להגדיל את הדיוק הסופי של הפייפליין. לצורך המשימה נכתב מודול נפרד בחבילה שאחראי על תיוג תאריכים. בגלל העיסוק באישים, הוחלט שהמודל צריך לדעת לזהות תאריכים בקנה מידה של שנים, ותאריכים שכוללים רק חודש לא מעניינים מספיק. לאחר הוספת המודול, המודל יודע לזהות תאריכים בשלל פורמטים, החל מתאריך הכולל מספרים בלבד, ועד לתאריכים הכוללים טקסט ומספרים במשולב (23 בנובמבר 1958), וגם ביטויי זמן כמו "לפנה"ס" או "שנה".
בשלב מאוחר יותר של המודל נוסף גם זיהוי פשוט של מספרים כדי לחלץ מידע כמו גיל פטירה, מספר ילדים ועוד.

חילוץ המידע: הקלט לשלב מתקבל הן משלב הניתוח המורפולוגי, והן משלב ניתוח הישויות במשפט. השלב מכיל סקריפט חכם שמוצא את השורש הפועלי בעבור כל מילה שסומנה בתור ישות בשלב השני של הפייפליין. בהינתן מילה שסומנה כישות, לדוגמה "גרמניה", נעלה בעץ הגזירה שהתקבל כפלט של Stanza, עד שנמצא מילה שהיא פועל, או עד שנגיע לשורש העץ. הפועל שנמצא הוא הפועל שהמילה קשורה אליו. במקרה שלנו, המילה "גרמניה" קשורה לפועל "נולד", וקיבלה תיוג של "מדינה", ולכן נוכל להסיק שמדינת הלידה של הישות שמדברים עליה במשפט היא גרמניה.
לכל מילה נבנה מילון שכולל את הטקסט המאוחד לאחר שלב ה-NER (לעיתים השלב מתייג שתי מילים כהמשך של שם הישות המשויימת, כמו ב"אלברט איינשטיין" ואנחנו רוצים שהמודל יתייחס אל כל הטקסט כאל מילה אחת). לכל טקסט כזה, נשמור את התיוג השיומי שלו, ואת השורש הפועלי שלו.

```
טקסט: אלברט איינשטיין, NER: שם, שורש: לידה
טקסט: גרמניה, NER: מדינה, שורש: לידה
טקסט: שוויץ, NER: מדינה, שורש: מגורים
```

הפלט: בשלב מוקדם של הפרויקט התכנון היה להמיר פעלים לשמות פעולה כדי לייצג את התוצאות הבדידות בצורה יותר קולחת. חיפוש של API שמבצע את הפעולה העלה חרס, ובניה ידנית מתוך PDF שמצאנו הייתה יקרה ולא שווה את המאמץ. כמו כן, הבנו שהוצאת הפלט כפעלים קריאה וברורה דיה. לכן, החלטנו לא להציג שמות פעולה, אלא פעלים לאחר שחוו את נחת ידה של Stanza.
השלב ראשוני של הפרויקט כן כלל המרה ידנית של מספר קטן של פעלים לשמות פעולה. לכן, לאחר העברת הטקסט "אלברט איינשטיין נולד בגרמניה וגר בשוויץ" בפייפליין, קיבלנו את הפלט:

{'שם': 'אלברט איינשטיין', 'מדינה – לידה': 'גרמניה', 'מדינה – מגורים': 'שוויץ'}

הקלט "אברהם נדל עבד בחברת אגד כנהג אוטובוס בישראל" הוביל לפלט שמשמאל:

שם: אברהם נדל
ארגון – עבודה: אגד
מדינה – עבודה: ישראל

בתהליך הפיכת המודל לגנרי יותר, עם השיפורים שהוספנו,
תצורת הפלט השתנתה וכעת הפלט של המודל נפלט בצורה מעט שונה שכוללת את התיוג, ולאחריו set של זוגות של מילים והפעלים שהן קשורות אליהם. במקרה של המשפט העוסק באלברט איינשטיין, נקבל כפלט {'שם': 'אלברט איינשטיין', 'מקום': {('גרמניה', 'נולד'), ('שוויץ', 'גר')}.

יתרונות המודל

כתוצאה ישירה של המבנה הייחודי של המודל המורכב ממספר חלקים, המודל שהתקבל גמיש מאוד. המבנה מאפשר להחליף כל חלק בו בחלק משופר או ייעודי לביצוע מטרה מסוימת. הוספה של חתימה על תאריכים היתה פשוטה מאוד ולא הצריכה שינוי כלשהו במודל מלבד הרצת סקריפט תיוג בין שלב התיוג הראשוני לבין בניית עצי הגזירה. המודל בנוי בצורה גנרית ולכן הוא שימושי למגוון משימות IE, כאשר השינוי היחיד שיצטרך להיות מבוצע במודל הוא שינוי של הקורפוס. במקרה שלנו, הוספנו לקורפוס תיוג של מקצועות, מכיוון שרצינו שהמודל יזהה גם את אלו כמאפיין של אדם. כמשימה עתידית, היה מעניין לבדוק החלפה של חלקים ספציפיים במודל כמו ניסיון להחליף את סטנזה ב-YAP, או כמו תיוג מקורי באמצעות NEMO[2] ולא באמצעות אלף-ברט מאומן. כל החלפה כזו היא פשוטה ולא מצריכה שינויים רבים.

חסרונות המודל

בגלל המבנה ה-pipeline-י של המודל, התוצאה הסופית מושפעת מחוסר דיוק בכל אחד מהשלבים שלו, כך שהרכבה של תיוג NER שאינו מושלם על חלוקה מורפולוגית של סטנזה שאינה מושלמת, עשויה להקשות על המודל להוציא ביצועים טובים.
לדוגמה, אם סטנזה פלטה פירוק של המילה כסלו למורפמות כ-"ה כסל של הוא", סקריפט החתימה על תאריכים לא ידע לזהות את כסלו כחודש. באופן דומה בעבור מילים שאלף-ברט מתייג.

הערכה, איוולואציה

מכיוון שאין לנו קורפוס מסודר שמכיל מידע טקסטואלי ואת הפלט המתבקש, ההערכה של טיב המודל לא הייתה טריוויאלית. לאחר מחשבות ודיונים, בחרנו להגדיר שני מדדים שדומים בהגדרתם ל-precision ול-recall, ולבצע הערכה ידנית תוך השוואה ל-infobox של הערך בוויקיפדיה.
כדי לבצע את ההערכה של המודל, הגדרנו מספר מושגים מחדש ב-scope של הפרוייקט.
לצורך ההערכה, בחרנו עשרים ושניים אישים בעלי מאפיינים מגוונים – שלל מקצועות, מדינות מגורים, תקופת פעילות בהיסטוריה ועוד. לכל אישיות יצרנו טקסט שמתבסס על הערך בוויקיפדיה, ומכיל את כל המידע שמופיע ב-infobox שהמודל אומן לפלוט. לאחר הרצה של המודל על הטקסט, השווינו את הפלט למידע הבדיד המופיע בוויקיפדיה, וביצענו חישוב precision ו-recall על פי ההגדרה שתופיע בהמשך.

הגדרות

True Positive - פיסת מידע שהופיעה ב-info-box, שנכתבה במשפט שניתן כקלט למודל, והופיעה בפלט שלו.
False Positive - פיסת מידע שלא הופיעה ב-info-box, שנכתבה במשפט שניתן כקלט למודל, והופיעה בפלט שלו וברור שהיא לא רלוונטית או לא נכונה.
נגדיר - False Negative פיסת מידע שהופיעה ב-info-box, שנכתבה במשפט שניתן כקלט למודל, ולא הופיעה בפלט שלו כלל.
False Negative - פיסת מידע שלא הופיעה ב-info-box, ולא הופיעה בפלט של המודל כלל.
True Negative - לא מוגדר בהקשר של הפרויקט, מכיוון שאין אפשרות לכמת את כל המידע בעולם שהיה פוטנציאלי לפלוט.
TP~FN - הגדרה חדשה שהוספנו לצורך הפרויקט, ופירושה פיסת מידע שהופיעה ב-info-box, נכתבה במשפט שניתן כקלט למודל, והופיעה בפלט באופן חלקי או שהופיעה במלואה תחת תיוג שגוי.

ניזכר בהגדרות של precision ו-recall:
Precision - כמה מהמידע שהוצא בפלט של המודל רלוונטי במציאות.
Recall - כמה מהמידע שהיינו מצפים שיצא, אכן נפלט במודל.
בהגדרות שלנו נחשיב גם הופעה של פלט בצורה חלקית.

$$precision\sim = \frac{TP + \frac{1}{2}TP\sim FN}{TP + FP + \frac{1}{2}TP\sim FN}$$

$$recall\sim = \frac{TP + \frac{1}{2}TP\sim FN}{TP + FN + TP\sim FN}$$

בחרנו לא לחשב accuracy מכיוון שהמדד תלוי גם ב-TN ובמקרה של המודל הוא לא מוגדר היטב. לא מוגדר מה כלל המידע הקיים בעולם שהיינו יכולים פוטנציאלית לפלוט ולא רצינו להכיל.
תהליך ביצוע ההערכה ותוצאותיה נמצאים במסמך נפרד.[6] התוצאות הסופיות תוצגנה בפרק "תוצאות".

[6] Evaluation Process (Google Docs) - https://tinyurl.com/wikie-eval

תוצאות

בשלב האיוולואציה כתבנו טקסטים בהשראת ויקיפדיה. כל טקסט עבר במודל, והפלט הושווה לinfobox של הערך בוויקיפדיה. התוצאות הוערכו בהתאם לנוסחאות שהצגנו בפרק "איוולואציה". את כל הטקסטים ששמשו בשלב הזה ניתן למצוא בקוד של הפרויקט ב-GitHub. ראשית, נציג תוצאות שהמודל הוציא, ולאחר מכן תוצג הטבלה הסיכומית של תוצאות כל הטקסטים. נזכיר שתהליך שערוך התוצאות מוצג במסמך נפרד (ראו נספחים).
בנוסף להצגת תוצאות הרצת המודל על טקסטים, נציג גם את אופן פועלו על משפטים קצרים ותמציתיים שכוללים בדיוק את המידע הרצוי.
כמו כן, נציג שני מקרי בוחן אמיתיים שבהם השתמשנו במודל כדי לעדכן את התבנית בוויקיפדיה עצמה.

תוצאות אימון אלף-ברט ל-NER

בסוף אימון המודל, בדקנו את הביצועים שלו אל מול ה-test שהשארנו בצד מבעוד מועד. קיבלנו תוצאות מרשימות למדי:

Accuracy: 0.885
Recall Micro: 0.885
Precision Micro: 0.885
F1 Micro: 0.885
F1 Weighted: 0.876

מקרה בוחן – נועה קירל

הטקסט שהעברנו למודל נכתב תוך שימוש במשפטים מוויקיפדיה, והושלם עם משפטים שמכילים מידע שקיים ב-infobox.
"נועה קירל (נולדה ב-10 באפריל 2001) היא זמרת, שחקנית ורקדנית ישראלית.
קירל נולדה וגדלה ברעננה שבישראל.
שם הלידה של קירל היה נויה קירל אך שונה בעצת רב.
נועה פעילה משנת 2015 ועד היום.
היא נמצאת בזוגיות עם יהונתן מרגי."

לאחר העברת הטקסט במודל, התקבלו התוצאות הבאות:
'שם': 'מרגי', 'נויה קירל', 'נועה', 'נועה קירל', 'קירל'
'תאריך לידה': '10 ב אפריל 2001'
'מקצוע': 'זמרת', 'רב', 'רקדנית', 'שחקנית'
'ארגון/עיר': ('רעננה', 'נולדה')
'מקום': ('ישראל', 'נולדה')
'תאריך': ('מ שנת 2015', 'פעילה')

ניתן לראות שהמודל זיהה בדיוק רב את מקום הלידה ואת התאריכים המזוהים עם פעילותה. שמו של יהונתן מרגי זוהה חלקית, והמודל זיהה גם את המקצוע "רב", שלא באמת קשור לקירל עצמה, אך הופיע בטקסט.

תוצאות הערכת המודל

באופן דומה להערכה שהדגמנו באמצעות הטקסט של נועה קירל, בוצעה ההערכה על עשרים ושניים אישים. תוצאות ביצוע ההערכה על כל אחד מהם מוצגות בטבלה הבאה:

האישיות	מספר נתונים לבדיקה	ניקוד - ~precision	ניקוד - ~recall
דוד בן גוריון	17	88.24	88.24
אלברט איינשטיין	14	90.48	67.86
פרדריק שופן	14	91.67	78.57
נועה קירל	10	100	95
ג׳ולי אנדרוז	22	89.47	77.27
הורטיוס	10	100	85
מרטי רובינס	16	93.1	84.38
נתן אלתרמן	26	95.12	75
ש״י עגנון	14	86.67	92.86
קרן פלס	13	100	73.08
דבורה עומר	14	92	82.14
אברהם למפל	10	81.82	90
בנג׳מין פרנקלין	15	92	76.67
גל גדות	14	100	67.86
אלי כהן	18	93.75	83.33
אילון מאסק	22	88.57	70.45
יונית לוי	14	90.91	71.43
ליא קניג	14	84.62	78.57
אלן טיורינג	10	75	90
גבי אשכנזי	16	92.59	78.13
בילי ג׳ואל	21	89.74	83.33
שמעון פרס	22	86.96	90.91
ממוצע	**15.7**	**91.05**	**80.35**

המדד של ה-recall מייצג כמה המודל היה טוב בזיהוי מידע מעניין. ה-precision מייצג את כמות ה"זבל" שהמודל הוציא, וככל שהוא יותר גבוה, כך המודל פלט פחות מידע זבלי. הציפיה שלנו לפני הרצת המודל היתה שה-precision יהיה נמוך, מפני שהמודל נוטה להוציא הרבה מידע לא קשור שמופיע בטקסט. עם זאת, תהליך הניקוד היה סלחן, ולא העניש את המודל על מידע שהיה לגיטימי שיפלוט בהינתן שזה אכן כתוב בטקסט (כדוגמת מקצוע ה"רב" בדוגמה של נועה קירל למעלה). אם תהליך ההערכה היה מחשיב מקרים כאלו, התוצאה אכן היתה נמוכה יותר, בהתאם למצופה. בפרק "רעיונות לעתיד" נדבר על אפשרות לווסת את הפלט שהמודל מוציא במקרים שבהם הוא אינו רלוונטי לטקסט. במודל כפי שהוא היום לא הוכנסה התחשבות כזו, ולכן לא הענשנו את המודל על הוצאת מידע לגיטימי שלא היינו רוצים שיצא במודל מושלם. ה-recall מראה שהמודל אכן מוצלח בזיהוי המידע הבדיד בטקסט, אך יש מידע שהוא מפספס. ניתוח מעמיק של אזורים שבהם המודל מתנהג בצורה פחות מוצלחת ושל הסיבות לכך, נמצא בפרק "מסקנות".

ניתוח משפטים בודדים

במהלך העבודה על המודל השתמשנו במשפטים קצרים וממוקדים לבדיקה שלו, ולא בטקסטים שלמים. הטקסים נכתבו רק לצורך שלב ההערכה. השימוש הצפוי במודל הוא אכן לחילוץ מידע מטקסטים שלמים, אך במסגרת בתוצאות בחרנו להציג גם את הביצועים שלו כשמדובר במשפטים קצרים.

אברהם נדל עבד בחברת אגד כנהג אוטובוס בישראל:
'שם': 'אברהם נדל'
'ארגון, עיר': ('אגד', 'עבד')
'מקצוע': 'כנהג'
'מקום': ('ישראל', 'עבד')
אלברט איינשטיין נולד בגרמניה ב-14 במרץ 1879 והיה מדען מפורסם:
'שם': 'אלברט איינשטיין'
'מקום': ('גרמניה', 'נולד')
'תאריך': ('14 ב ה_ מרץ 1879', 'נולד')
'מקצוע': 'מדען'
שמעון פרס היה מדינאי ונפטר בגיל 93:
'שם': 'שמעון פרס'
'מקצוע': 'מדינאי'
'מספר': ('93', 'נפטר')
יוסי מור (נולד ב-21 בדצמבר 1983) הוא מוזיקאי, קלידן, מפיק מוזיקלי:
'שם': 'יוסי מור'
'תאריך לידה': '21 ב דצמבר 1983'
'מקצוע': {'מוזיקאי', 'מפיק'}

שימוש במודל

כדי להכריז על כך שהמודל אכן מניח את הדעת, יש להשתמש בו במציאות למטרה שלשמה הוא נועד. לכן, מצאנו ערך בוויקיפדיה שעוסק באדם, ואין לו תבנית מידע – תומאס בלאט.

תומאס "טויבי" בלאט (באנגלית: **Thomas "Toivi" Blatt**, בפולנית: **Tomasz Blatt**; 15 באפריל 1927, י"ג בניסן ה'תרפ"ז - 31 באוקטובר 2015, י"ח בחשוון ה'תשע"ו) היה דובר וסופר אמריקאי יהודי יליד פולין. בהיותו בן 16 היה מבין הניצולים היחידים ממחנה המוות סוביבור ששרדו את ניסיון המרד והבריחה באוקטובר 1943, אירוע שבו ברחו 300 אסירים ממחנה ההשמדה. רובם נרדפו, נלכדו והוצאו להורג על ידי הגרמנים.


תומאס בלאט בשנות הארבעים

תוכן עניינים [הסתרה]

בהרצת המודל על טקסט שנלקח מהערך, התקבל הפלט הבא:

*{**'שם':** {'סוביבור', 'תומאס בלאט', 'תומאס', 'טויבי', 'יהודית', 'בלאט', 'איזביצה', 'טרנספורט', 'ברברה', 'סנטה'},*
*'**תאריך לידה':** {'י"ג ב ניסן תרפ"ז', '15 ב אפריל 1927'},*
*'**תאריך פטירה':** {'י"ח ב חשוון תשע"ו', '31 ב אוקטובר 2015'},*
*'**מקצוע':** {'סופר'},*
*'**מקום':** {('צרפת', 'נרצחו'), ('הולנד', 'נרצחו'), ('מפולין', 'נרצחו'), ('ישראל', 'עלה'), ('קליפורניה', 'התגורר'), ('פולין', 'נולד'), ('פולין', 'סופר'), ('ארצות', 'עלה')},*
*'**מספר':** {('1,000', 'נרצחו'), ('16', 'ניצולים'), ('400', 'נשלחו'), ('88', 'נפטר'), ('250,000', 'נרצחו'), ('300', 'ניצולים')},*
*'**תאריך':** {('ב שנת 1927', 'נולד'), ('ב שנת 1958', 'עלה'), ('ב אוקטובר 1943', 'שרדו'), ('31 ב אוקטובר 2015', 'נפטר'), ('ב שנת 1942', 'נשלחו')}}*

תוך שימוש בתוצאות המודל ובמידע שכבר היה בוויקינתונים, נוצר ה-infobox וכיום הערך בוויקיפדיה מכיל תיבת מידע, כפי שמוצג בתמונה משמאל.

תומאס בלאט
Tomasz Blatt


תומאס בלאט בשנות הארבעים

לידה	15 באפריל 1927 י"ג בניסן ה'תרפ"ז איזביצה, פולין
פטירה	31 באוקטובר 2015 (בגיל 88) י"ח בחשוון ה'תשע"ו סנטה ברברה, ארצות הברית
מדינה	פולין, ארצות הברית
תאריך עלייה	1958
מקצוע	סופר, כותב זכרונות
פרסים והוקרה	צלב הקצין של מסדר ההערכה של הרפובליקה של פולין

בהמשך, הוחלט להריץ את המודל על אישיות נוספת, ולמלא את הנתונים בהתאם לפלט של המודל. האישיות שנבחרה הייתה יעקב צים.
בהרצת המודל על הטקסט, התקבל הפלט הבא, שהפך לתיבת המידע שמוצגת משמאל:

*{**'שם':** {'סוסנוביץ', 'יעקב צימברקנופף', 'יעקב', 'צימברקנופף', 'פרס'},*
*'**ארגון, עיר':** {('בנק', 'זוכה'), ('בצלאל', 'למד'), ('מטרואה', 'עלה'), ('ירושלים', 'למד'), ('צים', 'נשוי'), ('משרד', 'זוכה'), ('יולס', 'נשוי'), ('צים', 'נולד'), ('לרות', 'נשוי'), ('צים', 'נפטר'), ('ירקון', 'נקבר'), ('צים', 'זוכה'), ('בוכנוואלד', 'עלה'), ('צים', 'עלה')},*
*'**תאריך לידה':** {'1920'},*
*'**תאריך פטירה':** {'27 ב דצמבר 2015'},*
*'**מקצוע':** {'צייר', 'מעצב'},*
*'**אירוע':** {('שואה', 'זוכה')},*
*'**מספר':** {('96', 'נפטר'), ('4', 'נולדו'), ('5', 'יוצרם')},*
*'**מקום':** {('פולין', 'נולד'), ('ישראל', 'זוכה'), ('ישראל', 'עלה')},*
*'**תאריך':** {('שנת 1947', 'למד'), ('2015', 'נפטר'), ('ב שנת 1920', 'נולד'), ('ב יולי 1945', 'עלה')}}*

יעקב צים
Jakob Cymberknopf

אין תמונה חופשית
היש בבעלותכם תמונה כזו?
אם כן, אנא לחצו כאן.

לידה	1920 סוסנוביץ, פולין
פטירה	27 בדצמבר 2015 (בגיל 95 בערך)
שם לידה	יעקב צימברקנופף
מדינה	ישראל
מקום קבורה	בית העלמין ירקון
תאריך עלייה	יולי 1945
השכלה	בצלאל, אקדמיה לאמנות ועיצוב
מקצוע	צייר, מעצב
בת זוג	רות צים
מספר צאצאים	4

מסקנות

נציג כעת את המסקנות הנובעות משאלת המחקר, ומשלב האיוולואציה והתוצאות.
שאלת המחקר ששאלנו היתה "האם וכיצד ניתן לחלץ מידע בדיד בעל משמעות סמנטית ממשפט בעברית העוסק באישיות, באופן אוטומטי, באמצעות כלי NLP?". מאחר שהמודל הניב תוצאות המניחות את הדעת, ניתן לומר שהמשימה אכן אפשרית, וניתן לחלץ מידע בעל משמעות סמנטית ממשפטים ואף מטקסטים בעברית העוסקים באישיות, באמצעות NLP. באשר לשאלה "כיצד", הפרויקט הנוכחי הציג מספר דרכים לביצוע המשימה, כאשר אחת מהן מומשה, נבדקה, ויעילותה הוכחה.
במהלך העבודה עלו גם מספר מסקנות נקודתיות שניתן לגזור מהן את חוזקות ואת חולשות המודל, ולאתר מקומות שניתן לשפר בעתיד.

אימון אלף-ברט ל-NER

במשך רוב תהליך העבודה על הפרויקט השתמשנו במודל NER שאומן בצורה חלקית על מספר קטן של משפטים מהקורפוס, ורק לקראת סוף העבודה השלמנו את האימון על הקורפוס כולו. אימון על חלק קטן מהקורפוס הביא לתוצאות טובות, אך האימון על כולו שיפר את המודל. דוגמה הממחישה את העניין תהיה משפט העוסק בזמרת אריאנה גרנדה. לאחר אימון המודל על מספר קטן של משפטים, הוא תייג את המילה "גרנדה" כמדינה (מכיוון שאכן קיימת מדינה כזו). בתום האימון, המילה גרנדה כבר לא מתויגת כמדינה במשפט על אריאנה גרנדה. מנגד, היא גם לא מתויגת כמדינה במשפט העוסק במדינה גרנדה.
ישנם אזורים שמדגימים תופעה של overfitting שבהם המודל תמיד מתייג מילה בתיוג מסוים, לא משנה באיזה הקשר היא הופיעה. דוגמה כזו תהיה המילה "פרס" שתמיד מתויגת כשם, על אף שברוב המקרים היא דווקא מתייחס לפרס שזוכים בו. כנראה שבקורפוס היו הרבה משפטים שעסקו בשמעון פרס.
כמו כן, המודל מתקשה בתיוג מקצועות שלא הופיעו בקורפוס. הדיוק שלו בתיוג מקצועות באופן כללי גבוה מאוד, אך הוא מתקשה בתיוג מקצועות איזוטריים כמו "מרגל". בנוסף, המודל מתקשה עם מקצועות המכילים יותר ממילה אחת, ונוטה לפספס את המילה השניה. דוגמה כזו היא "יועץ השקעות", המודל מתייג רק את המילה "יועץ" כמקצוע.
מכיוון שהמודל לא ראה שמות רבים של ערים באירופה, הוא בדרך כלל מתייג את אלו כשמות ולא כערים.

האספקט החלש ביותר של המודל נוגע לתיוג שפות. על אף שזו ככל הנראה המשימה שאדם היה מצליח לבצע בדיוק הכי גבוה בגלל המאפיינים היחודיים של מילה שמתארת שם של שפה, זו משימה שהמודל נכשל בה קשות. יתכן שלא הופיעו מספיק שמות של שפות בקורפוס. הדבר השפיע בצורה חזקה על ה-recall בהערכת טקסטים של אישים שמאפיין עיקרי שלהם הוא שימוש בשפות רבות – כמו נתן אלתרמן וליא קניג.
עם כל זאת, רצינו להשתמש במודל שמאומן על קורפוס שנוכל לשנות ולהוסיף לו מידע, כדי לאפשר התאמה של המודל לצורך הספציפי שלנו. כנראה ששימוש ב-NEMO מאומן היה מוביל לתוצאות טובות יותר, אך ההוספה של תיוג המקצועות לא הייתה מתאפשרת.
הבחירה האידיאלית, בהינתן יותר משאבים, הייתה ליצור קורפוס חדש שהטקסטים בו לקוחים מוויקיפדיה, והתיוגים בו מתאימים לאלו שהיינו רוצים שיופיעו בפלט. לו היו בידינו המשאבים, היינו משנות תיוגים בקורפוס הנוכחי, ומוסיפות עוד תיוגים שלא הוספנו כי מאתגר להוסיף אותם בצורה אוטומטית כמו שהוספנו את המקצועות.

השימוש בסטנזה

השימוש העיקרי של סטנזה בפרויקט היה לתרום להבנה של גזירת המשפט כדי לשייך מילה מתויגת לשורש הפועלי שלה (לדוגמה – [פולין, נולדה] לפליטת מדינת הלידה של אדם). במקרים רבים השימוש בסטנזה גם תרם לתיוג המשפט באלף-ברט בצורה נכונה. עם זאת, מניסויים שביצענו עלה שפעמים רבות אלף-ברט יודע לתייג גם מילים שמורכבות ממספר מורפמות. החיסרון העיקרי של סטנזה היה שהיא נוטה להתבלבל מניקוד, מסימנים באותיות לטיניות ומפיסוק, ומחליפה אותם באותיות אקראיות בעברית בלי להתריע. כך, משמו של בילי ג׳ואל התקבלה בפלט המילה ״גלואל״ שתויגה כשם. ניקוד בשמו של דוד בן גוריון גרם להוספת המילה ״חחן״ למשפט, ובמקרה אחר נוספה למשפט המילה ״פפפפפפק״. בנוסף, סטנזה לפעמים מפרקת מילים בצורה שמשפיעה על כל המודל וגורמת לNER לא לזהות את המילה כלל (לדוגמה, החודש ״כסלו״ פורק ל-״ה + כסל + של + הוא״). בעבודה עתידית על הפרויקט כדאי לשקול להכניס את המילים לתיוג לפני ההעברה בסטנזה, ולהשתמש בסטנזה רק מאחורי הקלעים כדי לשייך מילים שלמות לשורש פועלי ולא מורפמות בודדות.

מחשבות על FP ו-FN

המודל מוציא את כל מה שהוא מזהה בטקסט נתון, אין לו אמצעי סינון חכמים. לכן, מתן משפטים שמכילים הרבה מידע לא מעניין או לא קשור, ככל הנראה יוביל לקבלת מידע לא רלוונטי בפלט.
המודל טוב מאוד בעבודה עם שנים שמקושרות לפעלים מובהקים, ולכן הוא פולט בהצלחה צירי זמן כמו אלו שנמצאים בתיבת המידע בערך של שמעון פרס.
תוצאה בולטת של המודל היא שהוא לא מזהה מידע שניתן להסיק מהטקסט. ניתן דוגמה - אם בטקסט לא היה כתוב על בן גוריון שהוא פוליטיקאי אבל דובר על פעילותו המדינית, המודל לא יאמר שהוא פוליטיקאי. מצד אחד, התוצאה הזו הגיונית מאוד בגלל אופן הבניה של המודל. מצד שני, התוצאה קצת לא צפויה כי מסקנות כאלו באות לבני אדם בצורה טבעית, והעובדה שהמודל לא פולט אותן עשויה להרגיש כמו False Negative.

החלפת חלקי ה-pipeline לחלקים יעודיים

בגלל מבנה המודל, הוא מושפע בצורה חזקה מאי דיוקים קטנים בכל אחד מחלקיו. לכן, החלפה של החלק הראשון של NEMO במודל שיודע לתייג דברים יותר ספציפיים לuse-case, הייתה מובילה, ככל הנראה, לתוצאות טובות יותר. הבעיה היא שיקר לייצר קורפוס מתויג כזה, וזו הסיבה שבגללה השתמשנו בקורפוס המוכן.

רעיונות לעתיד

נציג את הרעיונות האפשריים לשיפור ולהרחבה של הפרויקט בעתיד, במגוון כיוונים.

- שיפור המידע שהמודל פולט. לדוגמה, הוספת הקטגוריה ׳בני זוג׳ - המודל תופס את שם בן הזוג רק כשם אבל לא מתייג בתיוג יעודי של ״בן זוג״. באופן דומה בעבור קטגוריות נוספות.
- מכיוון שהשתמשנו בקורפוס מוכן של NEMO, ישנם תיוגים בקורפוס שלא תאמו את הצרכים הספציפיים של ויקיפדיה. רעיון לשיפור הוא הוספת תיוגים שקשורים לויקיפדיה (באופן דומה להוספת התיוג ׳מקצוע׳), למשל פרסים (אוסקר, נובל, גראמי וכו׳), מקומות במדינות שאינן ישראל, תיוג מספרים שכתובים במילים ועוד.
  שיפור אחר בכיוון דומה יהיה ליצור קורפוס חדש לגמרי עם תיוגים של מילים שמופיעות בוויקיפדיה, כך שהקורפוס יהיה מותאם לחלוטין לצרכי המודל והמודל יוכל לזהות יותר ישויות.

- מכיוון שבפירוק של stanza היו הרבה טעויות שגרמו לזיהוי לא נכון של NER על ידי המודל, אפשר לבדוק שיפור באמצעות שימוש ב-stanza אחרי התיוג של המודל ולא לפניו כמו שעשינו.
- בבניית המודל התמקדנו בתפיסת המידע מתוך הטקסט, אך לא ייחסנו חשיבות גדולה לניקוי הפלט מזבל, כלומר להקטנת ה-FP. דרך טובה לשפר את המודל היא להוסיף דרכים לניקוי חכם של הפלט ובאופן הזה להגדיל את ה-precision.
- בפרויקט התמקדנו בשליפת מידע מטקסט בנושא "אישים". הרחבה מעניינת לפרויקט תהיה להוסיף את האפשרות לשלוף מידע מנושאים אחרים - צמחים, בעלי חיים, מדינות ועוד. כמו כן, ניתן לנסות להפוך את הפרויקט ליותר גנרי כך שישלוף מידע סיכומי "מעניין" מכל טקסט.
- שיפור תצוגת הפלט - כיום הפלט מוצג בצורת Dict, כך שנוח לעבוד איתו בסקריפטים שיתלבשו על הפרויקט. ניתן להוסיף אפשרות לתצוגה בצורת info-box בדומה לוויקיפדיה, כדי שיהיה יותר מסודר וקריא.

שימושים אפשריים ושיתופי פעולה

השימוש המקורי שהוצע לפרויקט היה הטמעה בויקיפדיה. ויקיפדיה מברכת שימוש בגאדג'טים ובכלים שנועדו להקל על העורך. במקרה הנוכחי, מהמטרה הראשונה ניתן לגזור גאדג'ט שיקבל את הערך בתור קלט, ויאפשר למשתמש לקבל את כל המידע שעליו להכניס לויקינתונים (או יעשה זאת באופן אוטומטי ויבקש מהמשתמש אישור לפני השמירה). מהמטרה השניה ניתן ליצור כלי שימושי, שיקרא את המידע בויקינתונים, ויציע הצעה התחלתית לערך, או הצעה לפסקת הפתיחה של ערך בנושא.

בנוסף, במהלך העבודה פנה אלינו אסף בר-לב מרשות התקשוב, וביקש שנשתף פעולה. הוא הציע שימוש נוסף לפרויקט שלא חשבנו עליו בעת בחירת הנושא – התממת מידע של משתמשים. כאשר משתמשים פונים לרשות ממשלתית, הם משאירים פרטים מזהים. כאשר רשות התקשוב רוצה להעביר משפטים לאקדמיה העברית לשם תיוג (למשימות NLP עתידיות), עליה להתמים ולהסתיר את פרטי הפונים. באמצעות שינוי קל בפרויקט, ניתן להסתיר את המידע האישי של אדם במקום להוציא מילון שמכיל אותו. למעשה, החלק הראשון בפייפליין מספיק לביצוע המשימה. מכיוון שהוא כולל הבנה של פרטי מידע, ניתן להחליף כל פרט כזה בתיוג שיצא מה-NER, ולקבל התממת פרטים. בשביל לבצע את המשימה ידרשו מספר שינויים, כגון הסרת האימון על מקצועות, ואימון נוסף על כתובות מגורים, על פרטי חשבון בנק, על מספרי טלפון ועוד. את אלו האחרונים ניתן כנראה לחלץ באמצעות regex ואין צורך לאמן מודל NLP בשביל להתמים אותם.

שאלת המחקר השניה

שאלת המחקר

האם וכיצד ניתן ליצור טקסט בעברית המתאר אישיות ממידע בדיד, באופן אוטומטי, באמצעות כלי NLP?

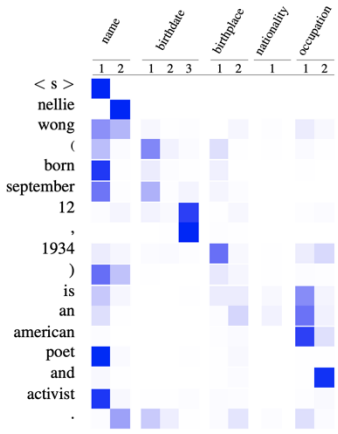
מאגר המידע

המאגר שנצטרך להשתמש בו הוא מאגר שכולל זוגות של מידע בדיד (מהצורה של הפלט של החלק הראשון) וטקסט שמתאים לו. נצטרך Dataset דומה ל-WikiBio שקיים באנגלית.[7] את המאגר נוכל ליצור באופן אוטומטי תוך שימוש במידע מוויקיפדיה.

[7] *WikiBio - https://rlebret.github.io/wikipedia-biography-dataset/*

המימוש

נוכל להשתמש ברשת נוירונים המכילה Attention Layer, כדי שהמודל יוכל להתמקד בכל פעם בחלק אחר של הקלט כדי לפלוט את המשפט הרלוונטי. ארכיטקטורה של Transformers מתאימה כאן מכיוון שהקלט בבעיה הזו הוא מאורך משתנה, והחלקים בקלט אינם בהכרח קשורים זה לזה. בנוסף, בפלט המבוקש אין סדר מסוים שיותר נכון מסדר אחר, ואלה תכונות שארכיטקטורה זו עונה עליהן.



הרעיון שלנו הוא להשתמש במודל שדומה במבנה שלו ל--Encoder Decoder, כפי שעשו במאמר Neural Text Generation from Structured Data with Application to the Biography Domain (Lebret et al.).[8]
במאמר מדגימים כיצד השתמשו ברשת נוירונים שמכילה שכבת LSTM וכן שכבת Attention.
בתמונה שמשמאל ניתן לראות את המחשת פעולת ה-Attention במשימה הזו (מתוך המאמר).

הרשת היעודית לפתרון המשימה בעברית תקבל כקלט מידע בדיד בצורת info-box ותפלוט משפט רציף המתאר בצורה הטובה ביותר את המידע. הפעולה היא מעין היפוך לא יחיד של הפעולה שמימשנו בשאלת המחקר הראשונה.

בידור, פספוסים ושעשועים

במסגרת העבודה על הפרויקט, היו לא מעט פלטים של המודל שהעלו חיוך על השפתיים ואף גרמו לפרץ צחוק בלתי נשלט. בחרנו לרכז כמה מהם כאן, ולהגישם כחלק מהדוח, מכיוון שאלו היו חלק בלתי נפרד מהעבודה על הפרויקט ומהאווירה שאפפה אותנו במהלכו.

- **'שם': {'דָּוִד', 'חחן'}** – סטנזה מצאה דרך יצירתית במיוחד לעוות את "דָּוִד בֶּן..." כי סטנזה לא אוהבת ניקוד. כך "בֶּן" הפך ל"חחן".
- **'ארגון': ('פפפפט', 'מייסד')** - לפעמים סטנזה לא אוהבת טקסט באנגלית, בפרט במקרה אילון מאסק שיסד את SpaceX. החברה קיבלה שם חדש מעשה ידיה של סטנזה – פפפפט.
- **'ארגון': ('אנדרוז', 'התגרשו')** - זה מה שקורה כשהמודל חושב ששם המשפחה של ג'ולי אנדרוז הוא ארגון במקום לתייג אותו כשם.
- **"נגלמין פרנקלין"** – סטנזה לא מחבבת במיוחד גרשיים, והגרש הוחלף באות ל' במחווה הפגנתית ביותר. כך יוצא ששמו הפרטי של בנג'מין פרנקלין הופך ל "ב + נגלמין".
- **'מקום': ('גפוליה', 'נולדה')** – דוגמה נוספת של חוסר האהדה של סטנזה לגרשיים. הפעם היא החליפה את הגרש ב"ג'וליה" באות פ', והתוצאה – מדינה חדשה בשם גפוליה.
- **אירוע': {('שואה', 'נולדה')}** – לא טעות, אך באופן מצחיק המודל פלט שנועה קירל נולדה בשואה. זה מה שהוא הסיק ממשפט שעסק גם בלידה שלה, וגם בעובדה שמצד אביה יש לה קרובי משפחה שנספו בשואה (החלק השני של המשפט היה שמני ולכן הפועל נותר "נולדה").
- **ט' בכסלו תרפ"ה - ("ט'", 'O'), ('ב', 'O'), ('כסל_', 'O'), ('_של_', 'O'), ('_הוא', 'O'), ('תרפ"ה', 'DATE_1')** – סטנזה מצאה את הדרך הכי מוזרה לפרק מילה סטנדרטית.

[8] *Neural Text Generation from Structured Data with Application to the Biography Domain - https://aclanthology.org/D16-1128.pdf*

סיכום

במהלך הסמסטר עבדנו על פרויקט שמטרתו לחלץ מידע בדיד מטקסט העוסק באישים. התוצר הסופי אכן עונה על הדרישות בצורה מוצלחת. עסקנו בעבודה עם כלים מגוונים מתחום ה-NLP בעברית כמו Stanza, NER, BERT וחיברנו את כולם ביחד לכדי מוצר עובד. העבודה שילבה גם עבודה עם טכניקות כמו תיוג של מידע, עבודה עם קורפוס, ואימון של מודל. קוד המעטפת של הפרויקט כלל הרבה לוגיקה ששימשה לחיבור הכלים השונים, לחילוץ המידע, לתיוג מידע ולהצגת המידע הסופי בצורה נוחה לצריכה. הפרויקט עטוף בחבילה פייתונית קלה להתקנה ולשימוש שמאפשרת עבודה גמישה ושילוב בסקריפטים בעתיד.

נספחים וקוד

חלקים מהפרויקט שביצענו נמצאים במקומות נפרדים מהדוח הנוכחי, ולכן נרכז את כל המקורות הנוספים תחת הכותרת הזו.

הקוד

הקוד משותף ב-GitHub - https://github.com/saifun/WikIE. הוראות התקנה ודוגמאות הרצה ניתן למצוא בקובץ ה-README.

ריכוז תהליך ההערכה

הסבר על התהליך וכן התוצאות המפורטות של התהליך נמצאים ב-GoogleDoc המצורף - https://tinyurl.com/wikie-eval.

הקורפוס

הקורפוס שהשתמשנו בו כדי לאמן את מודל האלף-ברט נוצר במעבדה של בר אילן - https://github.com/OnlpLab/NEMO-Corpus.

סקריפט תיוג המקצועות

הסקריפט נמצא גם הוא באותו ה-repository ב-GitHub, כחלק מהקוד לעבודה מול הקורפוס - https://github.com/saifun/WikIE/blob/main/src/hebrew_ner_model/ner_alephbert.py.