## סדנה בעיבוד שפה טבעית בעברית – פרויקט סופי

# כיף עם ויקיפדיה

## מריה צייטלין • 316984368 • unimaria

## סיפן נוה • 208647834 • nfun

# 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]](#footnote-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 IQD, 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.

# 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 העוסק באישיות, באופן אוטומטי, באמצעות NLP בעברית?״**

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

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

בכל הכותרות הבאות נעסוק בשאלת המחקר הראשונה.

## 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:

״אלברט איינשטיין נולד בגרמניה וגר בשוויץ״

שם: אלברט איינשטיין

מקום לידה: גרמניה

מקום מגורים: שוויץ

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

״אלברט איינשטיין היה גרמני שגר בשוויץ״

שם: אלברט איינשטיין

מקום לידה: גרמניה

מקום מגורים: שוויץ

# גישות ורעיונות + מה בחרנו בסוף

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]](#footnote-2) ניתן להשתמש בשאלות ותשובות על טקסט לביצוע מספר משימות NLP, שאחת מהן היא Information Extraction. הגישה נחקרה והוכחה ב-NLP בשפה האנגלית.

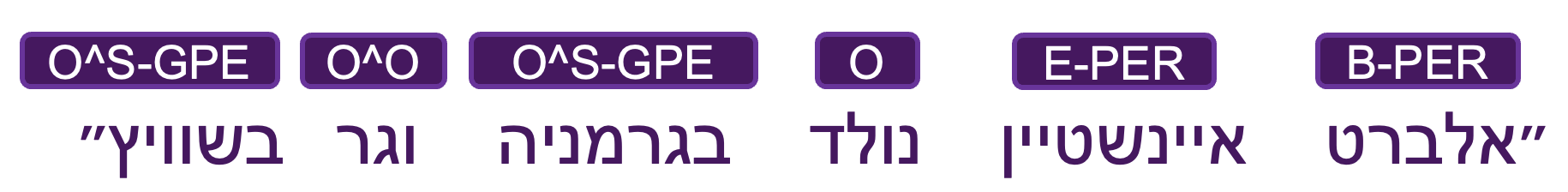
לאחר שבדקנו את הגישה, החלטנו שיישום שלה בעברית עשוי להיות מעניין ונוכל להשיג את המטרה שלנו באמצעותה. שאלות שבנויות מעל נשוא המשפט מאפשרות הבנה של יחסים במשפט, ונותנות מבנה אלגנטי של מידע בדיד. לדוגמה, במשפט ״אלברט איינשטיין נולד בגרמניה וגר בשוויץ״, שאלה כמו ״מי נולד?״ תוציא את שמה של הדמות שעליה מדברים 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 הצליח. לכן, החלטנו לחפש דרך חדשה לביצוע המשימה, ולהמשיך לחקור את כיוון השאלות והתשובות בעברית במחקר של 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 העברי. בחרנו 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:

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 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-NEMO2 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 חשוב 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 הקורפוס שהיינו צריכות להשתמש בו השתנה. הקורפוס שהשתמשנו בו בפרויקט הוא הקורפוס המוכן של NEMO-Corpus.[[3]](#footnote-3) 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 יצרנו רשימה של כ-350 מקצועות, וכתבנו סקריפט שיודע 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).

# המודל הנבחר

המודל שבחרנו בנוי בצורת פייפליין שהמידע עובר בו. מודלי ה-NLP שאנחנו משתמשים בהם הם 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 שנוספו עליו. נפרט על כל השלבים בפייפליין.

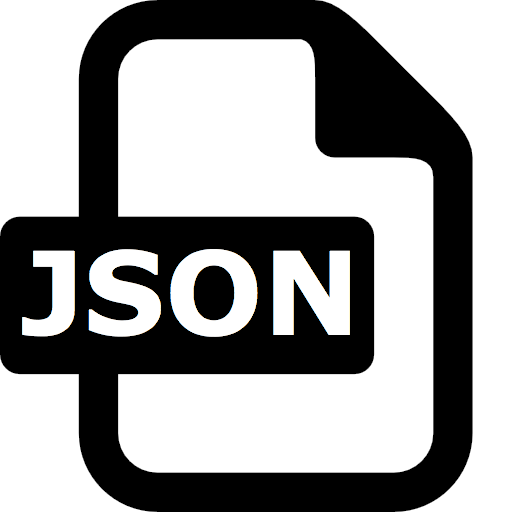
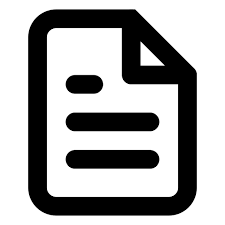
ניתוח מורפולוגי

אלף-ברט +

NER

חילוץ המידע

תוספות



ניתוח מורפולוגי: שלב הניתוח המורפולוגי הוא שלב קלאסי ב-NLP בעברית. בשלב הזה השתמשנו ב-Stanza לביצוע ניתוח סינטקטי וחלוקה מורפולוגית של הטקסט. השלב קיבל טקסט רציף כקלט, ומבין 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:

Text

Description automatically generated

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, העברנו את ה-data של NEMO-Corpus בשלב מקדים, שנועד לתייג אותו באופן אוטומטי, בהתאם לדרישות. רשימת 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 שלו לקורפוס המקורי, ולקבל אותם ״בחינם״ בפלט של המודל.

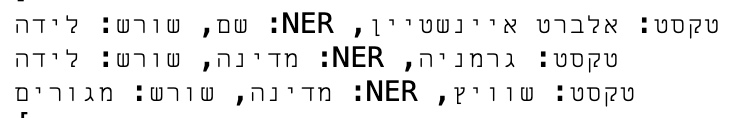
בבדיקת המודל הסקנו שהמידע מתויג בצורה אמינה יותר בשלב השימוש במודל אם מבצעים חלוקה מורפולוגית של המשפט לפני שלב התיוג, ולכן הטקסט נכנס לשלב הזה לאחר פרסור של Stanza. גם טקסט ללא פרסור של Stanza עובד כראוי, ואם לא היינו משתמשות ממילא בניתוח מורפולוגי בפייפליין, אז לא בטוח שהיה הכרחי להוסיף אותו רק בשביל השלב הזה.

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):



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.

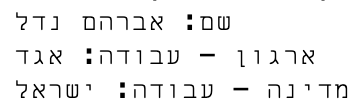


תוספות: בשלב הזה הוספנו חילוץ של שדות שאנחנו לא מעוניינים שהמודל ילמד, ונעדיף שיחולצו בדרך אחרת. דוגמה לשדה כזה היא תאריך. לתאריכים יש מבנה מאוד ספציפי שניתן לתפוס עם ביטוי רגולרי, ונעדיף לבצע התאמה מדוייקת על פני למידה, כאשר זו לא נחוצה, כדי להגדיל את הדיוק הסופי של הפייפליין.

פלט: לאחר העברת הטקסט ״אלברט איינשטיין נולד בגרמניה וגר בשוויץ״ בפייפליין, נקבל את הפלט



הקלט ״אברהם נדל עבד בחברת אגד כנהג אוטובוס בישראל״ יוביל לפלט:



במהלך העבודה התקבלו מספר החלטות כמו דרך ההצגה של הפלט. שקלנו להשתמש בהמרה של הפעלים לשמות פעולה, אך לא מצאנו API מתאים, ובניה של מאגר כזה היתה יקרה מדי.

# הערכה, איוולואציה

הערכה – איך מצאתם ערכים מספריים שמתארים את פתרון הבעיה? מה מדדתם?

# תוצאות

תוצאות – מה קיבלתן? האם יש תוצאות מפתיעות? איזה features/instances מעניינים מצאתן?

אריאנה גרנדה

# מסקנות

מסקנות – מה אומרות התוצאות לגבי שאלת המחקר?

# רעיונות לעתיד

רעיונות לעתיד – איך אפשר להרחיב את הפרויקט בהינתן עוד זמן וכוח אדם?

# 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 אף יותר, 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

# סיכום

# נספחים וקוד

1. לקוח מוויקיפדיה [↑](#footnote-ref-1)
2. *About the QA-SRL Project* - https://qasrl.org/ [↑](#footnote-ref-2)
3. *NEMO-Corpus on Github* - https://github.com/OnlpLab/NEMO-Corpus [↑](#footnote-ref-3)