האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

דקדוקית

סדנת עיבוד שפה טבעית בעברית | 67680 | פרוייקט מסכם

מגישים: רואי קריינר | נועם קסטן

תעודת זהות: 205731664 | 208289751

שם המרצה: מר יובל פיינשטיין

ספטמבר 2022

תוכן עניינים

1. הגדרת הבעיה

עברית הנה שפה "מחויבת מגדר", בשפות מסוג זה כל התייחסות לאדם או קבוצה מחייבת בחירה מגדרית בין נקבה לזכר, בפרט הכלל מאפיין מספרים בשפה העברית.

מחקרים מראים כי הטיה שגויה של שם המספר היא אחת מהשגיאות הנפוצות ביותר מבין שגיאות הדקדוק השגורות בשפה העברית. טעויות אלו נפוצות בעיקר בקרב קבוצות האוכלוסייה הבאות:

- ילדים
- עולים חדשים
- אנשים עם דיסלקציה

בשנים האחרונות ישנם מספר גורמים שמובילים לעלייה ניכרת בכמות השגיאות של הטיית שם המספר. ביניהם:

- שימוש הולך וגובר בכינויי "סלנג"
- כתיבה ופרסום ברשתות החברתיות
- חוסר בקיאות בשפה העברית
- שירים חדשים הכוללים שגיאות באופן מכוון, כחלק מתופעה תרבותית רחבה.

מטרת הפרויקט שלנו הוא לבדוק האם ניתן לזהות ולתקן שגיאות בשם המספר בשפה העברית באמצעות מודלים וחבילות NLP.

2. איתור וניקוי נתונים מתאימים

הערה מקדימה

המערכת שבנינו מכילה צד שרת, צד לקוח, קונטרולרים של ML שימוש בחבילת YAP ו REST API. לכן נשמח שהבודקים יריצו ידנית את המערכת כפי שמצוין בREADME.[1]לחילופין, נציע לצפות בסרטון הדמו של המערכת.[2]

מערכת ה"דקדוקית"

הפרויקט שלנו מציע כלי אשר נועד לפתור את הבעיה המתוארת לעיל, תוך מטרה להפחית באופן ניכר את כמות השגיאות בשפה העברית ולשמור על שפה תקינה ובהירה. על מנת לענות על צורך זה, יצרנו מערכת אשר מאפשרת תיקון שגיאות מגדריות בשם המספר:

- **מערכת**: צמצום שגיאות הטיה מגדרית של מספרים המופיעים במשפטים בתצורה מילולית.
- **תוסף**: המרת משפטים עם מספרים בתצורה מספרית לצורתם המילולית על ידי קלסיפיקציה מגדרית.
- **רכיב תוכנה ב REST API**: המרת מספר מצורה מילולית למגדרית.

איתור וניקוי נתונים מתאימים

ראשית נדגיש כי בפרויקט המוצע קיימות שתי אופציות:

א. תיקון משפט הכולל טעות בשם המספר.
ב. מציאת טעות בשם המספר במשפט על ידי מודל למידת מכונה.

עבור אופציה **א**, בנינו מודל אשר משלב שימוש ב-YAP על מנת לפרק את המשפט לחלקיו השונים (יורחב בסעיפים הבאים) ולאפשר שימוש תקין בחוקי השפה העברית.

עבור אופציה **ב**, הקורפוס האידיאלי בו היינו מעוניינים הוא כזה שמכיל את רשימה של משפטים שבהם ישנם שם עצם יחיד ושם מספר יחיד שנכתבו בשפה העברית ללא ניקוד, בנוסף היינו מעוניינים בתווית האם שם המספר במשפט הינו תקין.

---

[1] https://github.com/roykreiner-s/Final-Project-NLPH
[2] הסרטון זמין לצפייה כאן: https://github.com/roykreiner-s/Final-Project-NLPH/blob/main/System%20Demo.mov

לאחר ניסיון לאיתור הנתונים על מנת לבצע SCRAPING מאתרים רלוונטיים באינטרנט, הבנו שכיום לא קיימים בעברית אתרים אשר מכילים תצורה כזאת של נתונים. מלבד זאת, גם אם היו בנמצא אתרים המכילים רשימה ארוכה של משפטים הכוללים שמות של מספרים, נצטרך לתייג את תקינות המשפטים מבחינת שם המספר באופן ידני. מאגר תיאורטי שכזה היה דורש עבודה רבה בכל הנוגע ליצירת פורמט אחוד ונקי, ברמה שתאפשר את הזנת המאגר למודל NLP .

לכן החלטנו לג׳נרט (לייצר) באופן מלאכותי את מאגר הנתונים, שאבנו את ההשראה לרעיון זה משתי חברות ישראליות העוסקות בתחום של ייצור נתונים באופן סינטטי למודלים של למידת מכונה (חברת Datagen[3] וחברת Explorium[4]). בנוסף נפגשנו עם חבר לספסל הלימודים אשר עובד בחברת Mobileye ולוקח חלק בפרוייקט של יצירת נתונים סינטטיים (ופיקטביים) של תמונות רחוב עבור מודלי למידת המכונה שהחברה משתמשת בהם.

מאגר הנתונים שהחלטנו לייצר באמצעות המודל בקוד "data_generator" הינו מאגר הכולל רשימה של שם המספר ושמות עצם שונים.

למעשה מודל ג׳ינרוט הנתונים מצליב מספר אקראי עם שם עצם מתוך סל רחב של שמות עצם אפשריים אשר מצאנו במאגרים של ויקיפדיה ולאחר מכן המספר מתורגם לשם המספר בעברית בתצורות מגדריות שונות – זכר ונקבה. נוסף על כך, אנחנו מייצרים את המשפט התקין האמיתי עם שם המספר באמצעות המודל שבנינו לתיקון שגיאות בשם המספר (זוהי אופציה נוספת באפליקציה שיצרנו לתיקון שגיאות), ולבסוף אנחנו משווים את המשפט התקני הכולל את שם המספר למשפטים אשר ג׳ינרטנו בתצורות שונות של זכר ונקבה ונותנים תווית של אמת ושקר בהתאם לתקינות המספר.

כעת הגענו למצב בו אנחנו מחזיקים מאגר נתונים גדול יחסית עבורנו הכולל כ-10,000 רשומות של משפטים עם שם המספר שעליהם ניתן לבצע חיזוי.

3. <u>מיפוי ראשוני/סקירת ספרות</u>

כפי שתיארנו, הפרויקט מציע שתי חלופות לתיקון שגיאות בשם המספר בעברית וזיהוי שגיאה בשם במשפט הכולל את שם המספר.

עבור תיקון שגיאות בשם המספר בעברית ראשית ניגשנו לבחון את כללי שם המספר בשפה העברית:

הידע הנדרש בשפה העברית בהקשר של מערכת שם המספר דורש הבנה של המאפיינים הייחודיים לעברית אשר שונה באופן ניכר ממערכות וכללי דקדוק בשפות אחרות. כך למשל, אחת החידות הקשורות בשם המספר היא ההיפוך בצורת הזכר והנקבה מן הרגיל בעברית. בדרך כלל הסיומת "-ה" מציינת נקבה, כגון ילד-ילדה, טוב-טובה, הלך-הלכה, ואילו בשם המספר סיומת "-ה" מתייחסת דווקא לזכר: שבע (נקבה) – שבעה.

---

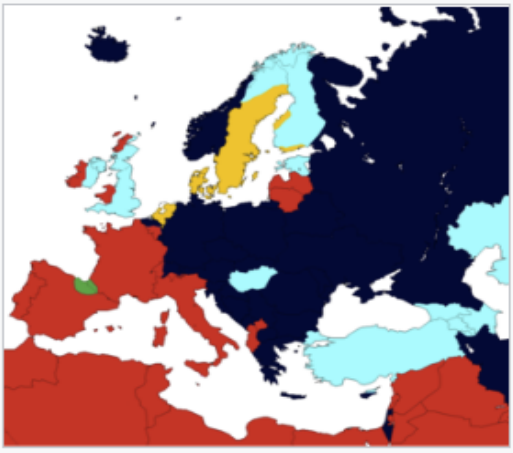
[3] https://datagen.tech/why-datagen/

[4] https://www.explorium.ai

ההתנהגות הלא צפויה הזאת משותפת לכל השפות השמיות וניתנו לכך הסברים שונים על ידי בלשנים וחוקרים. לפי אחד ההסברים, במילים שמציינות את שם המספר – שהן מילים בסיסיות בכל לשון – משתקף קו לשוני הקודם להתקבעות סיומות המין הדקדוקי.

כמו כן, מערכת שם המספר מורכבת ברובה משלשות של מספרים, אשר מרכיבים יחד מספר שלם. למשל: "שבע מאות עשרים ושלושה אלף ארבע מאות שישים וחמש". כללים נוספים ורבים חלים לגבי שם המספר וניתן למצוא את מרביתם בקישור המצורף.[5]

נראה כי מלבד הקושי שמתעורר בשפה העברית, הבעיה שזיהינו מהווה קושי גם בשפות רבות אחרות. המפה המצורפת מלמדת כי באירופה למשל, רק ב-30% מהשפות המדוברות אין מערכת מגדרית מובנית.[6] עם זאת, מערכת מגדר עבור שם המספר הינה מערכת מאוד ייחודית לשפה העברית.



Gender in European languages.
Light blue: no gender system.
Yellow: common/neuter.
Red: masculine/feminine.
Green: animate/inanimate.
Dark blue: masculine/feminine/neuter.
Standard Dutch has a three-gender structure, which fell in disuse in the North of the Netherlands but remains very much alive in Flanders and the South of the Netherlands.

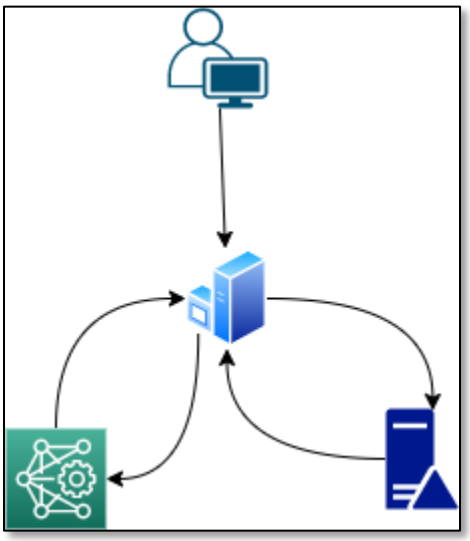
[5] https://www.safa-ivrit.org/dikduk/numbers.php

[6] https://en.wikipedia.org/wiki/Grammatical_gender

עבור זיהוי שגיאות בשם המספר במשפט בעברית, לא מצאנו גישות מקובלות בספרות, ולכן התייעצנו עם מכרים אשר עובדים בתחומים שונים של למידת מכונה, אשר בעצתם החלטנו לייצר כמויות עצומות של נתונים מלאכותיים, שיהוו מענה איכותי לבעיה שהצגנו.

4. המערכת המוצעת : "דקדוקית"

המערכת מורכבת מ-4 micro-services שמריצים צד שרת וצד לקוח של המערכת כדי לאפשר ממשק ידידותי למשתמש. הרכיבים מדברים אחד עם השני על ידי פרוטוקולי תקשורת של בקשות HTTP.[7]



רכיבי המערכת :

- **צד לקוח (Client Sevice):** נכתב ב REACT.js ומאפשר בצורה נוחה בדיקה והמרה של משפטים (המכילים מספרים בצורה מילולית ומספרית) למשפט הבנוי עם הטיה מגדרית נכונה.
- **צד שרת (Server):** נכתב ב Node.js כ-REST API - מכיל ראוטר לניתוב הפניות של הלקוח. ניתן להרחבה ואינטגרציה עם מכשירים אחרים בקלות.
- **קונטרולר (ML):** נכתב ב-numpy, pandas, Python3 ומטרתו לנקות את המשפט, לפרק אותו ולבצע קלסיפיקציה.
- **קונטרולר (Converter):** נכתב בPython3 כרכיב עצמאי שניתן להוסיפו לYAP או לכל מערכת אחרת בקלות. הרכיב ממיר מספרים מצורתם המילולית לצורתם המספרית וכן בכיוון השני.

[7] https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

בחירת מודל

שלב א׳ נסיון -

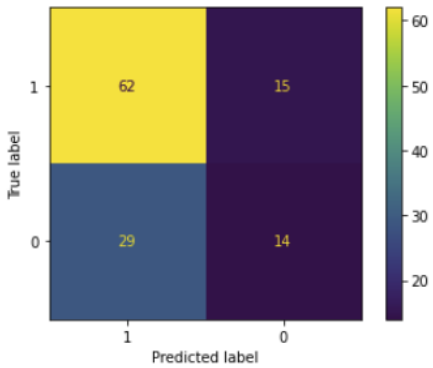
ראשית ניסינו לבנות מודלים של קלסיפיקציה כפי שלמדנו בשיעור ולאמן אותם על בסיס המידע שיצרנו באופן מלאכותי.

לאורך הפרויקט עבדנו עם שלושה מודלים : יער אקראי, לוגיסטי וקלסיפייר אקראי (הטלת מטבע).

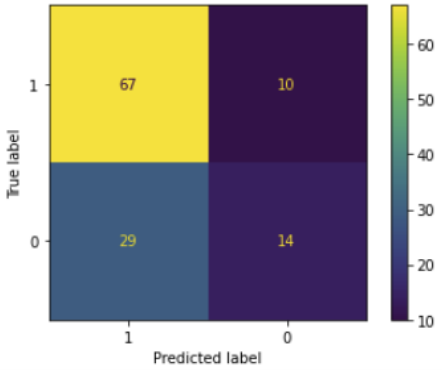
באיטרציה ראשונה הדאטה סט שלנו היה מורכב מ-300 מילים בלבד (sanity check) כדי להבין האם המודלים אכן מאפשרים להניב תוצאות רלוונטיות.

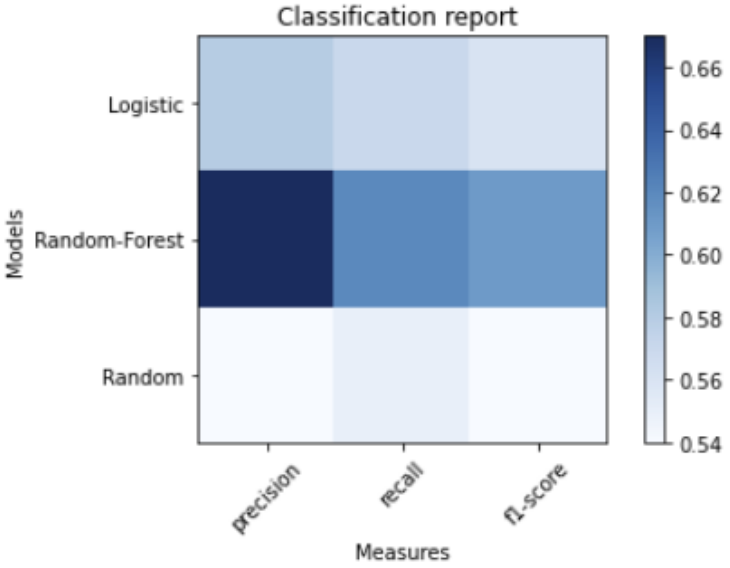
התוצאות של האיטרציה הן :

לוגיסטי



יער אקראי





לאחר מכן ביצענו איטרציה עם 10,000 והתוצאות היו זהות.

לבסוף בחרנו את מודל random forest ושמרנו את המודל כשהוא מאומן באמצעות **pickel** [8] (ספרייה בפייתון המאפשרת לשמור מודל למידת מכונה). כעת בעת פתיחה מחדש של המערכת נטען מחדש את המודל המאומן בכל פעם (ללא צורך באימון מחדש של המודל).

<u>המודל במערכת</u>

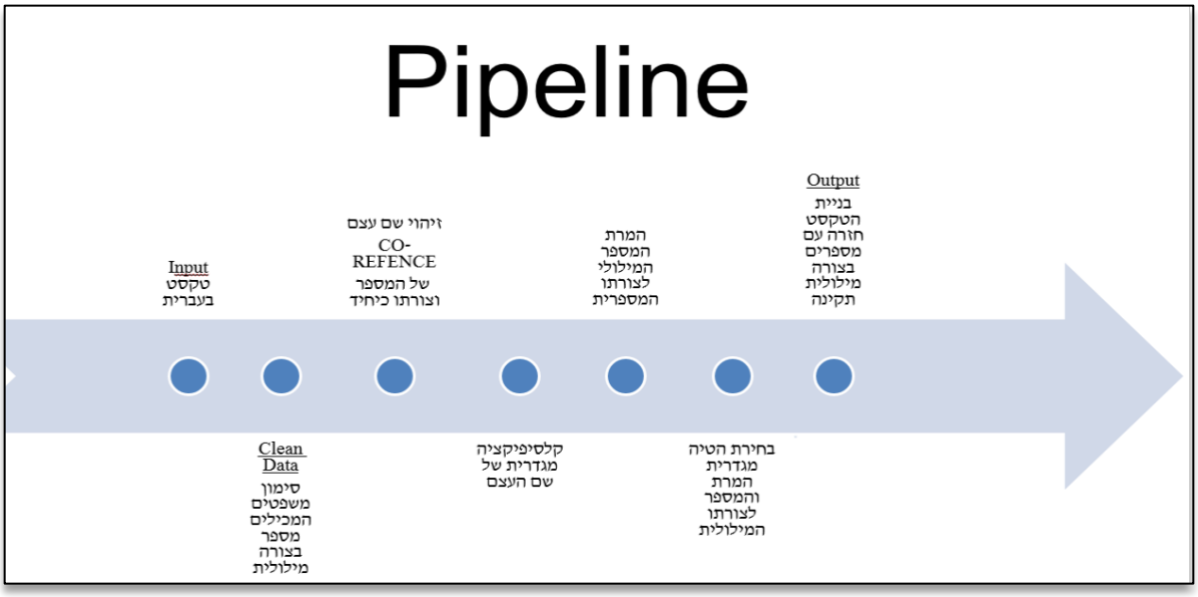
במהלך הפרויקט בנינו מערכת שמתקנת שגיאות בשם המספר אשר נבדקה על ידי מספר גישות.

המערכת מקבלת משפט, מנקה אותו, ולאחר מכן מפענחת את הסמנטיקה בו (פירוק על ידי YAP) לפי מורפולוגיה של השפה העברית – מזהה את המספר במשפט (בהינתן צורה מספרית או מילולית שלו) ומפענחת מהו/מהם ה- co-reference המיוחס/ים במשפט למספר.

לאחר שלבים ראשוניים אלו מתבצעת המרה של ה- co-reference לצורת יחיד, ולאחר מכן קלסיפיקציה (זכר / נקבה) של שם העצם. לבסוף מתבצעת המרה של המספר בהתאם לתווית של הדגימה לצורה מילולית על ידי קורפוס שבנינו שממיר מספרים מצורה מספרית למילולית והפוך עם הטיה מגדרית.

המודל פולט את המשפט עם המספר בהטיה מגדרית נכונה.

תרשים זרימה של המודל מוצג כאן:



[8] https://docs.python.org/3/library/pickle.html

טכנולוגיות שהשתמשנו בהם לבניית הפרויקט

- PICKLE
- NODE.JS
- NUMPY
- REST API
- REACT
- PYTHON
- YAP
- SKLEARN
- PANDAS

5. הערכה

המדידה שביצענו היא וידוא תיקון וזיהוי שגיאות בשם המספר בשפה העברית. ההערכה התבצעה על ידי תיקון וזיהוי משפט הכולל שגיאה בשם המספר. ניתן היה לבצע את ההערכה באמצעות הדאטא איתו אימנו את המודל, בנוסף במהלך הפרויקט הצלחנו ליצור מודל המתקן שגיאות בשם המספר באופן יציב באמצעות כללי השפה העברית בשילוב עם מערכת YAP (נציין כי בהתאם לשפה העברית יש יוצאי דופן אשר המודל לא מצליח לתקן). באמצעות מודל זה יכלנו להשוות את התוצאות אותם קיבלנו על ידי מודל למידת המכונה.

6. תוצאות

המערכת שיצרנו מזהה משפטים אשר כוללים מספרים הכתובים בעברית ומתקנת אותם בהתאם לחוקי הדקדוק של שם המספר בשפה העברית. עם זאת, כידוע השפה העברית הינה מורכבת מאוד, ולכן ישנם יוצאים מן הכלל אשר כעת המערכת שלנו לא תומכת בהם.

7. איטרציה

תחילה ניסינו לבצע SCRAPING של משפטים הכוללים את שם המספר ממספר מקורות, אך מעבר לכך שהתקשנו למצוא מאגר נתונים עם מספר רחב של משפטים, הבנו שיהיה אתגר בתיוג המשפטים (האופציה היחידה הייתה לבצע באופן ידני). בשל כך, החלטנו להשתמש בטכניקה חדשה ולייצר את הנתונים באופן מלאכותי כמו שפירטנו בסעיפים קודמים, ובנוסף לבצע את תיקון שגיאות במשפט באמצעות חוקי השפה העברית וכך גם לאמן את המודל לביצוע שגיאות וגם לייצר כלי אשר בו ניתן לתקן שגיאות בשם המשפט.

8. מסקנות

התוצאות מלמדות אותנו שניתן להשתמש במערכת שלנו כדי להרכיב מערכת מלאה לתמיכה בשם המספר העברי.

המערכת שלנו מתבססת על קלסיפקציה, מציאת תלויות וCO-REFERENCE לשם המספר אשר מבוצעת באמצעות YAP ולעיתים לא נכונה, ואנחנו מאמינים כי בהמשך כאשר YAP ימשיך להתפתח.

9. רעיונות לעתיד

- **Chrome Extension** : בניית תוסף לסריקת תוכן בעמוד אינטרנט וזיהוי/ המרה של מספר לתצורותו המילולית הנכונה.
- **Edge Cases** : תמיכה במקרי קצה כמו היינו 2 ילדים (כיום המערכת לא מצליחה להתמודד עם זה)
- **Voice Recognition** : הוספת אופציה למערכת של תיקון שגיאות בשם המספר בדיבור קולי. ישנו פער אדיר במערכות speach2text בשפה העברית כיום.
- **Game** : בניית תוכנה לשיפור כתיבה והבנה של בחירת הטיה מגדרית נכונה עבור שם המספר שתפעל בעיקר לעולים חדשים וילדים באמצעות ממשק מתאים.
- **Online Checks Editor** :ממשק נוח לכתיבת המחאות בצורה מדויקת לפי הטיה מגדרית
- **YAP Integration** : הוספת כלל המערכת ל-YAP.

10. שימושים ושיתופי פעולה אפשריים

המערכת שבנינו בנויה ממספר רכיבים נפרדים בצורה מודולרית ולכן ניתן להתייחס לכל אחד מהם בנפרד לטובת שיתופי פעולה אפשריים:

1. אינטגרציה עם **YAP -** ניתן לחבר לYAP את הקונטולר שממיר מספרים מצורתם המילולית למספרית בשני הכיוונים.
2. שיתופי פעולה עם בנקים - כאשר כותבים המחאות יש לכתוב בצורה מספרית ומילולית את הסכום. לפיכך ניתן לחבר את הקונטרולר שאחראי אצלנו בתוכנה על ההמרות לממשקים שלהם (אפליקציות/ אתר) כדי להקל על המשתמשים בעת הכתיבה.
3. מערכת החינוך - ניתן להשתמש במערכת שלנו כשכבה ראשונית במשחקים/ מדריכי לימוד של דקדוק השפה העברית.