עיבוד שפה טבעית תרגיל בית 1

מגישים: קרן חובב 208152439, עמית קדם ויצמן 203912506

אימון:

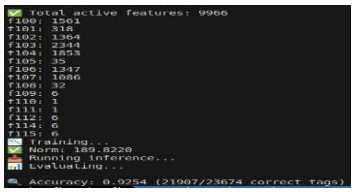
מימשנו את סט הפיצ'רים $f_{100} - f_{107}$ כפי שנלמד בהרצאה. בנוסף מימשנו את:

Features	Description (all = 1 when condition met AND t=Vt, else 0)
f108	Word capitalization: starts with capital OR all capitals
f109	Digit patterns: contains digits OR all digits OR decimal number
f110/f111	Punctuation: contains '.' (f110) OR contains '-' (f111)
f112	Word structure: length = l
f114/f115	Shape patterns: matches word shape (f114) OR short shape (f115)
f113	Domain-specific units: word matches measurement unit (mg, g, kg, ml)
f116	Acronym pattern: word consists of all uppercase letters (≥2 letters)
f117	Numeric patterns: matches number+unit format (e.g., 10mg, 5ml, 25%)
f118	Medical terminology: ends with biomedical suffix (ase, itis, emia, oma)
f119	Specific capitalization: only first letter capitalized (InitCap pattern)

שיפורים במודלים:

ראינו שיש שימוש במס' גדול מידי של פרמטרים וחלוקה לא שווה של שימוש בפיצ'רים. לכן ב106-107 קיצרנו את אורך המילה עליה מסתכלים(מילים קצרות הן לרוב מילות קישור ולכן "נתפסות" בפיצ'רים אחרים). ב102-103 השתמשנו במילים עם תחילית וסופית של לפחות 2 אותיות וגג 3 שכן אלו הכי נפוצות ורלוונטיות. בנוסף הגדרנו עבור הקורפוס של מודל 2 שהוא רפואי פיצ'רים ייעודיים שתופסות תכונות של טקסט מסוג זה.

הגדרנו פונקציה בשם *filter top scoring feature* שנותנת סקור לפיצ'רים ולקחנו את האחוזונים הכי גבוהים עם מינימום הופעות(היפר פרמטר). בפונקציה *preprocess train* הגדרנו threshold באופן ידני.



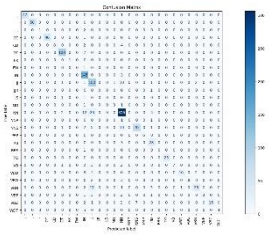
Model 2

```
✅ Best lambda value: 0.1 with average accuracy: 0.8968
🎯 Using optimal lambda: 0.1
🔧 Preprocessing full training data...
you have 4499 features!
```

הסקה:

מימשנו את אלג' Viterbi עם חיפוש Beam בגודל 5 ע"מ להקטין זמן הריצה . בנוסף מימשנו מנגנון pruning של soft margin לפלטר מסלולים באלגוריתם עם score נמוך מידי – מה שגם סייע באופטימיזציה של זמן הריצה.

מבחן:



על מנת להתמודד עם זה שאין לנו סט ולידציה במודל 2, השתמשנו באלגוריתם k-fold ומצאנו כך את ה=0.1 *lambda* האופטימלית שהביאה לאחוז דיוק מקסימלי בניסויים אמפיריים שעשינו. השתמשנו במטריצת הבלבול לזהות איפה יש הכי הרבה טעויות, וכך תיקנו ב hard code את משפחות הפיצ'רים הנ"ל.

תחרות:

מצד אחד במודל הגדול יש לנו גם סט אימון וגם סט ולידציה יחסית גדולים ביחס לסט המבחן .מצד שני ע"מ להתגבר על מגבלות שניתנו, בחרנו היפר פרמטרים וסיננו פיצ'רים באופן ידני מה שלדעתנו עלול לתרום לoverfit – כלומר בפועל אנו צופים שאחוזי הדיוק במבחן יהיו **נמוכים** יותר מהאימון: **85-90% דיוק**. יתכן ואם היינו משתמשים ביותר אלגוריתמים שעושים אופטימיזציה בכיול של היפר פרמטרים ובבחירת פיצ'רים היינו משפרים את הדיוק. לגבי המודל הקטן - חשוב לציין כי ההתמחות בטקסטים רפואיים גם מהווה חרב פיפיות - מצד אחד היא מאפשרת למודל להתמקד בדפוסים ייחודיים לתחום, אך מצד שני היא מגבילה את יכולתו לעבד טקסטים רפואיים בסגנונות שונים. למשל, טקסט רפואי אקדמי עשוי להיות שונה מהותית מטקסט המיועד לסטודנטים או לציבור הרחב. שוני זה עלול להתבטא בירידה משמעותית בביצועים כאשר המודל נחשף לסגנונות כתיבה רפואיים שלא נכללו בסט האימון. בשל כך גם במקרה זה אנו צופים לקבל אחוזי דיוק נמוכים יותר**80-85% דיוק** .