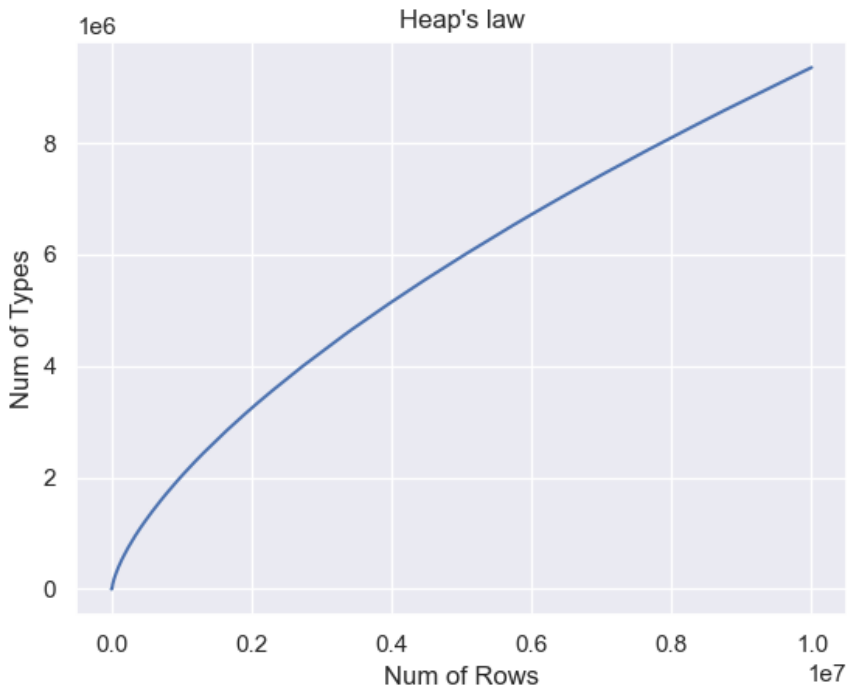
Amit Sindani 206902694

Raz Olewsky 315341396

Natural Language Processing – Assignment #1 - Solution

Task #1: testing the Heaps' law in natural language

1.



Heaps's law represents the relation between the number of tokens and the number of types in corpus. We switched the axis of the tokens with the rows and still got a valid result due to the increase respectively (growth in rows represents growth in tokens).

Task #2: statistical language models – solving a cloze

```
Mean-random chance accuracy:  0.08166666666666664
ML chance accuracy:  1.0
```

Mean-random chance accuracy represents the mean accuracy of 100 random solutions for the given cloze.

ML chance accuracy represents the chance accuracy we got from our solution.

בהנחיות המטלה נדרשנו לצרף את תהליך החישוב של ה-chance accuracy לקובץ main.py.

את החישוב מימשנו באמצעות 2 פונקציות:

compute_chance_accuracy – הפונקציה מחשבת את הדיוק בין הפתרון הנכון של ה-cloze (נמצא במשתנה candidates) לבין הפתרון המוצע (נמצא במשתנה res).

compute_mean_random_chance_accuracy – הפונקציה מחשבת את הדיוק הממוצע של 100 פתרונות רנדומליים.

בנוסף, בתוך הפונקציה solve_cloze, נמצאים בהערה מס' שורות קוד שבהם עשינו שימוש לצורך החישוב:

```
# candidates_list_copy = copy.deepcopy(candidates_list)
```

וגם

```
"""
print("Mean-random chance accuracy: ", compute_mean_random_chance_accuracy(candidates_list_copy))
print("ML chance accuracy: ", compute_chance_accuracy(candidates_list_copy, candidates_result))
"""
```

<u>הערות ונקודות שונות לבודקת התרגיל:</u>

1) בחרנו לממש את הקוד בצורה כזאת שבכל ריצה של התוכנית על cloze אחר, נייצר מחדש את מבני הנתונים של ה-bigrams וה-trigrams, במבנים אלה נשמור רק את ה-bigrams וה-trigrams שבהם קיימת לפחות מילה אחת השייכת לקבוצת המועמדים להשלמה. עשינו זאת כדי לעמוד בזמנים של 20 דקות פר ריצה, מה שלא התאפשר עבור מבנה נתונים של כלל ה-trigrams ב-corpus (הריצה של יצירת מבנה זה לקחה לנו יותר מחצי שעה).
2) עבור ה-smoothing לחישוב ההסתברויות בחרנו K=0.001, עבורו קיבלנו את אחוזי ההצלחה הגבוהים ביותר על ה-cloze שצורף למטלה ועל עוד 5 cloze-ים נוספים שהרצנו.