מעבדה בעיבוד שפות טבעיות

סמסטר ב' 2017

פרופ' וינטנר שולי

סביבת עבודה:

התוכנה פותחה במערכת הפעלה Ubuntu 16.04LTS ופועלת גם במערכת הפעלה Windows10. התוכנה נכתבה בשפה פייתון 3.5.4 גרסת אנקודה ועל מנת להריץ את התוכנה יש להתקין גרסה זו או להתקין את החבילות pandas ו-skleran דרך pip.

סביבת עבודה:

הוספנו לתוכנה קובץ קונפיגורציה (config.ini) דרכה ניתן לשנות שמות תיקיות, להריץ 'מוד בדיקות' ולאפשר הרצה במקביל בחלקים ב' ו-ג'.

* על מנת להריץ עבודה במקביל, יש צורך להגביל את מספר ה-cores בהתאם לכמות הראם שיש במחשב. בהתאם לכמות הchunks שיש בידנו, במחשב של 16 גיגה אפשר לעבוד על 2 ליבות במקביל.

חלק א':

התוכנית מחולקת לכמה חלקים:

* הורדת קבצי html של הסיפורים.
* יצירת קובץ json עם מטא דאטא אם קיים.
* הוספת מדטא דאטא לסיפורים שאינם מכילים שפות מקור.

על מנת להריץ את חלק, ניתן להריץ מתוך התיקייה Part\_a את הפונקציה main\_part\_a.py אשר מכיל את 3 החלקים שכתבנו למעלה.

על מנת להוריד את כל הסיפורים מכול אחד מן האתרים, כתבנו את המחלקה StoriesURL אשר מקבלת 2 נתיבי תיקיות:

* הנתיב הראשון – מקבל את התיקייה Website Stories Page, אשר שומר בתוכו את כל הקישורים ושמות הסיפורים (במקרה של dortome זה קובץ txt שנשמר כhtml-).
* הנתיב השני – מקבל את התיקייה Data Stories Page אשר שומר עבור כל מן האתרים את קובץ ה-html של הסיפור באתר.

המחלקה מקבלת את המשתנה download\_new\_stories\_flag, אשר ניתן לערוך דרך קובץ הקונפיגרציה.

על מנת לחלץ את המידע מקבצי html של הסיפורים הגדרנו את המחלקה HTMLParser אשר יודעת לחלץ את המידע באופן אוטומטי ובדיוק די גבוהה (היה התאמה רבה ועובדה פיזית על למצוא את כל הלוגיקה של האתר – יש באתרים מסוימים מבני html שונים מסיפור לסיפור).

לאחר שיש בדינו את המידע של הסיפור, אשר שמור כקובץ json יחד עם מטא דאטא (שם הכותב, שפה, שם המתרגם וכד'), גילנו שכמעט חצי מן המידע אינו מכיל שפת מקור. לכן, עבור חלק זה הגדרנו את המחלקה RenameAnddDetails אשר מקבל קובץ unknown stories dictionary.txt ואת התיקיה Part\_A\Json Stories\unknow (תיקייה שמציינת שאין שפת מקור לסיפור)

מבנה קובץ unknown stories dictionary.txt, נראה ככה:

15 שעות : דייוויד קסלר : טל ארצי : אנגלית

אדיוס, המינגוויי (אדיוס, המינגווי) : ליאונרדו פדורה : טל ניצן : ספרדית

הקובץ חייב להכיל בדיוק ':', החלק הראשון זה שם הסיפור, החלק השני זה שם המחבר, החלק השלישי שם המתרגם והחלק האחרון זה שפת המקור.

במידה וקיים '()' בחלק הראשון, השם בתוך הסוגריים מציין את השם הקובץ החדש.

חלק ב'

על מנת להריץ את חלק זה, בדיוק כמו בחלק הקודם יש להריץ את הקובץ main\_part\_b.py.

באותו תיקיה Part\_B יש את הקובץ corpos\_split\_into\_rows.py, אשר תפקידו לעבור על כל הסיפורים של אותו שפה ולפצל לשורות על פי התווים הבאים:

';', '.', '?!', '!?', '!', '?', ':'.

עבור חלק זה הגדרנו 2 מחלקות נוספות, מחלקה Vectors ו-Classify.

עבור חלק זה, הגדרנו כמה משתני "על" בקובץ הקונפיגרציה:

* מצב בדיקה - במידה ומשתנה part\_b\_debugging\_mode הוא True, אנחנו עוברים על דטא ישן (במידה והוא נשמר תחת התיקייה Debugging Files) והוא עובר על חלק קטן של הווקטורים ועליהם עושה את הריצה. בנוסף, במידה ואני במצב בדיקה, אנחנו לוקחים כמות ווקטורים לפי ערך המשתנה feature\_vectors\_size.
* גודל chunk – ניתן להגדיר אותו לכל גודל על ידי שינוי המשתנה chunk\_size בקובץ הקונפיגרציה.
* הרצה במקביל – במידה ומשתנה multi\_process הוא Ture, התוכנה תעובר להרצה במקבילץ
* הדפסה של התוצאות לקובץ – במידה ומשתנה print\_predict\_analysis\_flag הוא True, בתיקייה יופיע הקובץ classifier\_analysis.txt אשר מפרט את תוצאות עבור כל kfod עבור כל אחד מן המסווגים.
* כמות הליבות – ניתן לקבוע את כמות הליבות לפי המשתנה cores\_size ולמקסימום מספר הליבות שיש במחשב.

חלק זה דומה לחלק ג', השוני היחידי, שבחלק ג' דרשנו שחלוקה של kfold תכיל בכול חלוקה דוגמא של טסט, עבור כל אחד מן הקורפוסים.

לכן, במידה וגודל החלוקה של kfold גדול ממספר ה-chumks, התוכנה תודיע הודעה ותצא מן התוכנית.

חלק ד'

על מנת להריץ את חלק זה, בדיוק כמו בחלק הקודם יש להריץ את הקובץ main\_part\_d.py.

הרצת חלק זה מספקת 3 תשובות:

* דיאגרמת העץ, שעליה ממוספרים, אשר תופיע בחלון נפרד
* תודפס רשימת שמות השפות שהשתתפו, לפי סדר מספריהם בעץ. כלומר העלה הממוספר 0 יהא המקביל לשפה הראשונה ברשימה, והעלה הממוספר 5 יהיה המקביל לשפה השישית ברשימה.
* יודפס ערך ה Cophenetic Correlation Coefficient של החלוקה. זאת לצורכי בירור טיב החלוקה.

חלק זה משתמש בהגדרות מיקום תיקיית קבצי השפות, תיקיית הדפסות ה debug, וגודל ה chunk; בדומה לסעיפים הקודמים.

עם זאת, רשימת השפות המשתתפות מצוייה בתוך הסקריפט:

המשתנה used\_languages נועד לתיוגים פנימיים.

המשתנה corpora\_path\_files מחזיק את הנתיב היחסי של קובץ הטקסט של כל שפה, ביחס לתיקיית קבצי השפות.

דבר אחרון, הגדרנו מחלקה נוספת בשם Graphs אשר מדפיסה עבור כל קורפוס את תוצאות המסווג בניהם לבין עצמם באותו הקורפוס.

כלומר, רצינו לראות כמה באילו קורפוסים המסווג טועה ואיזה מסווג יותר טוב מהשני.