**תיאור המנוע**

מטרת המנוע- לסמן מילים לפי אחת מ-5 קטגוריות-

1. שם של אדם
2. שם של חברה / ארגון
3. שם של מיקום
4. שם כלשהו (NE)
5. מילה רגילה

**תיאור כללי**

נתאר את ה-flow הכללי שעובר הטקסט במנוע שלנו, ובהמשך נרחיב על כל שלב-

1. טוקניזציה- חלוקה לשורות והפרדת סימני פיסוק
2. עבור כל שורה-
   1. מתחילים ב-ngram מקונפג כלשהו
   2. לכל n-יה שעדיין מסווגת בתור מילה רגילה-
      1. מבצעים תיקונים והרחבות למונח (הוצאת סימני פיסוק, הסרת stop words מהקצוות)
      2. מריצים את כלל המסווגים במערכת
      3. קובעים את התיוג המנצח
      4. אם קיבלנו שם, מנסים להרחיב את המונח- בהתחלה אחורה ואז קדימה
   3. מייצאים את השורה המתוייגת כפלט

**טוקניזציה**

**עיבוד שורות**

**מסווגים**

התשתית מאפשרת הפעלה של כמות מסווגים שונה כרצוננו, כאשר ציוניהם משוקללים יחד.

כל מסווג צריך לממש ממשק קבוע-

* פונקצית סיווג- מקבלת מונח (string, מילה אחת או יותר) ומחזירה ציון לכל אחת מהקטגוריות (אדם, חברה, מיקום ומילה רגילה)
  + חלק מהמסווגים יודעים לתת אינדיקציה רק להיות המילה שם או לא שם, ולא את הסוג שלה. במקרה כזה, המסווג יתן ציון זהה לקטגוריות המציינות שמות, וציון אפס לקטגוריית מילה רגילה
* פונקצית shutdown לסגירה מסודרת של המסווג (למשל- שמירת cache לקובץ)

**קביעת התיוג המנצח**

באופן כללי, סוג התיוג שקיבל את הציון הגבוה ביותר הוא התיוג המנצח. עם זאת, ברור כי אם נשתמש במסווגים שונים עם משקלות שונים, ייתכן שנגיע לסיווגים קרובים מאוד זה לזה, ואז סטיה קטנה לכיוון מסויים לא אמורה לשנות את ההחלטה מקצה לקצה. לכן, אם יש כמה ציונים גבוהים קרובים זה לזה, אזי המונח מקבל את התיוג- "NE", כלומר- מדובר בשם, לא ידוע מאיזה סוג.

הבדיקה נעשית לפי המדד של כמה מסווגים נתנו ציון שהוא מעל לאחוז מסוים (קונפיגובילי) מהציון המקסימלי.

**הרחבת חלון**

**תיאור המסווגים**

**1. UpperCaseClassifier**

מסווג זה בא לבדוק את המאפיין הבסיסי והשכיח ביותר של שמות- התחלת כל מילה ב-Capital Letter. אנו כן מאפשרות למילה שהיא stop word להתחיל באות קטנה (למשל- United States of America).

מסווג זה לא מבחין בין סוגי ה-NE השונים, ולכן אם מונח מקיים את התנאי למעלה, המסווג נותן ציון אחיד לאדם, חברה ומיקום (ציון מקונפג). במידה ויש מילה שאינה חוקית, המסווג נותן ציון קבוע (מקונפג) לקטגורית "מילה רגילה".

**2. DuckDuckGoWordOccurrenceClassifier**

מסווג זה משתמש במנוע החיפוש של DuckDuckGo, ומחפש תדירות מילות מפתח אינדיקטיביות בהתאם לקטגוריה. למשל, אם נחפש שם של אדם, כנראה יופיעו בפסקה חלק מהמילים- הוא/היא, גיל, נולד, בן/בת. באופן דומה, אם נחפש שם של חברה נקבל מילים שקשורות לעסקים, למטה החברה ולייסוד שלה, ואם נחפש שמות מקומות נמצא מילים אינדיקטיביות לערים, איזורים, ארצות וכו'.

כמו כן, מסווג זה משתמש ב-cache על מנת לחסוך עומס של חיפושים חוזרים ורצופים.

מסווג זה פועל בכמה שלבים, אותם נתאר כעת.

**א. חיפוש ב-DuckDuckGo לקבלת פסקת תיאור**

DuckDuckGo הינו מנוע חיפוש המאפשר שימוש ב-API חינמי וללא הגבלות נראות לעין. בניגוד לחיפוש של גוגל, לא חוזרת ממנו רשימה של לינקים, כי אם ישויות מסוגים אחרים (ייתכן שלא יחזרו כולם, ואפילו לרוב זה המצב)-

* abstract- תיאור כללי של המונח. לעיתים, הפסקה הראשונה מתוך ערך הוויקיפדיה המתאים
* definition- הגדרה מילונית של המונח המבוקש
* answer- עבור חיפוש תשובה לשאלה
* results- אתרים קשורים שונים (בדומה לגוגל)- לרוב רק אחד
* related- מונחים קשורים. בפרט עבור מילים עם כמה משמעויות שונות, יופיעו כמה ערכים פוטנציאליים מתאימים (למשל- apple החברה ו-apple הפרי)
* redirect- העברה לנושא רלוונטי אחר

לחלק מהסוגים הללו יכולים להיות גם categories ו-topics רלוונטיים.

השתמשנו בממשק python חינמי ל-duckduckgo, שמחזיר אובייקט היררכי המכיל חלק מהשדות הללו לכל חיפוש. בנוסף לחיפוש רגיל, הממשק מכיל פונ' חיפוש נוספת בסגנון "I Feel Lucky" שמחזירה תוצאה אחת אחידה שמוצעת כטקסט הרלוונטי ביותר.

הקוד שלנו מבצע את שני סוגי החיפוש הללו, ומאחד את התוצאה לכדי פסקה מאוחדת אחת המתארת את המונח המבוקש. ישנה התייחסות פרטנית למקרים השונים בהם קיימים רק חלק מהשדות בתוצאה.

הממשק תומך בריבוי שפות לחיפוש, אך לא כל החיפושים קיימים בכל השפות, ואם חיפוש לא קיים, חוזרת התוצאה באנגלית.

**ב. Preprocessing לפסקת התיאור**

פסקת התיאור שחוזרת מ-duckduckgo מכילה סימני פיסוק שונים שיש לסנן לפני חיפוש מילות מפתח.

כמו כן, הפסקה כוללת לינקים שונים, כאשר ה-domain ב-duckduckgo והסיומת הנובעת מהחיפוש לא מעניינים אותנו, ולכן יש לפשט את הלינקים, ולחלץ רק את המונח המסתתר בתוכם.

בנוסף, ישנו באג בחיפוש של duckduckgo שבמקרים מסויימים מצמיד בטעות מילה למילה שאחריה, ואילו בלינק הנלווה המילים מופרדות ע"י התו \_, לכן חשוב כן לפרסר את הלינקים ולחלץ מהם מידע שאולי אבד בטקסט הרגיל בעקבות החיבור הלא נכון של המילים.

**ג. חיפוש מילות מפתח**

בשלב זה אנו סופרים את כמות המופעים של כל מילה אינדיקטיבית בהתאם לקטגוריה. נציין שאם המשתמש חיפש כמונח מילה אינדיקטיבית בעצמה, אזי נתייחס אליה בתור מילה רגילה. למשל, המילה age היא אינדיקטיבית לאדם, אבל אם המשתמש חיפש את המילה age בעצמה, הוא יקבל הרבה תוצאות לא רלוונטיות. המילים האינדיקטיביות אינן שמות בעצמן (אבל כן יכולות להיות חלק משם, כמו New York City), ולכן ניתן הציון בהתאם.

מאחר וחלק מהמידע שחוזר מ-duckduckgo הינו קטגוריה, נושא כללי או נושא רלוונטי קשור, מתבצע חיפוש גם על מילות מפתח ברבים- למשל, ייתכן וחברה תתואר ע"י המילה company וגם תופיע תחת הקטגוריה companies.

**ד. נירמול**

הגדרנו רף קונפיגורבילי מינימלי למילים אינדיקטיביות בקטגוריה שמעידות על שייכות לקטגוריה. רף זה נקבע לאחר הרצות על ביטויים ומונחים שונים.

אם בקטגוריה מסוימת יש פחות מכמות זו של מילים, הציון בקטגוריה הזו נחתך לאפס.

אם לבסוף בכל הקטגוריות המונח קיבל ציון אפס, אזי כנראה מדובר במילים רגילות ולא שם, וניתן ציון גבוה בקטגורית "מילה רגילה".

**3. CompanyDuckDuckClassifier**

מסווג זה מסתמך על מנוע החיפוש DuckDuckGo, אך משתמש בו באופן אחר מהחיפוש הקודם.

מצאנו כי שמות של חברות עשויות להחזיר disambiguation של אפשרויות שונות (למשל- Apple הינו חברה וגם תפוח). עם זאת, אם נחפש Apple Inc. או Apple company, נגיע לפסקת מידע ייעודית לגבי החברה. מעניין עוד יותר, שאם נחפש מילה אחרת בתוספת Inc או company, שאינה מתארת שם של חברה, נקבל תוצאה ריקה בחיפוש.

לכן אנו מבצעות חיפוש דומה לקבלת אינדיקציה האם מונח מייצג חברה או לא- מצרפים לו אחת מרשימת מילות מפתח רלוונטיות, ובודקים כמה מהן גרמו להחזרת תשובה כלשהי ממנוע החיפוש.

מונח שכאשר מצרפים לו מילת מפתח כמו inc לא מחזיר תשובה בחיפוש, הוא כנראה לא שם של חברה.

גם מסווג זה משתמש ב-cache על מנת למנוע חיפושים כפולים.