בניית מנוע לאחזור מסמכים – חלק ב'

**שינויים שנוספו במחלקות קיימות**:

* במחלקה ReadFile יצרנו קובץ – בו עבור כל מספר מסמך שמרנו את הנתיב (בתיקייה הפנימית של הפרויקט). קובץ זה יעזור לנו, כפי שיתואר בהמשך, לתמצת את 5 המשפטים המשמעותיים ביותר במסמך.
* במחלקה Indexer:
  + נעשתה שמירה של קבצי ה-posting בתיקיות שונות – עבור stemming/ללא stemming.
  + נשמר שדה ה-idf עבור כל term שנמצא במילון.
  + רשימות ה-posting שיצרנו בחלק א' נשמרו ב-6 קבצים (כפי שתואר בדוח עבור חלק א'). בחלק ב', ביצירת ה-inverted index, יצרנו 27 קבצי posting סופיים – אחד עבור כל אות ואחד עבור terms שאינם מתחילים באות.

בהתאם, גם קריאת שורה מקבצי ה-posting נעשתה בצורה שונה.

**מחלקות שנוספו בחלק זה**:

Ranker:

המחלקה שמדרגת את המסמכים בהתאם לשאילתה שהתקבלה.

לטובת חישוב הנוסחאות נשמרו במחלקה זו מבני נתונים עבור:

* Doc Weight – "משקל" המסמך – כלומר: (סכום משקלי כל ה-terms במסמך j).

ערך זה מחושב עבור נוסחת CosSim – ומכיוון שהוא ניתן לחישוב מראש ב-Offline, החלטנו לשמור אותו מראש ולהימנע מחישוב שלו בזמן אמת (בזמן בקשת שאילתה).

* Doc Max TF – תדירות ה-term השכיח ביותר עבור כל מסמך
* Docs Length – אורך כל מסמך
* בשיטה CalculateWeight נעשה חישוב "משקל" המסמך שתואר, ובנוסף חושב שדה

avgDL – אורך ממוצע של מסמך עבור נוסחת הדירוג BM-25 (יפורט בהמשך).

* הלוגיקה העיקרית, ובפרט חישוב הדירוג של כל מסמך בהינתן השאילתה, בוצע בשיטה RateDocs.

פירוט נוסחת הדירוג הסופית:

Sim(dj, q) = 0.4∙(CosSim(d, q)) + 0.6∙BM-25(d, q))

לאחר חישובים רבים – אלו המקדמים שהביאו לתוצאות המיטביות ולהחזרה של מספר גדול של מסמכים רלוונטיים.

פירוט הנוסחאות:

השימוש ב-CosSim הוגדר לנו במסמך המטלה, הנוסחה נלמדה במסגרת הקורס.

הפרמטרים של BM-25:

* + |d| - אורך המסמך d.
  + avgDL – אורך מסמך ממוצע במאגר (חושב ב-CalculateWeight).

לאחר מספר ריצות, ראינו כי התוצאות הטובות ביותר התקבלו עבור k1 = 1.3 (פרמטר k1 מקבל ערכים בין 1.2 ל-2).

לאחר דירוג המסמכים – נעשה מיון שלהם לפי הדירוג ו-50 המסמכים בעלי הדירוג הגבוה ביותר הוחזרו.

במידה ומריצים קובץ שאילתות (בפורמט שניתן לנו) – מלבד השאילתא, המצויה תחת התגית title, הוספנו לשאילתה גם את ה-description (לאחר שהבחנו כי הוספה זו שיפרה את התוצאות המוחזרות).

Searcher

במחלקה זו בוצעו השאילתות.

* השיטה ParseQuery מקבלת כקלט את מחרוזת השאילתה ושולחת אותה לפירסור. מהפלט של הפרסר נקבל את ה-terms שבשאילתה.

קראנו מהדיסק את רשומות ה-posting של ה-terms, ומהם יצרנו קבוצה של מסמכים אשר מכילים את ה-terms.

* בשיטה ReturnDocs נעשתה קריאה לשיטה RankDocs במחלקה Ranker, אליה שלחנו את ה-terms שבשאילתה, את קבוצת המסמכים המכילים אותם (עם פירוט tf עבור כל מסמך) – וממנה קיבלנו רשימה של 50 מסמכים שהוחזרה כ-50 המסמכים הרלוונטיים ביותר עבור השאילתה.
* השיטה FiveSents טיפלה בדירוג 5 המשפטים המשמעותיים במסמך.

השיטה קיבלה כקלט את מספר המסמך, ובאמצעות הקובץ שנוסף במחלקה ReadFile (כפי שתואר לעיל).

אלגוריתם לבחירת 5 המשפטים:

1. פצל את המסמך למשפטים (כעת לכל משפט יש מספר משפט)
2. שלח כל משפט ל-Parser ושמור את רשימת ה-terms שהוא מכיל
3. עבור כל משפט:
   1. עבור כל term במשפט:
      1. הוסף את ציון ה-term לציון המשפט
4. מיין את רשימת ה"משפטים-ציוני משפטים" לפי ציוני משפטים
5. עדכן את הרשימה כך שתחיל רק את 5 המשפטים הראשונים (שלאחר המיון – יהיו למעשה 5 המשפטים בעלי הדירוג הגבוה ביותר)
6. עבור הרשימה המעודכנת (מכילה את 5 המשפטים בעלי הציונים הגבוהים ביותר), דרג את הגבוה מבניהם ב-1, השני הכי גבוה ב-2, וכן הלאה כך שהמשפט ה-5 יקבל כעת את הציון 5.
7. מיין את רשימת ה"משפטים-ציוני משפטים" לפי מספר המשפט
8. החזר את 5 המשפטים שברשימה לפי הסדר (לפי מס' המשפט), כשליד כל משפט ציונו

ראשית חילקנו את המסמך שקיבלנו כקלט למשפטים שהוא מכיל (נעשה בשיטה DocToSents – אשר קיבלה כקלט את הטקסט של המסמך והחזירה List שבה כל תא הוא משפט במסמך).

כל משפט נשלח לפירסור בכדי לקבל את רשימת ה-terms שהוא מכיל. ציון כל משפט חושב, כאמור, לפי ציון ה-term שבמשפט, אשר חושב בצורה הבאה:

כלומר ציון של כל term נקבע:

* 95% בנרמול לפי המסמך – כלומר: .
* 5% לפי האם ה-terms הוא אחד מ-25 ה-terms הראשונים במסמך.

נזכיר כי עוד בחלק א' של בניית המנוע, מופע של term במסמך אופיין לפי כמות המופעים של ה-term במסמך (tf), ובשדה בוליאני שנקרא isHeader – האם המופע הראשון של ה-term היה באחד מ-25 ה-terms הראשונים במסמך.

**השדה הבוליאני הומר למס' (0/1) וכך נכנס לנוסחת המשקל.**

נדגיש – הציון של כל משפט נקבע לפי דירוג כל ה-terms שהמשפט מכיל. דירוג ה-terms נעשה, כפי שניתן לראות בנוסחה, בצורה לוקאלית עבור המסמך – ללא התחשבות במופעי ה-term במאגר כולו (כלומר idf לא בא לידי ביטוי בנוסחה).

לאחר חישוב הנוסחה, מיינו את המשפטים לפי דירוגם, ולפי הדרישה – החזרנו אותם לפי סדר הופעתם ובצירוף הדירוג כך שהמשפט שקיבל את הדירוג הגבוה ביותר צוין עם "Score: 1", ואילו המשפט ה-5 בדירוגו עם "Score: 5".

* השיטה WikiExp ביצעה את הפונקציונליות של הרחבת שאילתה באמצעות וויקיפדיה.

השיטה מקבלת כקלט את השאילתה עצמה (המכילה מילה אחת בלבד), יוצרת WebClient ונעזרת ב-API של וויקיפדיה בשם WikiSynonyms על מנת למצוא קטגוריות הקשורות למילה שבשאילתה.

מהשרת חוזר טופס בפורמט json, ממנו חילצנו מונחים נוספים הקשורים למילה שבשאילתה. מונחים נוספים אלו שורשרו לשאילתה הראשונית, כך שקיבלנו שאילתה מורחבת.

מכאן – השאילתה המורחבת עוברת תהליך זהה לשאילתה רגילה.

האלגוריתם:

1. קלוט את השאילתה
2. חפש את המילה שבשאילתה ב- WikiSynonyms
3. קבל את טופס התוצאות בפורמט json וחלץ ממנו מונחים דומים למילה שבשאילתה
4. שרשר את המונחים הדומים לשאילתה (כעת השאילתה מורחבת)
5. המשך תהליך עבור שאילתה רגילה

Json

תפקידה לבנות אובייקטים שיכילו את המידע שהתקבל מ-WikiSynonyms. הבנייה נעשית על פי מבנה הפורמט json, כך שנוכל לקרוא את הנתונים בקלות.

המחלקה נבנתה באמצעות האתר [www.json2csharp.com](http://www.json2csharp.com).

המחלקה מכילה 2 תתי-מחלקות:

* + Term:
    - מכילה מחרוזת עם מילה מאותה קטגוריה של המילה שבשאילתה
  + RootObject:
    - מכילה את המידע על העמוד בפורט json.
    - מכילה List של Term.

**שימוש בקוד פתוח**:

* במחלקה json – כאמור – באמצעות האתר [www.json2csharp.com](http://www.json2csharp.com).

מדדים:

* **עבור ריצה עם stemming**:
  + Recall כולל – 0.151
  + Precision כולל – 0.25

|  |  |
| --- | --- |
| **Precision@N** | |
| 0.24 | 5 |
| 0.24 | 10 |
| 0.226 | 15 |
| 0.251 | 30 |
| 0.125 | 100 |

* + למנוע כולו:

* + עבור כל שאילתה:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Recall** | **Precision@N** | | **Precision** | **Query** | **Query ID** |
| 0.354 | 0 | 5 | 0.34 | Falkland petroleum exploration | 351 |
| 0.1 | 10 |
| 0.266 | 15 |
| 0.266 | 30 |
| 0.34 | 50 |
| 0.016 | 0.2 | 5 | 0.08 | British Chunnel impact | 352 |
| 0.2 | 10 |
| 0.1 | 15 |
| 0.133 | 30 |
| 0.08 | 50 |
| 0.392 | 0.2 | 5 | 0.4 | blood-alcohol fatalities | 358 |
| 0.2 | 10 |
| 0.4 | 15 |
| 0.366 | 30 |
| 0.4 | 50 |
| 0.214 | 0 | 5 | 0.12 | mutual fund predictors | 359 |
| 0.1 | 10 |
| 0.133 | 15 |
| 0.166 | 30 |
| 0.12 | 50 |
| 0.205 | 0 | 5 | 0.16 | human smuggling | 362 |
| 0 | 10 |
| 0.066 | 15 |
| 0.1 | 30 |
| 0.16 | 50 |
| 0.08 | 0.2 | 5 | 0.3 | piracy | 367 |
| 0.3 | 10 |
| 0.2 | 15 |
| 0.266 | 30 |
| 0.3 | 50 |
| 0.125 | 0.4 | 5 | 0.04 | encryption equipment export | 373 |
| 0.2 | 10 |
| 0.133 | 15 |
| 0.066 | 30 |
| 0.04 | 50 |
| 0.107 | 0.8 | 5 | 0.44 | encryption equipment export | 374 |
| 0.7 | 10 |
| 0.533 | 15 |
| 0.4 | 30 |
| 0.44 | 50 |
| 0.333 | 0 | 5 | 0.24 | cigar smoking | 377 |
| 0.2 | 10 |
| 0.133 | 15 |
| 0.266 | 30 |
| 0.24 | 50 |
| 0.285 | 0 | 5 | 0.04 | obesity medical treatment | 380 |
| 0.1 | 10 |
| 0.066 | 15 |
| 0.033 | 30 |
| 0.04 | 50 |
| 0.215 | 0.6 | 5 | 0.22 | space station moon | 384 |
| 0.5 | 10 |
| 0.333 | 15 |
| 0.233 | 30 |
| 0.22 | 50 |
| 0.176 | 0 | 5 | 0.3 | hybrid fuel cars | 385 |
| 0 | 10 |
| 0 | 15 |
| 0.2 | 30 |
| 0.3 | 50 |
| 0.246 | 0.2 | 5 | 0.36 | radioactive waste | 387 |
| 0.2 | 10 |
| 0.2 | 15 |
| 0.3 | 30 |
| 0.36 | 50 |
| 0.42 | 0.6 | 5 | 0.42 | organic soil enhancement | 388 |
| 0.6 | 10 |
| 0.66 | 15 |
| 0.6 | 30 |
| 0.42 | 50 |
| 0.123 | 0.4 | 5 | 0.3 | orphan drugs | 390 |
| 0.2 | 10 |
| 0.2 | 15 |
| 0.366 | 30 |
| 0.3 | 50 |

* + MAP למנוע – 0.073
* **עבור ריצה ללא stemming**:
  + Recall כולל – 0.138
  + Precision כולל – 0.233

|  |  |
| --- | --- |
| **Precision@N** | |
| 0.173 | 5 |
| 0.213 | 10 |
| 0.24 | 15 |
| 0.244 | 30 |
| 0.114 | 100 |

* + למנוע כולו:

* + עבור כל שאילתה:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Recall** | **Precision@N** | | **Precision** | **Qurey** | **Query ID** |
| 0.375 | 0 | 5 | 0.36 | Falkland petroleum exploration | 351 |
| 0.1 | 10 |
| 0.2 | 15 |
| 0.3 | 30 |
| 0.36 | 50 |
| 0.016 | 0.2 | 5 | 0.08 | British Chunnel impact | 352 |
| 0.2 | 10 |
| 0.133 | 15 |
| 0.133 | 30 |
| 0.08 | 50 |
| 0.431 | 0.2 | 5 | 0.44 | blood-alcohol fatalities | 358 |
| 0.3 | 10 |
| 0.466 | 15 |
| 0.4 | 30 |
| 0.44 | 50 |
| 0.321 | 0.2 | 5 | 0.18 | mutual fund predictors | 359 |
| 0.2 | 10 |
| 0.2 | 15 |
| 0.266 | 30 |
| 0.18 | 50 |
| 0.153 | 0 | 5 | 0.12 | human smuggling | 362 |
| 0 | 10 |
| 0.066 | 15 |
| 0.133 | 30 |
| 0.12 | 50 |
| 0.059 | 0 | 5 | 0.22 | piracy | 367 |
| 0.3 | 10 |
| 0.266 | 15 |
| 0.233 | 30 |
| 0.22 | 50 |
| 0.187 | 0.2 | 5 | 0.06 | encryption equipment export | 373 |
| 0.1 | 10 |
| 0.2 | 15 |
| 0.1 | 30 |
| 0.06 | 50 |
| 0.107 | 1 | 5 | 0.44 | encryption equipment export | 374 |
| 0.7 | 10 |
| 0.533 | 15 |
| 0.5 | 30 |
| 0.44 | 50 |
| 0.25 | 0 | 5 | 0.18 | cigar smoking | 377 |
| 0.1 | 10 |
| 0.066 | 15 |
| 0.2 | 30 |
| 0.18 | 50 |
| 0.571 | 0 | 5 | 0.08 | obesity medical treatment | 380 |
| 0.1 | 10 |
| 0.2 | 15 |
| 0.1 | 30 |
| 0.08 | 50 |
| 0.196 | 0 | 5 | 0.2 | space station moon | 384 |
| 0.2 | 10 |
| 0.2 | 15 |
| 0.133 | 30 |
| 0.2 | 50 |
| 0.188 | 0 | 5 | 0.32 | hybrid fuel cars | 385 |
| 0 | 10 |
| 0.066 | 15 |
| 0.233 | 30 |
| 0.32 | 50 |
| 0.178 | 0.2 | 5 | 0.26 | radioactive waste | 387 |
| 0.1 | 10 |
| 0.2 | 15 |
| 0.233 | 30 |
| 0.26 | 50 |
| 0.24 | 0.2 | 5 | 0.24 | organic soil enhancement | 388 |
| 0.4 | 10 |
| 0.266 | 15 |
| 0.333 | 30 |
| 0.24 | 50 |
| 0.106 | 0.4 | 5 | 0.26 | orphan drugs | 390 |
| 0.4 | 10 |
| 0.533 | 15 |
| 0.366 | 30 |
| 0.26 | 50 |

* + MAP למנוע – 0.062