

תיאור העזרים למנוע החיפוש :

postings_body.pickle	postings_title.pickle	postings_anchor.pickle	pageviews.pkl	pageRank.csv
אינדקס body	אינדקס title	אינדקס anchor	כמות צפיות בכל מסמך	דירוג עבור כל מסמך
doc_len.pickle	normal_values	Model	inverted_index_gcp.py	search_frontend.py
שומר עבור כל מסמך : doc_id,title_len(doc)	ערכי הנרמול עבור כל מסמך	עבור שימוש ב- word2vec	בניית האינדקסים	הרצת המנוע

1. בחירת מסמכים**a. כמות מסמכים עבור כל term**

לאחר שאנו מקבלים מילה, נרצה לקבל מהאינדקס הרלוונטי את רשימת המסמכים בהם מופיעה המילה. כמות מסמכים אלו יכולה להיות גדולה, לכן בחנו את האפשרות לקחת חלק יחסי ממנה. לדוגמה, למילה animal יש 200k מסמכים שמכילים אותה וניקח אחוז מסוים מכמות זו. ביצענו ניסוי עבור 80%/75%/50% והגענו למסקנה שאין מנוס מלקחת את כמות המסמכים השלמה בהתאם לאיכות התוצאות. נציין כי במידה והיה לנו זמן נוסף היינו בוחנים את האפשרות לקחת את המסמכים להם ערך tf-idf גבוה מגודל מסוים.

b. שימוש באינדקסים שלנו (body_index, title_index)

עבור פעולת search בחרנו לשלב את השימוש בפונקציונליות של search_body ושל search_title. עבור תוצאות search_body ביצענו דירוג מסמכים, ועבור תוצאות search_title קיבלנו תוצאות עם ערך בינארי.

c. חיתוך כמות המסמכים

לאחר דירוג כלל המסמכים הרלוונטיים שהתקבלו ממילות השאלתה, בחרנו לקחת את 1500 התוצאות עם הערך הגבוה ביותר. קראנו על שיטת cluster pruning בפרק 7 הנלווה למסמך המשימה, בשיטה זאת ישנו שימוש ב- \sqrt{N} מכמות האיברים עליהם נרצה לעבור. ביצענו בדיקת מדדים עבור גודל המסמכים -1000/1500/2000/2500/3000, והגענו למסקנה שבהתאם לזמן ריצה ולדירוג התוצאות נישאר עם 2500.

2. Body Index – דירוג מסמכים

ביצענו חישוב cosineSim לכל מסמך שקיים בו מילה מהשאלתה. את החישוב ביצענו בעזרת הנוסחה המוצגת בתמונה מתוך הרצאה 3 שלמדנו. על מנת להתאים את המשוואה למערכת, התאמנו את משקלי המשתנים להיות על בסיס tf-idf. ביצענו מראש את כל החישובים שאינם תלויים במילות השאלתה, ואנו שומרים אותם במערכת בעת הרצת הinstance. חישובים התלויים במילות השאלתה מתבצעים בזמן ריצה.

3. Title Index – דירוג מסמכים

תוצאות אלו אנו מקבלים באופן בינארי לפי מילות השאלתה, נבצע נרמול לפי אורך השאלתה בכדי להשאיר את ערכי התוצאות בטווח 0-1.

$$\text{CosSim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^l (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^l w_{ij}^2 \cdot \sum_{i=1}^l w_{iq}^2}}$$

$$\text{sim}(q, d_i) = \text{sim}(q, d_i) * (1/|q|) * \text{nf}[i];$$

4. איחוד התוצאות של שני האינדקסים

לאחר שקיבלנו מסמכים משני האינדקסים, נרצה לאחד את תוצאותיהן באופן יחסי. ביצענו השוואה עבור איכות התוצאות שהתקבלו בכל אינדקס ועל פיהן קבענו את האופן היחסי בו נאחד את המסמכים. לדוגמה: ממוצעי איכות התוצאות של האינדקסים הם:

$Avg_title_index = 0.2$, $Avg_Body_Index = 0.4$, לכן נאחד את התוצאות בהתאם

$$\text{למשקלים: } Avg_{title_index} = \frac{0.2}{0.4+0.2}, \text{ Body_Index_weight} = \frac{0.4}{0.4+0.2}$$

לאחר איחוד התוצאות החלטנו לקחת את 200 מסמכים עם הדירוג הגבוה ביותר. (בדיקת 200)

5. Page Views

כעת כשיש לנו 200 תוצאות נרצה להוסיף את מדד הפופולריות של המסמך בכדי להניב את התוצאה שכלל האוכלוסייה ככל הנראה תרצה לקבל. בהרצת המערכת אנו מחזיקים עבור כל מסמך את כמות הצפיות שלו ולכן נרצה להשתמש במידע זה בכדי לשלוף את התוצאות הטובות ביותר. לקחנו את 250 המסמכים שיש לנו ומצאנו לכל אחד את כמות הצפיות שלו.

על מנת לקבוע את אחוז הרלוונטיות של מדד זה ביצענו ניסוי מדדים מופיע בדיאגרמה הנ"ל. תוצאות הניסוי מיקדו אותנו לקחת משקל של 0.25 עבור דירוג כמות הצפיות של המסמך ו-0.75 עבור התוצאות שיש לנו כרגע. לאחר מכן ביצענו איחוד נוסף של התוצאות בהתאם למדד זה ומשם לקחנו את 30 התוצאות הטובות ביותר.

נציין כי ניסינו להוסיף שימוש בתוצאות Page_Rank של המסמכים אך לא הועילו בצורה חד ערכית.

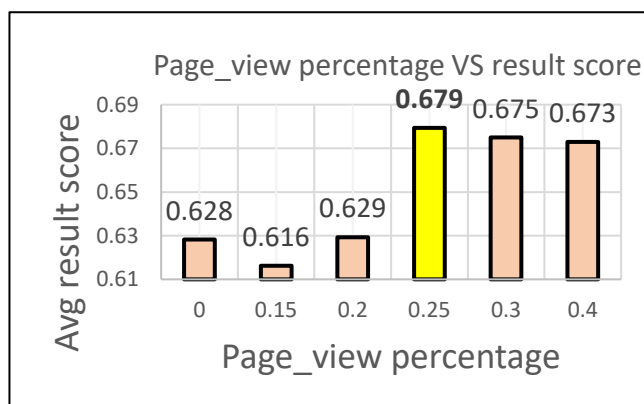
6. חיתוך תוצאות שאינן רלוונטיות

עבור כל שאילתה בחנו את תוצאותיה לעומק. הסתכלנו על שם הכותרת שלה ועל דירוג קרבתה הסופי לשאילתה הנתונה. מצאנו כי קיים פער משמעותי בין דירוג התוצאות הראשונות לבין דירוג התוצאות שאחריהן. למשל, שלושת התוצאות הראשונות נמצאות בטווח 0.6-0.8 ואחריהן טווח הדירוג נמצא ב-0.2-0.4. לכן ביצענו חיתוך לתוצאות אלו והשארנו את התוצאות הרלוונטיות בלבד.

7. ריצת חיפוש נוספת

הגענו לתוצאות טובות לשאילתה, אך על מנת למקסם את החיפוש שלנו בחרנו לעשות ריצת חיפוש נוספת. מכיוון שכבר יש לנו תוצאות עליהן אנו סומכים שהן קרובות לשאילתה, נוכל להשתמש בהן על מנת להניב עוד תוצאות רלוונטיות לחיפוש.

- שיקול זמן ריצה** - מכיוון שאנו נמצאים בנקודה בה זמן הריצה כבר יחסית גדול, נשתמש בחיפוש שזמן הריצה שלו קצר – title_search.
- בניית שאילתה חדשה** – ניקח את הכותרות של התוצאות הכי טובות שלנו ונבנה שאילתה חדשה, נוסיף להם את המילים של השאילתה המקורית ונבצע חיפוש נוסף. החלטנו להוסיף מריצה זאת 5 מסמכים נוספים לתוצאה הסופית של השאילתה.
- משקל מילים מקוריות** - נציין כי בשביל לא לקבל תוצאות שאינן קשורות למילות השאילתה, החלטנו לתת למילות השאילה משקל גדול יותר עבור החיפוש הבינארי. כלומר, מילה שהופיעה בשאילתה המקורית תוסיף לדירוג (+2) ולא (+1) כפי שמבוצע בשיטה הבינארית.



d. **מחשבה נוספת עבור שלב זה** - רצינו לשמור עבור כל מסמך רשימה של doc_id אליהם הוא מצביע, פעולה אשר תתבצע מראש לפני זמן ריצה. בצורה זאת, כאשר נגיע לתשובה סופית נוכל לבדוק את ההצבעות של המסמכים הרלוונטיים ביותר (אפילו רק אחד), לבדוק את קרבתם לשאלתה ובהתאם להוסיף אותן לתוצאות. אנו מאמינים כי בצורה זאת היינו מפיקים מסמכים רלוונטיים נוספים בצורה טובה יותר מהאחת שביצענו.

למשל עבור השאלתה "Argentina football player",
 נקבל עמוד הוויקיפדיה 'argentina national football team',
 משם נוכל להגיע ל- 'lionel messi' עליו העמוד מצביע ואכן רלוונטי לשאלתה.

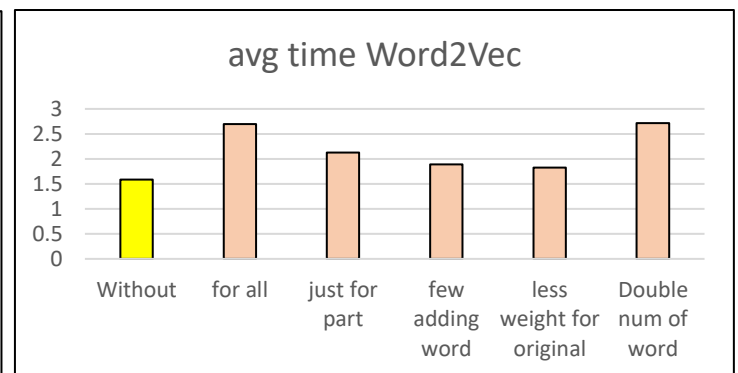
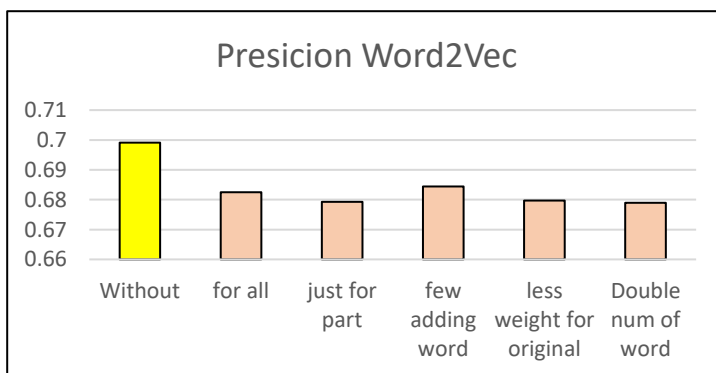
לצערנו, לא הספקנו מפאת חוסר זמן 😞

8. שימוש במודל – Word2Vec

a. במהלך העבודה צפינו בתוצאות שקיבלנו וראינו שתוצאות המנוע טובות אך לא נותנות תשובות בהקשר סמנטי, החלטנו להוסיף את השימוש ב- Word2Vec לטובת חידוד התוצאות. ניסינו להריץ מודל על כל מסמכי הקורפוס ללא הצלחה מפני שגודל הזיכרון להרצת המודל הייתה גדולה מדי. על מנת להתמודד עם הבעיה, תחילה, הרצנו את המודל על גבי 100k מסמכים ובחנו את השימוש במודל. לאחר שראינו כי במקרים ספציפיים השימוש מסייע לנו, הרצנו את המודל על 80% מהקורפוס. לבסוף, בבחינת המנוע על כמות שאלות גדולה, הבחנו כי ההבדל בין תוצאות המנוע עם או בלי השימוש ב- Word2Vec אינו נותן הבדל משמעותי באיכות התוצאות ונותן הבדל רב בזמן החישוב (לרעתו של המודל) ולכן החלטנו לא להשתמש בו.

b. ניסינו להשתמש במודל במספר תרחישים אפשריים:

- i. כמות מילים בשאלתה לאחר tokenization – שימוש במודל רק בהורדת מספר יחסי מסוים של מילים מהשאלתה. למשל: לפחות שלוש מכמות המילים - הורדת 2 מילים מתוך 6. שמנו לב שכאשר מורידים מספר רב של מילים מהשאלתה, ההקשר הסמנטי בין המילים שנשארות משתבש ולכן כדאי להוסיף מילים שימקדו את החיפוש שאנו רוצים לעשות.
- ii. בחינת מילים להוספה –
 1. בחינת כמות מילים – כגודל השאלתה, כפליים וכדומה.
 2. בחירת המילים - לקחנו את המילים הקרובות ביותר עבור כל term בשאלתה, שלפנו כמות מילים הקרובות ביותר לכל ה-term יחד.
 - iii. אורך שאלתה מקורי – למשל מצאנו כי לשאלות קצרות (מילה אחת-שתי מילים), הוספת המילים אינה מיעלת את תוצאות המנוע.
 - iv. התאמת משקלים – הבחנו כי בעת השימוש במודל נצטרך לתת משקל רב יותר למילים המקוריות של השאלתה. זאת מפני שאם לכל המילים יהיו משקלים זהים, חלק מתוצאות המנוע יניבו תוצאות רבות שאינן רלוונטיות למילות השאלתה, אלא למילים שהוספנו.



לדוגמה, בשימוש המודל נביא לכל מילה המתווספת לשאילתה משקל = 1 ולכל מילה שבמקור הייתה בשאילתה משקל = 3, כך בחיפוש המסמכים נוודא כי תוצאות המנוע יתמקדו במילות השאילתה ולא יסטו.

9. הוספת מילים ל-Tokenize

ישנן מילים רבות שמופיעות במסמכים רבים בקורפוס, בשל כך שאילתות שיכילו אותן יכולות להעלות מסמכים רבים שלא מתאימים לשאילתה. בהתאם למה שלמדנו בעבודה 2, מצאנו מילים כאלו והחלטנו להוסיף אותן לתהליך tokenization. למשל, בהרצת שאילתה שניתנה לנו כדוגמה "best marvel movie", המילה best הרסה את התוצאות והוספתה ל-tokenize הראתה הבדל משמעותי בקבלת תוצאות רלוונטיות.

10. שאילתה לא טובה – "How do you breed flowers?"

השאילתה הנ"ל מקבלת אצלו באופן גורף לאורך כל גרסאות המנוע 0 ב-Precision. המרכיב המשמעותי שבגללו המנוע לא מאחזר תוצאות נכונות הוא המשמעות הכפולה שיש למילה breed. מצד אחד, המשמעות היא גזע של חיות בית, מצד שני, המשמעות היא להתרבות, להרבות. לכן בפעולת סינון התשובות הראשונה חוזרות תשובות מעורבות, בין עמודים על סוגי כלבים לבין

עמודים שמדברים על פרחים או אנשים שבשם יש את המילה Flowers. גם בעזרת Word2Vec לא הצלחנו לאחזר דפים הקשורים לגידול פרחים. בסינון השני והאחרון לפי הכותרת חוזרים דפים מעורבים חלק לפי Flowers וחלק לפי breed, אך לא דף המקשר בין שניהם. האפשרות לתקן זאת היא על ידי יצירת מודל המאפשר לתת משמעות סמנטית לשתי מילים ועל הקשר שלהן, בעזרתו שאילתות כמו זו היו מאחזרות דפים רלוונטיים יותר.

11. שאילתה טובה – "Rick and Morty" – precision – 0.97

לאורך הגרסאות העיקריות של המנוע השאילתה הזו מקבלת ציון גבוה, מכיוון שהמנוע מבצע פעמיים חיפוש לפי כותרת. עבור שאילתות כמו זו, באיחוד תוצאות חיפוש לפי טקסט וחיפוש לפי כותרת, השני מקבל משקל רב יותר ומדייק את אחזור הדפים. בעזרת שימוש חוזר של חיפוש לפי כותרת, אנחנו מנצלים את היתרון של שאילתה כזו מכיוון שקיימים דפים רבים אשר מכילים את כלל המילים של השאילתה כבר בכותרת. ניתן לראות בתשובות המצורפות מטה, כי רוב מוחלט של הדפים מכיל בכותרת את כל מילות השאילתה, פרמטר זה עוזר להחזיר בצורה קלה יותר דפים רלוונטיים.

רשימת כותרות ומספרי דפים שהמנוע מאחזר

923811 Brandon Flowers
24016757 Flowers (name)
9507921 The New Breed
801773 Breed standard
39676695 Breed (surname)
15446490 The Breed
895650 Breed club
20897928 Breed (disambiguation)
13780186 Rare Breed
37570236 London Breed
15119070 Brandon Flowers (American football)
27162198 Flamingo (Brandon Flowers album)
27763291 Crossfire (Brandon Flowers song)
29075001 Only the Young (Brandon Flowers song)

רשימת כותרות ומספרי דפים שהמנוע מאחזר

41283158 List of Rick and Morty episodes
41185040 Rick and Morty
67830379 Rick and Morty (franchise)
47762921 List of Rick and Morty characters
49128142 Pilot (Rick and Morty)
65819511 Rick and Morty (season 5)
49029294 Rick and Morty (season 3)
51082764 Rick and Morty: Virtual Rick-ality
67520032 Jerry Smith (Rick and Morty)
49134425 Close Rick-counters of the Rick Kind
41699729 Morty Smith
51759111 Rick Sanchez
57390230 Rick and Morty (season 4)
43794572 Rick and Morty (season 1)
43794574 Rick and Morty (season 2)
64413225 Rick and Morty (soundtrack)

4.47MiB gs://final-bucket-general/help_files/Model.model	1.91MiB gs://final-bucket-general/anchor_index/101_002.bin	513.34KiB gs://final-bucket-general/anchor_index/100_posting_locs.pickle
55.92MiB gs://final-bucket-general/help_files/Model.model.syn1neg.npy	841.84KiB gs://final-bucket-general/anchor_index/101_003.bin	500.49KiB gs://final-bucket-general/anchor_index/101_posting_locs.pickle
55.92MiB gs://final-bucket-general/help_files/Model.model.wv.ve	.	502.69KiB gs://final-bucket-general/anchor_index/102_posting_locs.pickle
ectors.npy	1.91MiB gs://final-bucket-general/anchor_index/97_000.bin	494.66KiB gs://final-bucket-general/anchor_index/103_posting_locs.pickle
127.06MiB gs://final-bucket-general/help_files/dict_norm_list.pkl	1.91MiB gs://final-bucket-general/anchor_index/97_001.bin	504.67KiB gs://final-bucket-general/anchor_index/104_posting_locs.pickle
249.9MiB gs://final-bucket-general/help_files/doc_len.pickle	1.6MiB gs://final-bucket-general/anchor_index/97_002.bin	510.48KiB gs://final-bucket-general/anchor_index/105_posting_locs.pickle
170.16MiB gs://final-bucket-general/help_files/pageRank.csv	1.91MiB gs://final-bucket-general/anchor_index/98_000.bin	.
73.5MiB gs://final-bucket-general/help_files/pageviews.pkl	1.91MiB gs://final-bucket-general/anchor_index/98_001.bin	.
-----anchor-----	1.91MiB gs://final-bucket-general/anchor_index/98_002.bin	.
1.91MiB gs://final-bucket-general/anchor_index/0_000.bin	764.61KiB gs://final-bucket-general/anchor_index/98_003.bin	501.33KiB gs://final-bucket-general/anchor_index/90_posting_locs.pickle
1.91MiB gs://final-bucket-general/anchor_index/0_001.bin	1.91MiB gs://final-bucket-general/anchor_index/99_000.bin	504.41KiB gs://final-bucket-general/anchor_index/91_posting_locs.pickle
1.91MiB gs://final-bucket-general/anchor_index/0_002.bin	1.91MiB gs://final-bucket-general/anchor_index/99_001.bin	503.94KiB gs://final-bucket-general/anchor_index/92_posting_locs.pickle
1.67MiB gs://final-bucket-general/anchor_index/0_003.bin	1.91MiB gs://final-bucket-general/anchor_index/99_002.bin	497.14KiB gs://final-bucket-general/anchor_index/93_posting_locs.pickle
1.91MiB gs://final-bucket-general/anchor_index/100_000.bin	822.98KiB gs://final-bucket-general/anchor_index/99_003.bin	501.6KiB gs://final-bucket-general/anchor_index/94_posting_locs.pickle
1.91MiB gs://final-bucket-general/anchor_index/100_001.bin	1.91MiB gs://final-bucket-general/anchor_index/9_000.bin	493.27KiB gs://final-bucket-general/anchor_index/95_posting_locs.pickle
1.91MiB gs://final-bucket-general/anchor_index/100_002.bin	1.91MiB gs://final-bucket-general/anchor_index/9_001.bin	
1.17MiB gs://final-bucket-general/anchor_index/100_004.bin	1.91MiB gs://final-bucket-general/anchor_index/9_002.bin	
1.91MiB gs://final-bucket-general/anchor_index/101_000.bin	200.4KiB gs://final-bucket-general/anchor_index/9_003.bin	
1.91MiB gs://final-bucket-general/anchor_index/101_001.bin	500KiB gs://final-bucket-general/anchor_index/0_posting_locs.pickle	

499.83KiB gs://final-bucket-general/anchor_index/96_posting_locs.pickle
503.23KiB gs://final-bucket-general/anchor_index/97_posting_locs.pickle
495.3KiB gs://final-bucket-general/anchor_index/98_posting_locs.pickle
504.2KiB gs://final-bucket-general/anchor_index/99_posting_locs.pickle
494.64KiB gs://final-bucket-general/anchor_index/9_posting_locs.pickle
94.99MiB gs://final-bucket-general/anchor_index/anchor.pkl

-----title index-----

862.98KiB gs://final-bucket-general/title_index/0_000.bin
865.72KiB gs://final-bucket-general/title_index/100_000.bin
638.64KiB gs://final-bucket-general/title_index/101_000.bin
570.03KiB gs://final-bucket-general/title_index/102_000.bin
711.14KiB gs://final-bucket-general/title_index/103_000.bin
1.48MiB gs://final-bucket-general/title_index/104_000.bin
1.04MiB gs://final-bucket-general/title_index/105_000.bin
539.9KiB gs://final-bucket-general/title_index/106_000.bin
1.06MiB gs://final-bucket-general/title_index/107_000.bin
538.8KiB gs://final-bucket-general/title_index/108_000.bin
593.35KiB gs://final-bucket-general/title_index/109_000.bin

841.94KiB gs://final-bucket-general/title_index/10_000.bin
1MiB gs://final-bucket-general/title_index/110_000.bin
.
.
.
762.52KiB gs://final-bucket-general/title_index/90_000.bin
693.05KiB gs://final-bucket-general/title_index/91_000.bin
630.26KiB gs://final-bucket-general/title_index/92_000.bin
697.75KiB gs://final-bucket-general/title_index/93_000.bin
736.15KiB gs://final-bucket-general/title_index/94_000.bin
782.96KiB gs://final-bucket-general/title_index/95_000.bin
944.4KiB gs://final-bucket-general/title_index/96_000.bin
557.64KiB gs://final-bucket-general/title_index/97_000.bin
837.83KiB gs://final-bucket-general/title_index/98_000.bin
649.58KiB gs://final-bucket-general/title_index/99_000.bin
641.88KiB gs://final-bucket-general/title_index/9_000.bin
324.67KiB gs://final-bucket-general/title_index/0_posting_locs.pickle
329.91KiB gs://final-bucket-general/title_index/100_posting_locs.pickle
325.9KiB gs://final-bucket-general/title_index/101_posting_locs.pickle
324.53KiB gs://final-bucket-general/title_index/102_posting_locs.pickle
320.23KiB gs://final-bucket-general/title_index/103_posting_locs.pickle

327.38KiB gs://final-bucket-general/title_index/104_posting_locs.pickle
328.89KiB gs://final-bucket-general/title_index/105_posting_locs.pickle
.
.
.
327KiB gs://final-bucket-general/title_index/90_posting_locs.pickle
326.92KiB gs://final-bucket-general/title_index/91_posting_locs.pickle
330.34KiB gs://final-bucket-general/title_index/92_posting_locs.pickle
322.4KiB gs://final-bucket-general/title_index/93_posting_locs.pickle
329.19KiB gs://final-bucket-general/title_index/94_posting_locs.pickle
322.77KiB gs://final-bucket-general/title_index/95_posting_locs.pickle
326.72KiB gs://final-bucket-general/title_index/96_posting_locs.pickle
325.71KiB gs://final-bucket-general/title_index/97_posting_locs.pickle
326.49KiB gs://final-bucket-general/title_index/98_posting_locs.pickle
328.4KiB gs://final-bucket-general/title_index/99_posting_locs.pickle
326.21KiB gs://final-bucket-general/title_index/9_posting_locs.pickle
67.6MiB gs://final-bucket-general/title_index/index_title.pkl

-----body index-----

101.39KiB gs://final-bucket-
general/body_index/0_posting_locs.pi
ckle
101.28KiB gs://final-bucket-
general/body_index/100_posting_locs
.pickle
103.67KiB gs://final-bucket-
general/body_index/101_posting_locs
.pickle
100.34KiB gs://final-bucket-
general/body_index/102_posting_locs
.pickle
99.84KiB gs://final-bucket-
general/body_index/103_posting_locs
.pickle
101.01KiB gs://final-bucket-
general/body_index/104_posting_locs
.pickle
100.84KiB gs://final-bucket-
general/body_index/105_posting_locs
.pickle
99.31KiB gs://final-bucket-
general/body_index/106_posting_locs
.pickle
.
.
.
96.67KiB gs://final-bucket-
general/body_index/90_posting_locs.
pickle
99.83KiB gs://final-bucket-
general/body_index/91_posting_locs.
pickle
103.27KiB gs://final-bucket-
general/body_index/92_posting_locs.
pickle
103.24KiB gs://final-bucket-
general/body_index/93_posting_locs.
pickle

101.54KiB gs://final-bucket-
general/body_index/94_posting_locs.
pickle
99.12KiB gs://final-bucket-
general/body_index/95_posting_locs.
pickle
99.66KiB gs://final-bucket-
general/body_index/96_posting_locs.
pickle
101.83KiB gs://final-bucket-
general/body_index/97_posting_locs.
pickle
99.44KiB gs://final-bucket-
general/body_index/98_posting_locs.
pickle
101.47KiB gs://final-bucket-
general/body_index/99_posting_locs.
pickle
100.18KiB gs://final-bucket-
general/body_index/9_posting_locs.pi
ckle
18.45MiB gs://final-bucket-
general/body_index/index_body.pkl
1.91MiB gs://final-bucket-
general/body_index/0_000.bin
1.91MiB gs://final-bucket-
general/body_index/0_001.bin
1.91MiB gs://final-bucket-
general/body_index/0_002.bin
1.91MiB gs://final-bucket-
general/body_index/0_003.bin
1.91MiB gs://final-bucket-
general/body_index/0_004.bin
1.91MiB gs://final-bucket-
general/body_index/0_005.bin
1.91MiB gs://final-bucket-
general/body_index/0_006.bin
1.91MiB gs://final-bucket-
general/body_index/0_007.bin
1.91MiB gs://final-bucket-
general/body_index/0_008.bin
1.91MiB gs://final-bucket-
general/body_index/0_009.bin

1.91MiB gs://final-bucket-
general/body_index/0_010.bin
1.91MiB gs://final-bucket-
general/body_index/0_011.bin
1.91MiB gs://final-bucket-
general/body_index/0_012.bin
1.91MiB gs://final-bucket-
general/body_index/0_013.bin
1.91MiB gs://final-bucket-
general/body_index/0_014.bin
1.91MiB gs://final-bucket-
general/body_index/0_015.bin
1.91MiB gs://final-bucket-
general/body_index/0_016.bin
1.91MiB gs://final-bucket-
general/body_index/0_017.bin
1.91MiB gs://final-bucket-
general/body_index/0_018.bin
1.91MiB gs://final-bucket-
general/body_index/0_019.bin
1.91MiB gs://final-bucket-
general/body_index/0_020.bin
1.91MiB gs://final-bucket-
general/body_index/0_021.bin
1.91MiB gs://final-bucket-
general/body_index/0_022.bin
1.91MiB gs://final-bucket-
general/body_index/0_023.bin
1.91MiB gs://final-bucket-
general/body_index/0_024.bin
1.91MiB gs://final-bucket-
general/body_index/0_025.bin
1.91MiB gs://final-bucket-
general/body_index/0_026.bin
1.91MiB gs://final-bucket-
general/body_index/0_027.bin
57.93KiB gs://final-bucket-
general/body_index/0_028.bin
1.91MiB gs://final-bucket-
general/body_index/100_000.bin
.
.
.

