דו"ח חלק א'

1. עיצוב התוכנה:
2. תחילה, אנו מבצעים קריאה של המידע אותו אנו רוצים לעבד. בצורה איטרטיבית אנו עוברים כל פעם על קובץ המכיל מידע על טוויטים.

לאחר מכן מפרסרים את המידע על הטוויט ומסווגים אותו לשדות שונים כאשר ישנם שדות לURL, לTWEETID לFULL TEXT ועוד מידע נחוץ שיכול לעזור לנו בהמשך. במהלך הפירסור על הטוויט, אנו מסווגים כל מילה לפי הדרישות ובנוסף מורידים ממנו מידע אשר בעניינו נחשב כלא נחוץ (לדוגמה סמיילים, טקסט בערבית וסינית וכו').

לאחר שאנו סיימנו לפרסר את כל הטוויט, אנו מעבירים את הDOCUMENT שיצרנו לפי נושאים לשלב האינדוקס, בו אנו מחלקים את המידע (מילים שמצאנו לנכון והעברנו מהפרסר לDOCUMENT) בצורה שתהיה לנו נוחה לאחזר את המידע כאשר נדרש לעשות זאת. אנו שומרים עבור כל TERM בכל DOCUMENT בקובץ חיצוני לפי האות הראשונה (כאשר כל 25 אלף טוויטים, כך מצאנו לנכון בגלל אילוצי זמן ומקום לאחר), את כמות הפעמים שהיא מופיעה בקורפוס, ובאילו טוויטים היא הופיע.

לאחר שביצענו איטרטיבי את פעולות הקריאה-פרסור-אידוקס את כל הקורפוס, אנו מעבדים את המידע, ועושים איחוד למסמכים על מנת להקל על עצמינו בשלב החיפוש והדירוג, כך שהמידע יהיה נגיש בצורה נוחה יותר.

לאחר מכן, אנו בונים את המודל LDA שלנו, בעזרת כל הנתונים שקיימים, כך שאנו מעלים לפני כן, את כל המידע עבור המסמכים שנמצאים בזיכרון החיצוני, וכך לא תופסים מקום על הRAM.

כעת, אחרי שהמודל עלה, אנו מבצעים בדיקה ושמירה במבנה נתונים עבור כל מסמך את הנושא שהכי דומה לו ואת ההסתברות שלו, כך שבמידה ושאילתה מתאימה לנושא מסוים אזי בשלב הSEARCHER אנו מחזירים את כל המסמכים הקשורים לאותו נושא.

לבסוף, אנו מבצעים דירוג לפי COS SIMILARITY שנלמד בהרצאה ובתרגול וכך אנו מדרגים את הטוויטים.

1. למגבלת הזיכרון היינו צריכים לייעל כל אחד מהבאים:

1. תחילה בשלב הREADER ביצענו קריאה ושמירה של כל המידע (טוויטים) של כל הקורפוס של קבצי הפרקט במערך (LIST) אחד אשר מכיל את כל ה10 מיליון טוויטים, דבר אשר גרם בהמשך העבודה ככל שהתקדמנו לבעיית זיכרון ואיטיות שנבעה מכך, לכן החלטנו לעבור בלולאה ולעשות באופן איטרטיבי כך שקודם מכניסים למערך אחד את כל הטוויטים בקובץ פרקט אחד, עושים פרסור על כל המאגר ואינדקוס ואז מוחקים את המערך ומכניסים למערך קובץ טוויטים חדש מקובץ פרקט אחר.

2. בשלב הפרסר, אנו התחלנו לערוך ניסוי מה השיטה הטובה ביותר לרוקן את הטוויט ממידע לא רלוונטי, כגון, סמיילים, צורות למיניהם, טקסט בסינית וכו', וכל פעם מצאנו דרך טובה יותר באמצעות ביטוי רגקס מתאים כך שהמידע יכיל רק את התווים שאנו דורשים שהוא יכיל, דבר אשר מסייע בנוסף לשלב האינדוקס.

3. בשלב האינדוקס שבו החלק המרכזי של הזיכרון, אנו ערכנו כל פעם ניסוי במטרה להבין כיצד המחשב מתמודד בדרך הטובה ביותר עם המידע, כך שחלוקת המידע בין הRAM לבין הזיכרון החיצוני תהיה אופטימלית, הן עבור בעיית זיכרון והן עבור בעיית זמן. כיוון שאנו שומרים מידע רב, נאלצנו לעבוד ביסודיות, כך שכל פעם שמגיע לי TERM מסוים אני מכניס אותו למילון לפי האות הראשונה שלו, וכך אנחנו יודעים כאשר נצברים מאות מסויימת 500 אלף TERMS, אנו מעלים את המידע הזה של אותו אות לזיכרון הראשי ומוחקים את המידע הקיים מהזיכרון RAM, כך אנו דואגים שהכמות זיכרון של הRAM לא תתמלא.

4. בשלב הMERGE, זהו צעד אמנם יקר מבחינת זמן, אך בהמשך יעזור לנו להחזיר בצורה מהירה יותר את הדירוג של הטוויטים. אנו נאלצים לבצע MERGE של כל הקבצים לפי אותיות, כך שבסוף מתקבל קובץ פוסטינג אחד לכל אות.

5. המודל שקיבלנו היה LSI/LDA, כאשר בחרנו במודל LDA, אנו נאלצנו לשמור את המילון של הקורפוס ואת מאגר הTEXT TOKENS של כל DOCUMENT. התמודדנו עם הבעיה בכך שכל 100 אלף (לאחר ניסויים גילינו שזה המספר הטוב ביותר לשמור גודל של מידע כזה בזיכרון הראשי) אנו מעבירים לקובץ פוסטינג את כל המידע הזה ואז מפנים מחדש את הזיכרון במבנה נתונים הזה. לאחר אגירת כל המידע בקבצי פוסטינג אנו מבצעים MERGE שנועד לאחד את כולם לקובץ אחד אשר עוזר לנו להכניס את כל המידע לקורפוס ולמילון של המודל, כך שהוא לא נשמר על הזיכרון הראשי. לאחר יצירת המודל, אנו מתחילים ישר בביצוע ההסתברויות ושמירת עבור כל מסמך את ההסתברות של הנושא הקרוב ביותר, ולאחר מכן שומרים ישר את המודל בקובץ חיצוני על מנת שלא יישאר על הRAM.

1. במחלקת הINDEXER, אנו מבצעים שמירת עבור כל אות, סימנים מיוחדים כאחד, והספרה הראשונה של מספר, שמירת קובץ פוסטינג כאשר כל קובץ מכיל 500 אלף טוויטים. בסוף תהליך האינדוקס אנו מגיעים למצב של בממוצע (משתנה בין אותיות שכיחות לזניח) 20 קבצי פוסטים ומבצעים להם MERGE לקובץ אחד לכל אות. הקובץ במבנה נתונים של מילון, כך שהמפתח זה הTERM והערך זה הTWEETID שבו הוא מופיע. כמו כן, אנו שומרים גם קובץ מילים נפוצות

בנוסף אנו שומרים עבור המודל שקיבלנו (LDA) ועבור הרנקר (שם ביצענו COS SIMILARITY) את כל הטוקנס מכל מסמך כולל הTWEET ID, אנו שומרים כל 100 אלף DOCUMENTS כיוון שהם שוקלים משמעותית יותר משאר מבני הנתונים.

לאחר תהליך הMERGE אנו מקבלים קובץ מספרים אחד, קובץ אחד לכל אות, קובץ DOCUMENTS, וקובץ מילים נפוצות.

1. לאחר ניסויים של ערכים בהם בחרנו כל פעם להעלות את הנתונים לקובץ פוסטינג הגענו להחלטה משותפת בגלל בעיות מקום (אשר משפיע ישירות על הזמן) שכל קובץ לאות אנו מעלים כאשר נאגרים בו 500 טוויטים שמוזכרים בו. עבור המסמכים החלטנו כל 100 אלף כיוון שכל המילים עבור DOCUMENT מסוים גוזלים יותר מקום. כך הגענו למצב שאנו שולטים בצורה מבוקרת על הזמן ועל המקום ככל שניתן.

להוסיף תיעוד!!! (של קבצי הפוסטינג)

1. אנו בחרנו להוסיף לכל TERM וTWEETID המציין באיזה טוויט הוא הופיע בקורפוס את ה-tf ואת ה-idf שלו במסמך הנתון, דבר אשר מאוד מכל עלינו בתהליך הדירוג, כיוון שבשביל לחשב את הCOS SIMILARITY, אנו זקוקים לנתונים אלו.

בנוסף, עבור המודל שלנו שמרנו גם את כל הTERMS שקיבלנו עבור המסמך בתהליך הפרסר, בשביל ליצור את המודל.

1. שני החוקים שהוספנו בשלב בפרסר הם:
2. ציטוטים: תהליך המימוש מתבצע כך שבכל פעם שאנו מגיעים בFULL TEXT לביטוי מהצורה גרשיים (לא משנה אם גרש או שניים) + סטרינג בפנים (לא משנה אם מדובר בסטרינג אחד או כמה) ואז שוב פעם גרשיים, אנו שומרים את כל הביטוי (ציטוט) כאחד, בנוסף לשמירת כל TERM בפני עצמו. הבחנו שפעמים לא מעטות ישנם ציטוטים, ולכן החלטנו שלא ניתן להתעלם מהם, ושהם במידה מסויימת בהחלט יכולים לעזור לנו לגבש מידע יותר מדויק עבור הטוויט.

להוסיף דוגמה, איך ב2 טוויטים שונים מגיעים התוצאות

1. הורדנו מידע מיותר: תהליך המימוש מתבצע כך שביצענו ביטוי רגקס עבור על מילה בטוויט שבו החלטנו בדיוק מה התווים אותם אנו מעוניינים שיכלול (כמו מספרים תווים באותיות גדולות וקטנות, נקודה, סלאש, גרשיים, סימן קריאה ושאלה וסימן דולר), כך שאם ישנם תווים/סימנים/שפות זרות וכו' הוא לא יבין בהם (לדוגמה מילה בערביץ, כל מיני טעויות כתיב בהוספת סימנים בטעות באמצע מילה, סמיילים, טקסט בסינית וערבית, וסימנים אשר נמצאים בשפות זרות שאינם נמצאים במקלדת שלנו ונדרש להביאם ממקלדת זרה). במצב זה ניקינו המון "מידע מלוכלך" אשר אינו תורם לנו כלל אם היינו משאירים אותו במאגר הTERMS עבור הטוויט, ומנפח את המילון במקום. תהליך זה שיפר לנו גם את זמני הריצה כיוון שניקינו מידע שלטעמינו לא היה קריטי בעת ביצוע שלב האיחזור.

לתקן את סעיף 2 כי אולי לא חכם לרשום שלנקות מידע לא שימושי זה כולל לנקות ערבי וסינית למרות שעשינו את זה כי יש מצב שאם מישהו ירשום נגיד שאילתה בסינית אז זה יעזור לו להגיע לטוויטים המיועדים, צריך לחשוב איך לנסח טוב.

להוסיף דוגמה, איך ב2 טוויטים שונים מגיעים התוצאות

1. כללים וחוקים שהוספנו בשלב הפרסר הם:
2. הוספנו אופציה למקרים של מילה מחוברת לסימן קריאה, סימן שאלה, נקודה או פסיק ואז מיד עוד מילה, וזה יפרש אותן כ2 מילים שונות.

כללים וחוקים שהוספנו בשלב האינדקסר הם: