עבודה 3- מערכות מבוזרות

תחילה נשים לב שהמערכת שלנו מחולקת לשני חלקים (קומפוננטות)

**1.החלק הראשון הוא חישוב ה-MI (Mutual information)**

בחשיוב ה-MI אנחנו מעוניינים לחשב את הנוסחא הבאה :



חילקנו את העבודה לכמה שלבים שאותם נפרט בהמשך:

לחלק זה של המערכת קיבלנו כקלט: 10\1 קבצי biarcs .

1. פילטור הקלט: פילטרנו את הקלט. כך שpath הוחשב אצלנו כתקין אם יש שני שמות עצם בצדדים, שהם מסוג נושא או מושג . וביניהם יש פועל שמחבר ביניהם. JobFiltered

Mapper:

קלט: קבצי biarcs -- <LongWritable,Text> (160,050,720 **k-v**)

פלט: המסלולים שלנו כלומר: <p,slot,w>. (9,280,063 **k-v**)

Reducer:

קלט: הפלט מהמאפר <Text, Text> (9,280,063 **k-v**)

פלט: N – שמכיל את הערך  (7,191,822 **k-v)**

ואת המסלולים שלנו כלומר: <p,slot,w>.

ב

1. בשלב זה נחשב כל אחד מהחלקים במונה והמכנה בנפרד.
2. נחשב את <p,slot,\*>
3. נחשב את <\*,slot,W>
4. נחשב את <p,slot,w>

Mapper:

קלט: נקבל את האינפוט מהjob הקודם (9,280,063 **k-v**)

פלט: נוציא קלט ופלט לפי ספירה של כל אחד מהם שרלוונטי לנו כלומר הזוגות (9,280,063 **k-v**)   
מפתחות:

עבור ה-1 : <Path,numberOfocc>

עבור ה-2: <slot+w>,numberofocc>>

עבור ה-3: <p,slot,w>,numberofocc>>

Reducer:

קלט: הפלט של המאפר (5,436,744 **k-v**)

פלט: סכימה של הזוגות לפי מפתחות זהים. (1,980,511 **k-v**)

סהכ קיבלנו בשלב הזה כל אחד "מהאיברים" בנוסחא.

1. בשלב זה נממש את החלק של הנוסחא. נבצע את החישוב על ידי חישוב הנוסחא בשלבים.

קודם כל נחשב את החלק של:

נקבל כקלט את ה-N שמייצג את <\*,slot,\*> <p,slot,w>,<p,slot,\*>

Mapper:

קלט: נקבל את המפתחות מסוג <Text,Text>

כאשר המפתחות יקבלו את הערכים הבאים: <p,slot,w>,<p,slot,\*> (4,125,779 **k-v**)

פלט: נסדר את הפלט כך ששני הקלטים הללו יהיו תחת אותו מפתח כלומר <p,><slot,w>

<p,slot> (4,125,779 **k-v**)

Reducer:

קלט: נקבל את הפלטים של המאפר, ואת ה-N מהסביבה. (4,125,779 **k-v**)

פלט: נקבל חישוב חלקית של הנוסחא Mi (4,125,779 **k-v**)

והמפתח והערך יהיו:

<slot,w><p,result>

את החלק השני של הנוסחא נחשב באופן דומה :

Mapper:

קלט: מקבל את הפלט מהגוב הקודם + פלט של המפתח <\*,slot,w> (4,125,779 **k-v**)

פלט: מסדר את שני הקלטים ככה שיהיו בעלי מפתח משותף (4,125,779 **k-v**)

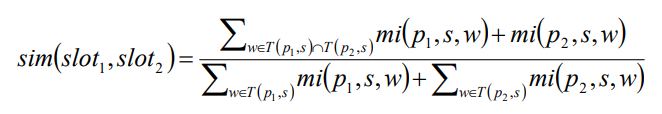
Reduce:

קלט: מקבל את הפלט מהמאפר (4,125,779 **k-v**)

פלט: מחשב את הנוסחא. (4,125,779 **k-v**)

2.**החלק השני הוא חישוב הוא חישוב ה-SIM ) הדמיון בין כל שני SLOT -ים) ולבסוף נחשב את הדמיון בין שני מסלולים S(p1,p2).**

על מנת לחשב את הדמיון בין שני המסלולים אנחנו נצטרך לחשב את הנוסחה הבאה:



הנוסחה הנ"ל מחשבת את הדמיון בין שני slot-ים.

ולבסוף על מנת למצא את הדמיון בין שני מסלולים נשתמש בנוסחה הבאה:



חילקנו את החלק זה לכמה שלב:  
  
א. פילטור הקלט- (4,125,779 **k-v**) -   
אנחנו נפלטר את החישוב מהשלב הקודם- כלומר הנוסחה הראשונה (MI) ע"פ test set המוכנס בתור קלט. בtest set יש את המסלולם שאנחנו רוצים לצמצם את החישוב להם.  
Mapper:

קלט: הפלט מהשלב הקודם <Mi calc> <p ,“/x” or “/y”,w > , Test set הנתון לנו בהתחלה לכל :Mapper

קלט: [<Mi calc>] <p ,“/x” or “/y”,w >  
פלט: <Mi calc> <p ,“/x” or “/y”,w > מסוננים לפי הפרמטרים

Reduce:

קלט: [<Mi calc>] <p ,“/x” or “/y”,w > מסוננים  
פלט: <Mi calc> <p ,“/x” or “/y”,w >

ב. חישוב מקדים בשביל חישוב הנוסחה sim(slot x, slot y)

אנחנו נרצה לעשות זאת ע"י חישוב החישוב המונה והמכנה בכל תת שלב

ולפני כן חישוב נחשב לכל מילה את כל המסלולים שיכולים להשלים אותה ובנוסך נכניס את הערך מהשלב הקודם.

:Mapper

קלט: <Mi calc> <p ,“/x” or “/y”,w > מסוננים  
פלט: אם זה עם התו X:   
<w$>,<$P>   
 אם זה עם התו y:  
<w%>,<%P>

Reduce:

קלט: <w><p[p0,p1,p2..]>  
פלט: <w>,<path + path count>בעצם אנחנו יוצרים מבנה נתונים ששומר את כל המילים שמשלימות את כל המסלולים האפשריים

חישוב המונה- עבור כל מילה שמשלימה נחשב את הערך של המשקל שלו לכל מסלול משלו

:Mapper

קלט: <p[]> <line id>  
פלט: <w1 + “\” + w2><sum sim>

Reduce:

קלט: <[sum0, sum1, sum2..]> <w1 + “\” + w2>  
פלט: <w1 + “\” + w2>< total sum sim>

חישוב המכנה- בעזרת המבנה נתונים שיצרנו בתוכנית map-reduce הראשון בשלב זה

:Mapper

קלט: <p[]> <line id>  
פלט: <path + “%” or “$”><sum sim>

Reduce:

קלט: <[sum0, sum1, sum2..]> <path + “%” or “$”>  
פלט: <path + “%” or “$”>< total sum sim>

ג. חישוב הנוסחה הכללית sim:

יש שני Mapper בגלל שאנחנו מקבלים גם את המונה וגם את המכנה של הנוסחה

:1Mapper

קלט: <path + “%” or “$”>< total sum sim>  
פלט: <path ><Den>

:Mapper

קלט: <w1 + “\” + w2>< total sum sim>  
פלט: <path><Num>

Reduce:

קלט: <[ num\den]> <path >  
פלט: <key>< num\den>

ד.זהו השלב האחרון שבו אנו מחשבים את הנוסחה שניה וממינים אותה

בהתחלה אנחנו מחשבים את הנוסחה:



:Mapper

קלט: <key>< num\den>  
פלט:

: Reduce

קלט: <key>< num\den>  
פלט: ממוין (3,979,842 **k-v**)

והשלב האחרון מתבצע מיון בסדר יורד- map-reduce נוסף

גרף היחס בין מספר ההצלחות (recall) לעומת מספר הצלחות חלקי מספר ניסיונות (precision) הוא:

בקלט הגדול כאשר הכנסנו את כל הקלט התוצאה הייתה 0.65 ב 0.01 לעומת 0.97 ב 0.001

בקלט הקטן כאשר הכנסנו עשירית מהקלט התוצאה הייתה 0.75 ב 0.01 לעומת 0.90 ב 0.001

השגיאות:

עבור הקובץ הקטן:

True negative:

X constrict Y X develop Y   
X produce Y X treat with Y

False negative:

X use in Y Y give to X

True positive:

X die from Y X die of Y  
X require Y X control with Y

False Positive:

עבור הקלט הגדול:

עבור הקובץ הקטן:

True negative:

False negative:

X use in Y Y give to X

True positive:

X make from Y X make of Y

X suggest Y X be in Y

False Positive: