אחזור מידע

תרגום חופשי של הספר Information Retrieval

של C. J. van RIJSBERGEN

פרק 5 - אסטרטגיות חיפוש

תרגום : פנינה קרמר ורחל ריינפלד
עריכה : עפר דרורי

הקדמה

עד עתה לא עסקנו בפרוצדורה המשמשת לזיהוי אינפורמציה נדרשת. כשמדובר בשליפת מסמכים, המידע הוא קבוצת המסמכים הרלוונטיים לשאילתה.

בפרק 4 הוזכרה יעילות החיפוש והתאמת מבנה הקובץ לחיפוש יעיל. סוג החיפוש שאנו נעסוק בו הוא לא הסוג הרגיל בו התוצאה היא שהמבוקש נמצא/לא נמצא. סוג זה מתאים כשיוצרים מילונים עבור עיבוד טקסטים. אולם, אנו מעוניינים באסטרטגיות חיפוש שתוצאתם פחות או יותר תתאים לדרישות השאילתה.

כל אסטרטגיות החיפוש מבוססות על השוואה בין שאילתה למסמכים השמורים. לעיתים, השוואה זו מושגת באופן בלתי ישיר כאשר השאילתה משוואת לאיגודים/אשכולות (או ביתר דיוק לפרופילים המייצגים את האיגודים). ניתן להבין את האבחנות בין סוגים שונים של אסטרטגיות חיפוש ע״י הסתכלות בשפת השאילתה שהיא השפה המבטאת את המידע הדרוש.

מאפייני שפת השאילתה מכתיבים את מאפייני אסטרטגיית החיפוש. לדוגמה, שפת שאילתה שמאפשרת משפטי חיפוש במונחים לוגיים של אוסף מילות מפתח, מחייבת חיפוש בוליאני. חיפוש זה מניב תוצאות ע״י השוואות לוגיות (לעומת נומריות) בין השאילתה למסמכים. לא נבחן שפות שאילתה אלא נבחן את ההבדלים בין שפות בעזרת התמקדות במכניזם החיפוש.

חיפוש בוליאני

באסטרטגיות החיפוש הבוליאני, המסמכים הנשלפים הם אלה שתוצאתם מתאימה בדיוק לשאילתה. ניסוח זה הגיוני רק במקרים שהשאילתות באות לידי ביטוי במונחי אינדקס (או במילות מפתח) ומקושרים בעזרת קישוריות לוגית של AND, OR, ו - NOT. למשל, אם השאילתה

Q= (K1 AND K2) OR (K3 AND (NOT K4)) אז החיפוש הבוליאני ישלוף את כל המסמכים שסימן ההיכר שלהם כולל את K1 ו - K2 ואת המסמכים שסימן ההיכר שלהם כולל את K3 ולא את K4.

ישנם מערכות, הפועלות עפ״י עקרון החיפוש הבוליאני, המאפשרות למשתמש להרחיב לולהצר את החיפוש ע״י מתן גישה למילון מובנה. עבור כל מילת מפתח, מילון זה מאחסן מילים הקשורות למילה זו שיכולות להיות כלליות או מדויקות יותר.

למשל במבנה העץ בדוגמא 5.1, מילת המפתח K מוכלת במילת מפתח כללית יותר K אך יכולה גם -כן להתחלק לארבע מילות מפתח מדויקות יותר (K1^2, K2^2, K3^2, K4^2). לכן, כאשר המערכת אינטראקטיבית החיפוש יכול לעבור ניסוח מחדש בעזרת שימוש במונחים מקושרים למילת המפתח.

Some systems, which operate by means of Boolean searches, allow the user to narrow or broaden the search by giving the user access to a structured dictionary which, for any given keyword, stores related keywords which may be more general or more precise. For example, in the tree structure in *Figure 5.1*, the keyword K_1^1 is contained in the more general keyword K_1^0, but it can also be split up into 4 more precise keywords K_1^2, K_2^2, K_3^2, and K_4^2. Therefore, if one has an interactive system the search can easily be reformulated using some of these related terms.

אחז

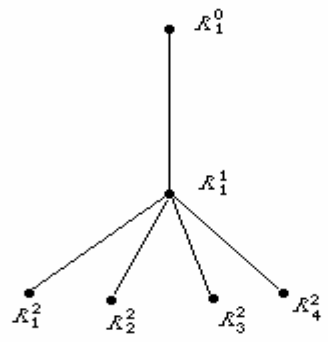


Figure 5.1. A set of hierarchically related keywords

שיטה ברורה לעשות אימפלמנטציה של החיפוש הבוליאני היא דרך הקובץ שעבר המרה. שומרים רשימה עבור כל מילת מפתח באוצר המילים ובכל רשימה משתילים את הכתובת (או המספרים) של המסמכים המכילים את המילה המיוחסת. על מנת לענות על צרכי השאילתה, מבצעים את הפעולות הקבועות המתאימות לקישורים הלוגיים ברשימת ה- Ki (Ki lists) .

למשל אם :

K1 - list: D1, D2, D3, D4

K2 - list: D1, D2

K3 - list: D1, D2, D3

K4 - list: D1

And Q = (K1 AND K2) OR (K3 AND (NOT K4))

כדי לענות על הדרישה של (K1 AND K2) אנו מצליבים את רשימות K1 ו - K2 וכדי לענות על הדרישה של ((K3 AND (NOT K4) אנו מפחיתים את רשימת K4 מרשימת K3. ה - OR מתבטא באוסף האיחוד של שתי קבוצות המסמכים שנשלפו עבור השאילתה. התוצאה מתבטאת בקבוצה {D1, D2, D3} שעונה על דרישות השאילתה כאשר כל מסמך הכלול בקבוצה מתאים בדיוק לתנאי השאילתה. (TRUE).

מודיפיקציה קלה של אסטרטגית חיפוש בוליאנית מושלמת היא כזאת שמאפשרת לוגיקת AND אך לוקחת בחשבון את מספר המונחים המשותפים לשאילתה ולמסמך. מספר זה מוגדר בתור "רמת תיאום" (Coordination Level). אסטרטגיית חיפוש מסוג זה נקראת התאמה פשוטה (Simple Matching) . בכל רמה יש יותר ממסמך אחד ולכן המסמכים מדורגים חלקית על פי רמת התאום/התאמה.

עבור הדוגמא הקודמת והשאילתה Q = K1 AND K2 AND K3 נראה את הדרוג הבא :

רמת תאום :

3 D1, D2

2 D3

1 D4

ניתן לראות התאמה פשוטה כהתאמה שנעשה בה שימוש בפונקצית התאמה פרימיטיבית. לכל מסמך D אנו מחשבים |D [(intersection)]Q| שמבטא את הגודל של החפיפה שין D ו Q כאשר כל אחד מהם מבוטא בתור קבוצת מילות מפתח. גודל זה הוא מקדם ההתאמה הפשוטה.

פונקציות תואמות

אסטרטגיות החיפוש המתוחכמות יותר מיושמות ע״י פונקציה תואמת. זוהי פונקציה הדומה למדד אסוציאטיבי אך שונה ממנו בכך שהיא מודדת את האסוציאציה בין שאילתה למסמך או לפרופיל של אשכול לעומת מדד אסוציאטיבי המופעל על אובייקטים מאותו הסוג. מתמטית, לשתי הקבוצות אותם המאפיינים אך התרגום שלכל אחת מהפונקציות שונה.

בספרות קיימות מס׳ רב של דוגמאות של פונקציות תואמות. הדוגמא הפשוטה ביותר היא זאת המקושרת לאסטרטגית החיפוש העובדת על בסיס עקרון ההתאמה הפשוטה.

באם M היא הפונקציה התואמת, D היא קבוצה של מילות מפתח המייצגות את המסמך ו - Q היא קבוצה המייצגת את מרכיבי השאילתה:

$$M = \frac{2|D \cap Q|}{|D| + |Q|}$$

זאת דוגמא נוספת של פונקציה תואמת שדומה ל - Dice's Coefficient בפרק 3.

פונקציה פופולארית שנעשה בה שימוש בפרויקט ה SMART נקראת קורלציית קוסינוס. ההנחה היא שהמסמך והשאילתה מיוצגים בתור וקטורים במרחב t.

Q= (q1,q2,…qt) D= (d1,d2,…dt) כאשר qi ו -di הם משקלים ספרתיים האסוציאטיביים למילת מפתח i. קורלציית קוסינוס עכשיו פשוטה:

$$r = \frac{\sum_{i=1}^{t} q_i d_i}{\left(\sum_{i=1}^{t} (q_i)^2 \sum_{i=1}^{t} (d_i)^2 \right)^{\frac{1}{2}}}$$

או בסימון של מרחב וקטורי עם נורמה אוקלידית:

$$r = \frac{(Q, D)}{\|Q\| \, \|D\|} = \text{cosine } \theta$$

כאשר תטה זאת הזוית בין הווקטור Q לווקטור D.

חיפוש סדרתי

אעפ״י שחיפושים סדרתיים נחשבים לאיטיים, משתמשים בהם כחלק ממערכות גדולות יותר. בנוסף, חיפושים אלה מהווים דוגמא לשימוש בפונקציות תואמות. נניח שישנם N מסמכים בשם Di במערכת. החיפוש הסדרתי מחשב N ערכים M(Q, Di) וכך קבוצת המסמכים המתאימה נשלפת. ישנם 2 דרכים להגיע לתוצאה זאת:

1. לפונקציה התואמת ניתן סף מתאים כאשר המסמכים מעל הסף בלבד נשלפים. אם T הוא הסף אז קבוצה B הנשלפת היא הקבוצה {Di| M(Q, Di)>T}.
2. המסמכים מדורגים בסדר עולה עפ״י התאמת ערך פונקציונאלי. רמת דירוג R מוגדרת כסף והמסמכים הנשלפים הם אלה שערכם נמוך מדרגה זאת כך ש B= {Di|r(i)<R} ו r(i) הוא הדרגה שניתנה ל Di. בכל מקרה בפני עצמו התקווה היא שהמסמכים הרלוונטיים מוכלים בקבוצה הנשלפת.

הקושי העיקרי עם אסטרטגיית חיפוש זו היא הגדרת הסף/ נק׳ החיתוך. הגדרה זו תהיה שרירותית הכיוון שאין דרך לנבא מראש את הערך המתאים שייתן את תוצאת השליפה הטובה ביותר עבור כל שאילתה.

מייצגי אשכולות

לפני שנוכל לעסוק באסטרטגיית חיפוש המוחלת על אשכול של אוספי מסמכים, נצטרך להתייחס למתודות המשמשות לייצוג אשכולות. באם בחיפוש סדרתי אנו מתאימים שאילתה לכל מסמך בקובץ, בחיפוש של מסמך מאוגד, אנו נדרשים להתאים שאילתות לאיגודים (אשכולות). למטרה זו, אשכולות מיוצגות ע״י פרופיל מסוים הנקרא ״נציג אשכול״. מטרת מייצג זה היא לסכם ולאפיין את אשכול המסמכים.

נציג אשכול צריך להיות כזה שיאפשר לשאילתה נכנסת להיות משויכת לאשכול המכיל מסמכים הרלוונטיים לשאילתה. במילים אחרות, אנו מצפים שנציג האשכול יפריד את המסמכים הרלוונטיים, המתאימים לדרישות השאילתה, מהמסמכים שאינם רלוונטיים. לצערנו, אין תיאוריה המאפשרת לבחור את נציג האשכול המתאים. הדרך היחידה היא להמשיך בדרך ניסויית. ישנם כמה דרכים הגיוניות המשמשות לאפיון אשכולות ועתה נותר לבצע את המבחנים בניסויים על מנת להחליט איזה דרך היא האפקטיבית ביותר.

נתחיל בדוגמא של נציג אשכול פרימיטיבי. באם נניח שאשכולות מופקות ממתודת איגוד המבוססת על מדד השונות, אז ניתן לייצג כל אשכול בדרגת שונות מסוימת ע״י גרף. בגרף זה A ו - B הם שני אשכולות. המפרקים מייצגים מסמכים והקו העובר בין שני מפרקים מעיד על כך שהמסמכים התואמים הם בעלי שונות נמוכה מדרגת שונות מוגדרת.

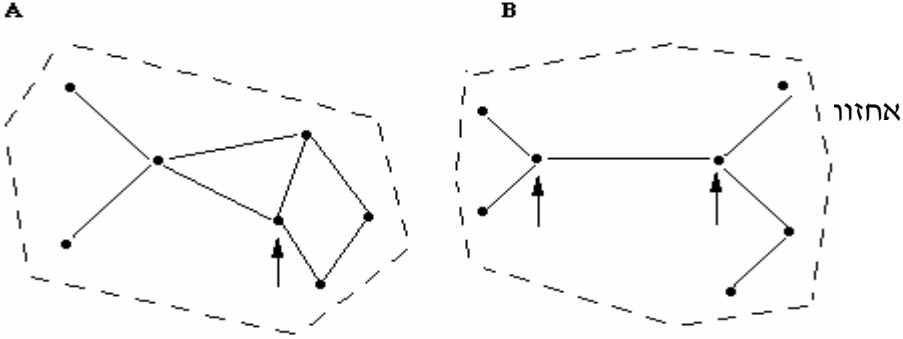


Figure 5.2. Examples of maximally linked documents as cluster representatives

כעת, דרך אחת לייצג אשכול היא בחירה של "חבר טיפוסי" מהאשכול. דרך פשוטה לעשות זאת היא ע"י מציאת המסמך המקושר למספר מסמכים מקסימאלי באשכול. השם המתאים לנציג אשכול מסוג זה הוא "מסמך המקושר מכסימלית". באשכולות A ו - B לעיל, החיצים מצביעים על מועמדים לתואר זה. כמצופה, במקרים מסוימים הנציג המתאים אינו ייחודי. למשל, באשכול B ישנם 2 מועמדים. על מנת להתמודד עם מצב זה ניתן לבחור 'מועמד' באופן שרירותי או לתחזק רשימה של נציגי אשכול המתאימים לאשכול מסוים. המוטיבציה לבחור בנציג האשכול הספציפי מפורטת ב Van Rijsbergen אבל זה לא שייך כרגע לענייננו.

נסתכל כעת בדרכים נוספות לייצוג אשכולות. אנו מחפשים מתודת ייצוג המשקללת את התיאורים של חברי האשכולות. המתודה העולה בראש היא זאת שבה מחשבים את מרכז הכובד של האשכול. באם {D1, D2…Dn} הם מסמכים באשכול וכל Di מיוצג ע"י וקטור מספרי {d1, d2, ….dn} אז מרכז הכובד C מחושב ע"י

$$C = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i}{\|D_i\|}$$

כאשר ||Di|| מייצג בד"כ את הנורמה האקלידיאנית :

$$\|D_i\| = \sqrt{d_1^2 + d_2^2 + \ldots + d_t^2}$$

במקרים רבים המסמכים לא מיוצגים ע"י וקטורים מספריים אלא ע"י וקטורים בינאריים (או קבוצות של מילות מפתח). במקרים אלה עדיין ניתן להשתמש בנציג אשכול בעל מאפיין מרכז כובד אך במקום נורמליזציה ישנו תהליך הקובע סף למרכיבי הסכום [Sigma]]Di . ליתר דיוק, Di יהווה וקטור בינארי כך ש - 1 במקום ה - j יעיד על המצאות מילת מפתח יעיד על המצאות מילת מפתח j במסמך ו - 0 יעיד על ההפך. מסיקים את נציג האשכול מהווקטור הסכומי עפ"י הפרוצדורה הבאה : $S = \sum Di$
(n הוא מספר המסמכים באשכול).

נניח C= נציג האשכול ו - [Di]j, המרכיב העקרי של הווקטור הבינארי. שתי השיטות הם:

$$(1) \qquad c_j = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} [D_i]_j > 1 \\ 0 \text{ otherwise} \end{cases}$$

or

$$(2) \qquad c_j = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} [D_i]_j > \log_2 n \\ 0 \text{ otherwise} \end{cases}$$

בסופו של דבר הווקטור הבינארי C הוא נציג האשכול. בשני המקרים האינטואיציה מכתיבה שיש להתעלם ממילות המפתח המופיעות באשכול פעם אחת בלבד. במקרה השני אנו מנרמלים את הגודל n של האשכול.

ישנם ראיות לכך ששתי מתודות הייצוג אפקטיביות כאשר משתמשים בהם בשיתוף עם אסטרטגיית החיפוש המתאימה (ראה לדוגמא Murray[5] and Van Rijsbergen[4]). ישנם וריאציות להשגת נציגי אשכולות אך כמו במקרה של מדדים אסוציאטיביים, לא סביר שאפקטיביות השליפה תשתנה במידה רבה ע"י מציאת נציגי אשכול מגוונים. סביר יותר שהדרך בה אסטרטגיית החיפוש עושה שימוש בנתונים בתוך נציגי האשכול תשפיע במידה רבה יותר.

דרך תיאורטית להסתכלות על מבנה של נציגי אשכול היא הסתכלות דרך רעיון המנבא המכסימלי לאשכול. המסמכים Di באשכול הם וקטורים בינאריים ונציג בינארי של האשכול הוא המנבא במידה וכל מרכיב (Ci) מנבא את הערך הסביר ביותר של המאפיין של המסמכים הרלוונטיים. המנבא הוא מקסימאלי באם הספר הניבויים שאומתו מקסימלים. באם מניחים שכל חבר באשכול מסמכים Di, ….,Dn סביר באותה מידה אז מספר הניבויים השקריים (או מכיוון שהכול בינארי ניתן להגיד בפשטות שמספר המקרים שבהם יש חוסר התאמה בין נציגי האשכול למסמכים באשכול) הוא:

$$\sum_{i=1}^{n} \sum_{j=1}^{t} ([D_i]_j - c_j)^2$$

זה יכול להיכתב בתור:

$$\sum_{i=1}^{n} \sum_{j=1}^{t} ([D_i]_j - D_{\cdot j})^2 + n \sum_{j=1}^{t} ([D_i]_j - c_j) \qquad (*)$$

where

$$D_{\cdot j} = \frac{1}{n} \sum_{i=1}^{n} [D_i]_j$$

הביטוי (*) יהיה מינימאלי תוך כדי מקסום מספר הניבויים המאומתים כאשר C = (C1,…Ct) נבחר כך ש:

$$\sum_{j=1}^{t} ([D_i]_j - c_j)^2$$

הוא מינימום. את זה ניתן להשיג ע"י:

$$c_j = \begin{cases} 1 & \text{if } D_j > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

במילים אחרות, מילת מפתח תשויך לנציג אשכול באם היא מופיעה ביותר מחצי האשכול. זה מהווה טיפול בטעויות ניבוי הנגרמות כתוצאה מחוסר או המצאות מילות מפתח במידה שווה. Croft [7] הראה שהגיוני יותר להבחין בין שתי סוגי הטעויות באפליקציה IR. הוא הראה שניבוי שקרי O(Cj=0) יגרום להפסד גדול יותר מאשר ניבוי שקרי 1(Cj=1). בהנחה זו הערך של 1/2 המופיע ב (3) מוחלף בקבוע הנמוך מ 1/2 כאשר ערכו המדויק מותאם לחשיבות היחסית של שני סוגי טעויות הניבוי.

אף על פי שהסיבה העיקרית לבניית נציגי אשכול היא הובלת אסטרטגיית חיפוש למסמכים רלוונטיים, יש צורך להבהיר שנציגים אלה יכולים לנתב חיפוש למסמכים העונים על תנאי בפונקציה התואמת. למשל, נרצה לשלוף את כל המסמכים Di שמתאימים יותר ל- Q מאשר ל- T. {Di|M(Q,Di)>T} לפרטים נוספים על הערכה של נציג אשכול (3) ניתן לבחון את עבודתו של Yu et al [8,9].

התנגדות מרכזית לעיסוק בנציגי אשכול טוענת שהסתכלות כזאת מטפלת בהתפלגות של מילות מפתח באשכולות בתור בודדים. גישה זו לא ריאלית. לצערנו, אין מחקר המצילה את המצב חוץ מזה של Ardnaudov and Govorun [1].

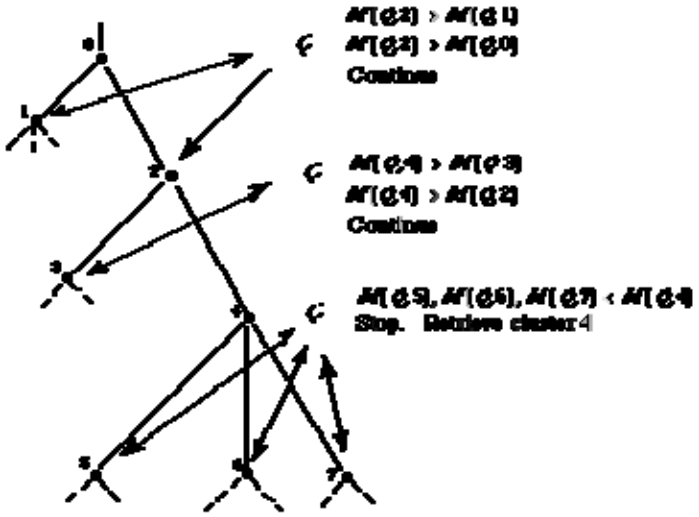
בנוסף, יש לזכור ששיטות איגוד הממשיכות ישירות מהגדרות מסמך לסיווגו בלי לחשב מראש את מקום שונות הביניים הצטרכו לבחור בנציג אשכול בעניים עצומות. נציגי אשכול אלו משופרים בנוסף לאלגוריתם וכתוצאה מכך מתאימים את הסיווג עפ"י פונקציה אובייקטיבית.

שליפה מבוססת אשכול

יסודה של שליפה מבוססת אשכול הנה היפותזת האיגוד שטוענת שמסמכים דומים זה לזה נוטים להיות רלוונטיים לאותה בקשה. פעולת איגוד בוחר מסמכים קשורים זה לזה ומקבץ אותם ביחד לאשכול אחד. בפרק 3, דנתי במספר רב של דרכים לעשות זאת, בפרק זה אתעלם מהמכניזם עצמו שיוצר את הסיווג ואתמקד על איך ניתן לערוך את החיפוש כשהמטרה היא שליפת מסמכים מתאימים.

בהנחה שיש לנו סיווג היררכי של מסמכים, אסטרטגיית החיפוש תתבצע (ראה תרשים 5.3 לפרטים). החיפוש מתחיל בשורש העץ, צומת 0 בדוגמא להלן. החיפוש מתקדם על ידי הערכת פונקציה מתאימה בצמתים היורשים הקרובים ביותר לצומת 0, בדוגמא אלו הם צמתים 1 ו- 2. הדגם הזה חוזר על עצמו לאורך העץ. החיפוש מודרך על ידי כלל החלטה, שעל בסיס השוואת הערכים של

הפונקציה מתאימה בכל שלב, מחליט איזו צומת להרחיב. בנוסף לכך, חשוב שיהיה כלל עצירה שמפסיק את החיפוש ומחייב שליפה. בתרשים 5.3 כלל ההחלטה הוא: הרחב את הצומת המתאימה המקבילה לערך המכסימלי של הפונקציה התואמת שהושג בקבוצת בת. כלל העצירה הוא: עצור אם המקסימום הנוכחי הוא פחות מהמקסימום הקודם.



תרשים 5.3 עץ חיפוש והערכים המתאימים של הפונקציה מתאימה מתאר את הפעולה של כלל החלטה וכלל עצירה.

להלן כמה הערות בנוגע לאסטרטגיה זו:

(1) אנחנו מניחים ששליפה יעילה ניתנת להשגה על ידי מציאת אשכול אחד בלבד;

(2) אנחנו מניחים שכל אשכול ניתן להצגה מספקת על ידי אשכול מייצג לצורך מציאת האשכול המכיל את המסמכים המתאימים.

(3) אם המקסימום של פונקציה מתאימה אינו יחיד, יהיה צורך בפעולה מיוחדת, למשל פעולת מבט קדימה.

(4) החיפוש תמיד מסתיים וישלוף לפחות מסמך אחד.

פועל יוצא של חיפוש זה הוא האפשרות לחיפוש להתקדם כלפי מטה ביותר מענף אחד מהעץ בכדי לאפשר שליפה של יותר מאשכול אחד. כלל ההחלטה וכלל העצירה יהיו בהכרח מעט יותר מורכבים, כאשר ההבדל העיקרי הוא הצורך בהכנות עבור הסתכלות רטרואקטיבית (back-tracking). דבר זה יקרה כאשר אסטרטגיית החיפוש מעריך (על סמך הערך הנוכחי של הפונקציה המתאימה) שהתקדמות לאורך ענף מסוים הנו בזבוז זמן, ובנקודה זו תתאפשר שליפה או חוסר שליפה של האשכול הנוכחי. לאחר מכן החיפוש חוזר אחורה לצומת הקודמת ובוחר בענף האלטרנטיבי.

את האסטרטגיות הנ"ל ניתן לכנות חיפושי *למעלה-למטה* (top-down searches). חיפוש *למטה-למעלה* (bottom-up) הוא כזה שמתחיל את החיפוש מאחת הצמתים הסופיים ומתקדם בצורה עולה לקראת שורש העץ. בצורה כזאת החיפוש יעבור דרך סדרה של אשכולות מוסתרים בגודל ההולך וגדל. כלל ההחלטה אינה דרושה, דרוש רק כלל עצירה שיכולה להיות בפשטות מקום הפסקה. חיפוש טיפוסי

היה מחפש את האשכול הגדול ביותר שמכיל את המסמך שהוצג על ידי הצומת ההתחלתית ושלא עולה על גודל מקום העצירה. כאשר האשכול נמצא, קבוצת המסמכים נשלפת. כדי ליזום את החיפוש כהיענות לבקשה, יש צורך לדעת מראש צומת סופי אחד המתאים לבקשה. אין זה יוצר דופן שהמשתמש כבר מכיר מסמך המתאים לבקשתו ומחפש מסמכים נוספים שדומים לו. אפשר להשתמש במסמך ה"מקור" הזה כדי ליזום חיפוש למטה-למעלה. כדי לקבל הערכה סיסטמאתית של חיפושי למטה-למעלה במונחי יעלות ראה Croft[7].

אם נעזוב עתה את הרעיון של אשכול רב-רמתי ונקבל את האשכול החד-רמתי, נקבל את הגישה של אשכול מסמכים עליו עבדו סלטון ושותפיו בצורה מעמיקה. שיטת האיגוד ההולמת מסווגת על ידי האלגוריתם של רושיו המתואר בפרק 3. אסטרטגיית החיפוש היא בחלקה חיפוש סדרתי. האלגוריתם מתקדם על ידי כך שתחילה מוצא את האשכול(ות) הטוב ביותר (או קרוב ביותר) ואז מחפש בתוך אשכולות אלו. השלב השני מושגת על ידי חיפוש סדרתי שי המסמכים באשכול(ות) שנבחר. הפלט הוא לעתים קרובות דירוג של המסמכים שנשלפו.

ניסוח חיפוש אינטראקטיבי

בדרך כלל משתמש הניצב מול מערכת שליפה אוטומטית לא יוכל לבטא את הצורך שלו במידע בפעם אחת. יש יותר סיכוי שירצה להשקיע בתהליך של ניסיון וטעייה שבו הוא מנסח את השאילתא שלו לאור מה שהמערכת תוכל לומר לו בקשר לשאילתא שלו. סוג אינפורמציה שהוא עשוי לרצות בכדי לנסח מחדש את השאילתא שלו הוא:

(1) תדירות החזרה של מושגי החיפוש שלו במאגר הנתונים,

(2) מספר המסמכים העשויים להישלל על ידי השאילתא שלו.

(3) מושגים אלטרנטיביים וקשרים שיהיו אלו שישמשו בחיפוש שלו,

(4) דוגמא קטנה של ציטטות העשויים להישלף,

(5) המושגים שנעשה בהם שימוש כמפתחות לציטטות ב (4).

את כל זה אפשר לספק למשתמש במשך זמן השאילתא על ידי מערכת שליפה אינטראקטיבי. אם הוא מגלה שאחד ממושגי החיפוש שלו חוזר לעתים קרובות, יתכן וירצה לפרט אותו יותר בעזרת מילון היררכי שיאמר לו מה הם האפשרויות שיש בידו. בדומה לכך, אם השאילתא שלי עשוי לשלוף יותר מידי מסמכים הוא יוכל לפרט את השאילתא.

הדוגמא של ציטטות והמפתחות שלהם יבהיר לו במידת מה איזה סוג של מסמכים עשויים להישלף וכן עד כמה היו מושגי החיפוש שלו יעילים בביטוי הצורך שלו במידע. הוא יכול לשנות את השאילתא שלו לאור השליפה לדוגמא. את התהליך שבו המשתמש משנה את השאילתא על סמך תוצאות חיפוש ממשי אפשר לתאר כצורה של משוב (feedback). דוגמאות תפעוליות וניסוייות של מערכות המספקים מכניזם מסוג זה הם MEDLINE וגם MEDUSA, המבוססים שניהם על מערכות MEDLARS. סוג נוסף מעניין של מערכת ניסויית מתוחכמות הינה זאת המתוארת על ידי Oddy.

כעת נסתכל על הגישה המתמטית לשימוש במשוב כאשר המערכת משנה בצורה אוטומטית את השאילתא.

משוב

המילה משוב בדרך כלל מתארת את המכניזם שבעזרתו מערכת יכולה לשפר את ביצועיה במשימה על סמך ביצועים בעבר. במלים אחרות, מערכת פשוטה של קלט-פלט מזין בחזרה את המידע מהפלטכדי שיהיה אפשרות לעשות בו שימוש כדי לשפר את הביצוע בקלט הבא. מושג המשוב מבוסס היטב במערכת בקרה ביולוגית ואוטומטית. הדבר נעשה פופולארי בספרו של Norbert Wiener, "Cybernetics" . בשליפת מידע, נעשה שימוש יעיל מאד במערכת.

הנח כעת אסטרטגית שליפה שהוחדר באמצעות פונקציה תואמת M. יתר על כן, נניח שהשאלת Q וכן מסמכים מייצגים D הם וקטורים t מימדיים בעלי מרכיבים ממשיים, כאשר t הינו מספר מושגי המפתח. מכיוון שהמטרה שלי היא להסביר משוב, אני אתייחס ליישום שלו בחיפוש סדרתי בלבד.

מטרתה של כל אסטרטגית שליפה היא לשלוף את המסמכים הרלוונטיים A ולעכב את המסמכים שאינם רלוונטיים A′. לצערנו, רלוונטיות מוגדרת על סמך הניסוח *הסמנטי* של השאילתא על ידי המשתמש. מבחינתו של מערכת השליפה, ניסוחו של המשתמש של השאילתא אינו בהכרח אידיאלי. ניסוח אידיאלי הינו זה שישלוף את המסמכים הרלוונטיים בלבד. במקרה של חיפוש סדרתי המערכת תשלוף כל D כאשר M(Q,D)<=T, כאשר T הוא גבול מוגדר. במקרה בו M הוא פונקצית קורלציה הקוסינוס, כלומר

$$M(Q,D) = \frac{(Q,D)}{\|Q\|\,\|D\|} = \frac{1}{\|Q\|\,\|D\|} \times (q_1 d_1 + q_2 d_2 \ldots q_t d_t),$$

תהליך ההחלטה M(Q,D)-T>0, מקביל לפונקציה דיסקרמננטית שנעשה בו שימוש להפריד ליניארית שני סטים: A ו- A′ בתוך R[T]. Nilsson[14] דן באריכות על איך אפשר "לאמן" פונקציות מסוג זה על ידי שינוי המשקלים qi כדי להבחין בין שתי קטגוריות בצורה נכונה. אם נניח לעת עתה ש-A ו-A′ ידועים מראש, אז נוסח השאילתא הנכון Qo יקיים M(Qo,D)>T כאשר D[[תת קבוצה מתאימה]] וגם M(Qo,D)<=T כאשר [[אלפא]]′ D[[תת קבוצה מתאימה P]].

הדבר המעניין הוא שכאשר מתחילים עם כל Q, ניתן להתאים אותו איטרטיבי על ידי שימוש במידע במשוב כדי שיתקרב ל-Qo. ישנה תיאוריה (Nilsson עמוד 81) הטוענת שבהינתן ש- Qo קיים, יש תהליך איטרטיבי שיבטיח ש Q יתלכד עם Qo המספר סופי של מהלכים.

התהליך האיטרטיבי פרוצדורת התוספת הקבועה לתיקון טעויות.

צורת התהליך:

Qi = Qi-1 + cD if M(Qi-1, D) - T <= 0

and D [[propersubset]] A

Qi = Qi-1 - cD if M(Qi-1, D) - T > 0

and D [[propersubset]] `A

אין שינוי ב- Qi-1 אם זה מאובחן נכון. C היא התוספת המתוקנת, ערכה שרירותית ולכן בדרך כלל מותאם ליחידה. בפועל, יתכן שיהיה צורך לעבור על קבוצת המסמכים מספר פעמים עד משיגים את ההרכב הנכון, במיוחד אלו שיפרידו את A ו-A' ליניארית (כל זה בהסתמך על כך שקיים פתרון)
המצב בשליפה עצמה אינו כל כך פשוט. איננו יודעים מראש את הקבוצות A ו-A', למעשה A היא הקבוצה אותה אנחנו מצפים לשלוף. בכל אופן, בהינתן ניסוח שאילתא Q והמסמכים שנשלפו על ידו, ניתן לשאול את המשתמש איזה מהמסמכים שנשלפו רלוונטיים ואיזה מהם לא. בשלב הזה המערכת יכולה לשנות את Q באופן אוטומטי כדי שלפחות תוכל לאבחן בצורה נכונה את המסמכים שהמשתמש ראה. ההנחה היא שזה ישפר את השליפה בסיבוב הבא בזכות העובדה שהביצוע של המערכת טובה יותר כשמתבצעת על מדם.
שוב אין זה כל התמונה. לעתים קרובות קשה להגדיר את הגבול T מראש כך שהמסמכים מדורגים בערך מותאם יורד על הפלט. כעת קשה ויתר להגדיר מהו ניסוח שאילתא אידיאלי. Rocchio [15] בתזה שלו הגדיר את השאילתא Qo האופטימאלי כזאת שאפשר להביא למקסימום את:

$$\Phi = \frac{1}{|\bar{A}|} \sum M(Q, D) - \frac{1}{|\bar{A}|} \sum M(Q, D)$$

אם משתמשים ב-M כפונקצית קוסינוס ((D|| || Q|| / (Q, D ‖ אז קל להראות את [[Phi]] אפשר להביא למקסימום ע"י

$$Q_0 = c\left(\frac{1}{|A|}\sum_{D\in A}\frac{D}{\|D\|} - \frac{1}{|\bar{A}|}\sum_{D\in \bar{A}}\frac{D}{\|D\|}\right)$$

כאשר c הוא קבוע פרופורציונאלי שרירותי.
אם במקום שהסיכום יעשה על A ו-A' יהיו על Bi[חותך]A ו-Bi[חותך]A' כאשר Biהוא קבוצה של מסמכים שנשלפו באיטרציה ה-I, אז יהיה לנו נוסח שאילתא אופטימאלי עבור Bi, תת קבוצה של אוסף המסמכים. על ידי אנלוגיה למיון הליניארי שנעשו בו שימוש לפני כן, נוסיף את הווקטור הזה לנוסח השאילתה בשלב ה-I כדי לקבל:

$$Q_{i+1} = w_1 Q_i + w_2\left[\frac{1}{|A\cap B_i|}\sum_{D\in A\cap B_i}\frac{D}{\|D\|} - \frac{1}{|\bar{A}\cap B_i|}\sum_{D\in \bar{A}\cap B_i}\frac{D}{\|D\|}\right]$$

כאשר w1 ו- w2 הם מקדמים נוספים. לאמיתו של דבר, Salton[2] בהשתמש בוורסיה שונה במקצת. ההבדל החשוב ביותר הוא שיש אפשרות ליצור Qi+1 מ-Qi, או Q, השאילתא המקורית. התוצאות של כל השינויים ניתן לסכם בכך שהשאילתא משתנית אוטומטית כך שמושגי המפתח במסמכים נשלפים רלוונטיים, מקבלים יותר משקל ומושגי מפתח שאינם רלוונטיים מקבלים פחות משקל.
ניסויים הראו שמשוב רלוונטיות יכול להיות מאד יעיל. לצערנו, מידת היעילות קשה למדידה, מכיוון שקשה להפריד את התרומה ליעילות השליפה הנוספת שנוצר כאשר מסמכים יחידים עולים בדרגה

מהתרומה הנוצרת כאשר מסמכים *חדשים* נשלפים. כמובן שהשני הוא מה שמעניין יותר את המשתמש.

לבסוף, כמה הערות בנוגע לטכניקת של משוב רלוונטיות באופן כללי. נראה לי שהביצוע של הטכניקה על בסיס תפעולי יכול להיות יותר בעייתי. לא ברור איך משתמשים יאמדו רלוונטיות או חוסר רלוונטיות של מסמך על סמך עודות מועטות כל כך כגון ציטטות. במערכת תפעולית קל לסדר שתקצירים יהיו הפלט אבל מן הסתם המשתמש יצטרך לדפדף במסמכים עצמם שנשלפו כדי לקבוע את הרלוונטיות ולאחר מכן הוא יוכל קרוב לוודאי לנסח שאילתא חדשה *בעצמו*.

הערות ביבליוגרפיות

הספר של Lancaster and Fanyen[16] מכיל פרטים של מערכות און-ליין תפעוליות. Barraclough[17] כתב מאמר על סקר מעניין בנוגע לחיפוש און-ליין. דיונים על אסטרטגיות חיפוש נמצאים בדרך כלל בתוך מאמרים כללים יותר על שליפת מידע. אך בכל אופן יש מספר הפניות מומחיות ששווה להזכיר.

מאמר קלסי חדש על המגבלות של חיפוש בוליאני הוא Verhoeff et al[18]. Miller[19] ניסה להתחמק מחיפוש בוליאני פשוט על ידי הצגת צורה של משוב אוטומטי לתוך חיפוש בוליאני. Angione[20] דך בשוויון בין החיפוש הבוליאני והחיפוש המשקלי. Rickmann[21] תיאר שיטה של הצגת משוב אוטומטי לתוך חיפוש בוליאני. Goffman[22] חקר אסטרטגית חיפוש מעניינת המבוסס על הרעיון שהרלוונטיות של מסמך לשאילתא היא תלויה ברלוונטיות של מסמכים אחרים לשאילתא. במאמר מוקדם של Hyvarinen[23] ניתן למצוא הגדרה אינפורמטיבי-תיאורטי של "החבר הטיפוסי" של נציג אשכולי. Negoita[24] מציג דיון באסטרטגית חיפוש למטה-למעלה בהקשר של שליפה מבוססת אשכול. הרבה מהעבודה שנעשתה בפרויקט SMART מודפס מחדש כעת ב-Salton[25]. עוד שתי מאמרים בלתי תלויים אל משוב הם Stanfell[26] ו Bono[27].

References

1. KNUTH, D.E., *The Art of Computer Programming*, Vol. 3, *Sorting and Searching*, Addison-Wesley, Reading, Massachusetts (1973).

2. SALTON, G., *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York (1968).

3. van RIJSBERGEN, C.J., 'The best-match problem in document retrieval', *Communications of the ACM*, **17**, 648-649 (1974).

4. van RIJSBERGEN, C.J., 'Further experiments with hierarchic clustering in document retrieval', *Information Storage and Retrieval*, **10**, 1-14 (1974).

5. MURRAY, D.M., 'Document retrieval based on clustered files', Ph.D. Thesis, Cornell University Report ISR-20 to National Science Foundation and to the National Library of Medicine (1972).

6. GOWER, J.C., 'Maximal predictive classification', *Biometrics*, **30**, 643-654 (1974).

7. CROFT, W.B., *Organizing and Searching Large Files of Document Descriptions*, Ph.D. Thesis, University of Cambridge (1979).

8. YU, C.T., and LUK, W.S., 'Analysis of effectiveness of retrieval in clustered files', *Journal of the ACM*, **24**, 607-622 (1977).

9. YU, C.T., LUK, W.C. and SIU, M.K., 'On the estimation of the number of desired records with respect to a given party' (in preparation).

10. ARNAUDOV, D.D.. and GOVORUN, N.N. *Some Aspects of the File Organisation and Retrieval Strategy in Large Databases*, Joint Institute for Nuclear Research, Dubna (1977).

11. Medline Reference Manual, Medlars Management Section, Bibliographic Services Division, National Library of Medicine.

12. BARRACLOUGH, E.D., MEDLARS on-line search formulation and indexing, *Technical Report Series*, No. 34, Computing Laboratory, University of Newcastle upon Tyne.

13. ODDY, R.N., 'Information retrieval through man-machine dialogue', *Journal of Documentation*, **33**, 1-14 (1977).

14. NILSSON, N.J., *Learning Machines - Foundations of Trainable Pattern Classifying Systems*, McGraw-Hill, New York (1965).

15. ROCCHIO, J.J., 'Document retrieval systems - Optimization and evaluation', Ph.D. Thesis, Harvard University, Report ISR-10 to National Science Foundation, Harvard Computation Laboratory (1966).

16. LANCASTER, F.W. and FAYEN, E.G., *Information Retrieval On-line*, Melville Publishing Co., Los Angeles, California (1973).

17. BARRACLOUGH, E.D., "On-line searching in information retrieval', *Journal of Documentation*, **33**, 220-238 (1977).

18. VERHOEFF, J., GOFFMAN, W. and BELZER, J., 'Inefficiency of the use of boolean functions for information retrieval systems', *Communications of the ACM*, **4**, 557-558, 594 (1961).

19. MILLER, W.L., 'A probabilistic search strategy for MEDLARS', *Journal of Documentation*, **17**, 254-266 (1971).

20. ANGIONE, P.V., 'On the equivalence of Boolean and weighted searching based on the convertibility of query forms', *Journal of the American Society for Information Science*, 26, 112-124 (1975).

21. RICKMAN, J.T., 'Design consideration for a Boolean search system with automatic relevance feedback processing', *Proceedings of the ACM 1972 Annual Conference*, 478-481 (1972).

22. GOFFMAN, W., 'An indirect method of information retrieval', *Information Storage and Retrieval*, **4**, 361-373 (1969).

23. HYVARINEN, L., 'Classification of qualitative data', *BIT, Nordisk Tidskrift för Informationsbehandling*, **2**, 83-89 (1962).

24. NEGOITA, C.V., 'On the decision process in information retrieval', *Studii si cercetari de documentare,* **15**, 269-281 (1973).

25. SALTON, G., *The SMART Retrieval System - Experiment in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, New Jersey (1971).

26. STANFEL, L.E., 'Sequential adaptation of retrieval systems based on user inputs', *Information Storage and Retrieval*, **7**, 69-78 (1971).

27. BONO, P.R., 'Adaptive procedures for automatic document retrieval', Ph.D. Thesis, University of Michigan (1972).